1 Introduction to Cryptology

The goals for this section are pretty simple, to introduce you to cryptology and familiarize you with some of the basic terminology. You will also learn how cryptology relates to cyber security, and who is in charge of the legal and technical aspects of cryptology. The specific things you should be able to do at the end of this section are:

1. Document why you are taking this class, and why cryptology is important in cyber security and information assurance.
2. Describe how cryptology relates to cyber security
3. Define cryptology and build the "family tree" showing the relationship between cryptology, steganography, cryptography, cryptanalysis, symmetric and asymmetric ciphers, and hashing.
4. Explain the best way to implement cryptologic solutions and explain why this is the best choice. That is, whether it's better to use functions built into an OS, or whether it's better to write custom solutions.
5. Describe the relationship between math and cryptology and explain why math is the language of cryptology.
6. Decide how you feel cryptology should affect society's balance between privacy and security. That is, start thinking about whether it's better for society if the use of cryptology is controlled or restricted, and if so, by who makes those decisions.
7. List the entities in charge of the legal aspects of cryptology.
8. List the entities in charge of the technical aspects of implementing cryptology on the Internet.
9. Identify Alice, Bob, and Eve.

Required Reading

1. **Read this document first as it will provide you with the information regarding all of the subjects covered in this section.**

(If you want to learn more about any of the subjects discussed in this chapter, I've added links to several web sites at the end of this document.)

Section Content

Do you have any secrets? Or information that you want to keep private? Sure you do, almost everyone does.

And how do you keep your secrets secret, or keep your private information private? (Besides not broadcasting it on social media. LOL.) A hundred years ago you could have locked away your private papers to prevent anyone else from reading them. But those days are long gone. In today's world, with information being stored and transmitted electronically a different form of protection is required. Your sensitive information is stored on systems controlled by your bank, your state government, the federal government, and maybe even Microsoft, Apple, Google, or Amazon if you've stored anything in the cloud. Do you trust the IT support people at all these organizations to keep your data secure and not read it themselves? And now consider that any information you send across the Internet such as ecommerce transactions will be passed through dozens of routers, which all have the ability to easily read the information.

In all these situations, where so many individuals and organizations have access to your data, how do we keep information private? How do we keep our secrets? Of course, I'm sure you know that the answer is we use encryption to scramble data to keep it private, and we use passwords to control access to the scrambled data. Knowing about encryption and passwords is considered basic knowledge. It's so basic that we expect everyone to know about them. One of the first things new computer users are taught is to do things like choose strong passwords, how to store the passwords safely, and to make sure and look for the lock icon or https in the web site URL when they're online shopping.

But how does the system of encryption and passwords really work? If you're going to work with computers or networks, or work in cyber security, or if you're going to do any software development; you're going to need to be able to answer this question. The reason individuals in these career fields need to know more than an average user is because there are several different ways to encrypt data, and several different password systems. Knowing how the various methods function, and how to configure them properly is key to providing adequate security. If you don't know how things work or how to configure them, it can be easy to make mistakes that will lead to gaping security holes.

Learning about encryption and password systems is where this book comes in. The material in this will help you learn about the science of cryptology, which encompasses the fields of cryptography, cryptanalysis, and steganography.

What Are Cryptography, Cryptanalysis, Steganography, and Cryptology?

Telling you that you're going to learn about cryptography, steganography, cryptanalysis and cryptology may be sufficient if you already know what all of those terms refer to, but if not, it might not be of much help. So, let's spend a few minutes and go over what these terms mean so you'll get a better idea what the book is about.

Cryptography

Most people are familiar with the term cryptography so let's start with there. Cryptography is associated with the process of scrambling messages or data to keep them private, as well as providing ways for authorized users to unscramble the data. Most people recognize that cryptography is the thing that makes it possible to shop online and connect to wireless networks. Cryptography and encryption are also common props in spy movies or TV shows, where a secret message or secret data needs to be decrypted to save the world. Plus, almost every Internet user knows they need to look for the lock icon on websites when they're sending sensitive data or shopping online, as it indicates the site uses encryption to keep their data safe.

In the past up through the early 1900s, cryptography could be done by hand, with code books or decoder rings. Classic cryptographic systems were fairly straightforward methods of scrambling the letters in a message, and they could be performed manually. That is, a cryptographer would take a plain text message and write out the enciphered text. They might have to use a table or chart to encipher the text, but they could do it by hand.

Modern cryptography has evolved to a distinct blend of mathematical theory and computer science. The days of a person using or discovering a secure encryption algorithm that can be done by a person with a pencil and paper are over, now it's all math and computers. And instead of working on text, modern systems are designed to work on binary numbers which makes it possible to encrypt any and all kinds of data.

Cryptography is now so based on math, and so reliant on math that the definition of modern cryptography is the design, analysis, and implementation of mathematical techniques for

securing information systems and computation. The mathematical theory is used to describe the process for scrambling the data, as well as provide proof that it cannot be unscrambled. The mathematicians that work on cryptography are geniuses, literal geniuses, and sadly most receive very little credit for the work they've done. Even though they've had a huge positive impact on almost everyone's life, they aren't famous like movie stars or athletes who most people know about, but have done little to actually improve anyone's life.

In cryptography, the theory is based on math. But as in any field, theory by itself doesn't do anything. Someone needs to find a way to apply the theory for it to become useful. This is where computer science and programmers become part of the process as they're the individuals that write the code that implement the algorithms developed by the mathematicians.

Cryptanalysis

The next term to define is cryptanalysis. This is the science of breaking encryption schemes and it's the flip side of the cryptography coin. That is, cryptography is used when a society or group wants to keep their messages secret, but just keeping their own information secret isn't enough for most spymasters. Typically, they also want to read everybody else's secret messages, which requires finding a way to crack the encryption schemes used by their opponents. And surprisingly they often want to read secret messages sent by their allies as well. The art or science of breaking encryption schemes is called cryptanalysis.

A notable point about cryptology is that throughout history, every encryption scheme invented was thought to be unbreakable. Most were secure for many years, and some of them proved to be extremely difficult to crack and provided security for hundreds of years. But sooner or later all these unbreakable schemes were broken, again by some literal geniuses who started the field of cryptanalysis. You may be familiar with the story of the Enigma code used by the Nazis in World War II, and how cryptanalysts were able to crack it. Enigma was extremely difficult to crack, but a group led by Alan Turing [1] built a specialized computer which allowed them to read some of the encrypted messages and shorten the war by years. There's more about Enigma later in the book, but for now the main concept you need to take away is the definition of cryptanalysis, which is the cracking of encryption schemes.

Another point to ponder is how secure is modern encryption? We might think the modern encryption algorithms can't be broken, which is exactly what people thought about every previous encryption scheme. But sooner or later all the previous schemes were cracked, so I

[1] https://www.iwm.org.uk/history/how-alan-turing-cracked-the-enigma-code

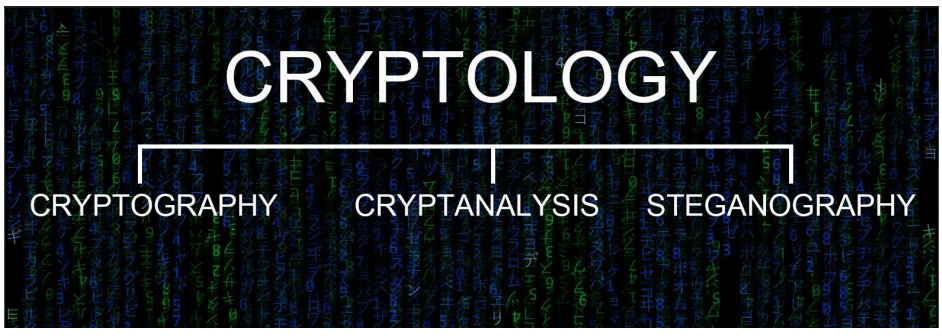
think there are good odds that someone has found a way around our current encryption schemes. In this class you'll learn about modern encryption and how it is supposed to provide perfect security. It will be interesting to see if it is truly unbreakable, or if another genius cryptanalyst will find a way to break the modern ciphers.

Steganography

The next term to learn about is steganography. It's a field related to cryptography, but it has some significant differences. The first difference is most people have heard of cryptography but very few know about steganography. Steganography refers to another way of keeping messages secret, however with steganography the characters aren't changed like they are in cryptography. Instead, the characters in a message are scrambled and put in a different order, or the messages are hidden by doing things like using invisible ink. You'll learn about some of the interesting historical steganography methods, and the modern methods later in the book. The main concept that you need to take away at this point is that with steganography the data or characters in a message are not changed, they're just hidden somehow.

Cryptology

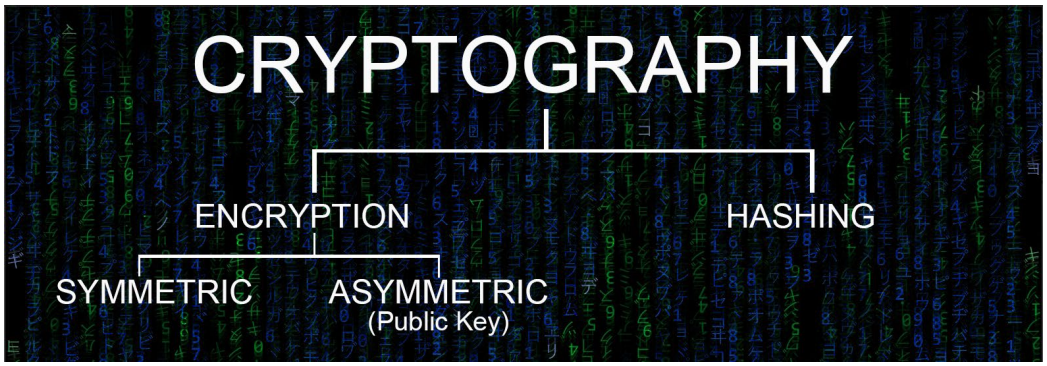
This brings us to the last term, which is cryptology. Cryptology is the overarching science or field that encompasses cryptography, cryptanalysis, and steganography. In other words, if you study cryptology, you're going to learn how to use cryptography and steganography to scramble or hide data, and how about the various methods used in cryptanalysis to crack codes.



Types of Cryptography

All fields of cryptology are important, and you will learn about each of them. However, currently by far the most widely used field is cryptography. It's used in almost every aspect of modern, connected life for doing things like e-commerce, securing wireless networks, providing copyright protection of DVDs, implementing password systems, in cryptocurrency, etc. Cryptography is also widely used by business and the military to secure their data and transmissions. Because it's the largest field in cryptology, most of the book will be dedicated to helping you learn cryptography and about the three main types of cryptographic techniques which are symmetric-key cryptography, public-key or asymmetric cryptography, and hashing.

You'll learn the theory of how each of these main techniques functions, what type of situations and problems each technique can solve, and some of the specific implementations of each technique.



Symmetric Key Cryptography This is a technique for encrypting and decrypting data where the sender and recipient share a single key. The sender uses the key to encrypt a message and the recipient decrypts the message using the same key. This can provide unbreakable security if implemented correctly but finding ways to share or exchange the keys is a major challenge.

Public Key or Asymmetric Cryptography This is an ingenious system of cryptography that solves the key exchange problem associated with symmetric cryptography. In asymmetric cryptography two keys are used. These are referred to as the public key, which as the name implies everyone can know, and the private key which only the recipient knows. Messages are encrypted with the public key and can only be decrypted by the recipient's private key. The two keys are related mathematically, but in a way that if implemented correctly, makes cracking the private key extremely time consuming, as in it should take hundreds of thousands of years.

Hash Functions These are functions which are used for doing things like protecting passwords or proving that data has not been changed. The hash functions are distinctly different than functions that are used to encrypt and decrypt data. The main differences are that hash functions are one-way, which means there's no way to reverse their output and calculate the

input. Another huge difference from encryption algorithms is that hash functions always produce the same size output regardless of what size the input is. To effectively use a hash, you need to hash the data and produce the output, and then hash the data again and compare the output. For example, after a password is hashed, the output is stored. When the user logs in again, the password they provide is also hashed and compared to the stored hash.

Etymology – or Origins and Meanings of Cryptography, Cryptanalysis, Steganography, and Cryptology

It seems like almost every cryptology textbook and blog I encounter provides the etymology for our terms. (Etymology refers to the word's origins.) This isn't crucial knowledge, but it may help you remember the terms and what they mean. And it may be valuable if you ever play Geek Jeopardy.

I'll start out by admitting that when I first encountered steganography, I thought it must be some Latin phrase that has something to do with dinosaurs. Obviously, my knowledge of languages isn't very good because as it turns out it's not even Latin. It comes from the Greek *steganos* which means cover or roof. (And for that matter, my knowledge of dinosaurs isn't that great either as it's a stegosaurus, not stegano.) And the *graphy* also comes from the Greek word *grapho* which means "I write". So, steganography means "covered writing".

I made a similar mistake with cryptography, where I assumed *crypt* had something to do with dead people and the crypts that hold dead bodies. As it turns out crypt comes from the Greek word kruptós which means hidden or secret. So, cryptography means "secret writing" and cryptology means "the science of secrets". (There actually is a study of dead people, but it's called thanatology.)

Cryptanalysis also uses the *crypt* followed by *analysis* which comes from the Greek which means "a breaking up, a loosening, releasing".

Cryptology and Cyber Security

When it comes to looking at the relationship between cryptology and cyber security there are two ways to go about it.

First, there's the traditional perspective, where we list the cyber security goals that cryptology can help achieve, which are confidentiality, integrity, authentication, non-repudiation, and access control. You may be familiar with these from other cyber security classes but take care that you don't confuse the first three, confidentiality, integrity, and *authentication*, with the CIA

triad of confidentiality, integrity, and *availability*. Hopefully you know the meanings of these terms, but here are some quick definitions just in case you need a refresher:

> **Confidentiality** – Cryptology ensures that only authorized individuals or entities are allowed to view protected data by encrypting it. Even if an attacker finds a way to access stored data or intercept a transmission, they won't be able to read the data if they can't decrypt it.
>
> **Integrity** – Cryptographic hashes provide a unique signature for a set of data, and if even a single bit of the data is changed the hash signature will change. Hashing a message before it's sent or stored, then hashing again later is used to provide assurance that messages or data have not been altered by unauthorized individuals.
>
> **Authentication** – Cryptographic hashes and digital signatures can be also used to validate the source of a message, to ensure that the person who sent the message is really who you think they are.
>
> **Nonrepudiation** – This is the flip side of Authentication. If a signature can be used to prove who sent a message, it can also be used to make it impossible for that person to deny having sent the message.
>
> **Access Control** – Cryptographic hashes are the basis for password systems, which only allow authorized users to access protected data or resources.

The second way to look at the relationship between cyber security and cryptology is to see what cyber security would be without cryptology. You've been using aspects of cryptology almost from the start of your degree program when you learned about choosing strong passwords to protect your identity or learned the importance of protecting data by encrypting it. And if you look back at what you've learned so far, you'll see that cryptology has played a key role in providing protection in almost everything you've learned about cyber security. In fact, without cryptology, cyber security wouldn't exist, at least not in the way it does today.

What do I mean by saying cyber security wouldn't exist without cryptology? Well, try to imagine how you would secure a computer system or how you'd secure data without cryptology.

Or, to better understand how crucial cryptology is to cyber security, let's try this. Let's look at analogy and compare a few of the main functions provided by cryptology to actions in the real, non-cyber world. The main functions provided by cryptology are protecting data using encryption, and authentication or proof of identity using hashing and password systems. Let's compare this to providing security in the real world, where physical possessions or items

can be protected by locks, and someone can prove their identity or authenticate themselves using some form of ID card. Locking things in a room or in a box would be analogous to encrypting it, and using an ID card for authentication would be analogous to using hashing and a password system.

Next, to see what it would be like if we took away cryptology, let's try and picture what security in the real world would be like if we took away locks and ID cards.

Can you imagine what it would be like trying to protect your possessions in the real world if couldn't use locks? No locks on your car doors. No locks on the doors on your home or apartment. No locks on bank doors. How would you protect anything?

And now assume that everyone was masked and cloaked, so you had no way to identify anyone. People could walk around and claim to be anyone they wanted, and you would have no way to check to see that a person was who they claimed to be. In this case, without ID cards, how could you trust anyone or know they are truly who they claim to be?

Without locks or ID cards we would live in a much different world. You'd have to either keep your possessions with you at all times or leave them with someone you trust. And if everyone were masked and cloaked, you wouldn't know who you could trust. You wouldn't be able to tell a stranger from someone in your immediate family.

Just as locks allow us to secure items in the real world, cryptology allows us lock systems and protect data in the cyber world. And like ID cards in the real world, password systems allow us to authenticate users and provide access to people who should be trusted. Without cryptology none of this would be possible in the world of computers and networks, and cyber security as we know it would not exist.

So, cryptology is the "key" building block for cyber security, and as such it's built into every aspect of protecting systems and data.

But up to this point you've probably never looked very deep at the algorithms and programs used to encrypt data or hash passwords. You might know the terms PKI or AES for encryption, or SHA-512 and MD5 for hashing. But knowing the names and knowing how they work are obviously much different points of knowledge.

Going back to our real-world analogy, up to this point you've learned about locks and ID cards, and that locks are good for protecting items from unauthorized access, and ID cards are a good way for someone to prove their identity. But if you only learned the names of various types of locks such as padlocks, or keyed door locks, or vault locks, and not how they work, it will be difficult to choose the correct lock for use in a specific situation. Without this knowledge you might choose the wrong lock for the job. For example, you might look at a padlock and chain, and decide to use it to lock your car doors since it's lightweight and portable. Or you might think that since bank vault locks are super secure, you'll install one on the garage door of your house. After all, home security is important, so why not use the strongest lock you can find. But imagine your surprise when you drive up to your house and find the garage door crumpled in the driveway because it couldn't support the weight of the lock used on a bank vault.

If you're interested in a career in cyber security, it's critical that you learn how the cryptologic algorithms work. One of the major goals of this book is to arm you with this knowledge so you'll know when and how to apply the various cryptologic algorithms and systems to best protect your data, passwords, systems, and networks.

Cryptology and Software Engineering & Development

Knowing how the cryptologic algorithms work is also critical knowledge for programmers and developers. While the majority of security issues are caused by users that fall for phishing attacks or even insider attacks, most of the vulnerabilities and holes in both application code and Operating System code are caused by problems with software development. Just look at the lists of vulnerabilities or lists of software updates required to patch security holes and you'll get an idea of the magnitude of the problem. The majority of these vulnerabilities are caused by one of the following programming mistakes:

1. **Not understanding the cryptographic algorithms and making an inappropriate choice**. This can be either choosing the wrong algorithm(s) altogether or choosing

implementation settings that create vulnerabilities. Going back to the lock analogy, this can be a mistake such as choosing the wrong type of lock, as we've discussed previously. Or it can be like choosing the right type of lock but not setting it up correctly. For example, some cryptographic algorithms can be configured to use different key sizes, or hash function can use something called a salt, which will affect their overall strength. This would be like choosing to use a combination lock where you could configure it to require between 2 and 10 numbers. If you configure the lock to use two numbers, it going to be much easier to crack than it would be if you configured it to use even 5 or 6 numbers.

2. **Writing custom code to implement a cryptographic algorithm**. Writing the actual code for *modern* encryption schemes is not easy, and it's strongly recommended that you do not attempt this. You'll soon see that writing the code for some of the classic ciphers can be quick and easy. For example, in the substitution cipher you simply swap one character for another, so the code is easy to write. But writing the code for modern ciphers presents several technical issues, such as generating truly random numbers, which may seem simple but are actually very difficult problems to solve. Most home brewed encryption programs will have weaknesses that will make them easy to break, leaving the data at risk.

   Luckily someone has already written encryption programming libraries for most systems. These libraries have been extensively tested so you can trust them to protect your data. The algorithms and code were tested and vetted by expert cryptologists and have been battle tested in real life use where hackers and attackers have had years to try and crack them. And if new vulnerabilities are discovered, these libraries will be patched and updated. If you want to write a program that uses encryption, make sure and use one of the built-in libraries.

   Going back to the lock analogy, don't try and build your own lock from scratch. You can get any type of lock you want from the Operating System; all you have to do is ask. So, don't drink and drive, don't run with scissors in your hand, and don't write your own encryption code.

   There are hundreds of examples of security holes caused by programmers with good intentions who tried to write their own code. If you want to read more about some of the more (in)famous failures check out the following:

   - BassOMatic which was the encryption scheme originally used in a product called PGP.

   - TrueCrypt which was a once vaunted whole disk encryption product.

3. **Not planning for security from the start of a project.** It's important for software engineers or developers, or anyone building a device that may handle sensitive data, to be aware that they'll need to include some cryptographic code to provide protection. It's critical that everyone involved is aware that the cryptographic code must be well integrated from at the initial stages of project, and not added on as an afterthought, or added after customers start to complain about security breaches. If the cryptographic code isn't integrated from the start it will have to be retrofitted, which may result in an end product that isn't as secure as it could be. For example, say you design a car but forget to add door locks. The door locks can be retrofitted but this may cause structural weakness in the car frame or the doors themselves.

What Cryptology is Not

While cryptology is the main feature that all cyber security is built on and can be an outstanding security tool if applied correctly, you also need to be aware that it's NOT the perfect solution to every security problem. Cryptology is only reliable if it's implemented and used properly. It's like the keys to your house or your car. They only secure your house or vehicle if you make sure and lock the doors when you leave. And keys will be much less effective if you hand out multiple copies to friends or neighbors or leave a spare in an obvious location like on top of the door sill.

A second point to be aware of is that most cryptographic implementations are not totally impervious. Symmetric ciphers can be broken if the key is ever reused. Asymmetric ciphers and password hashes can be cracked given enough time. The strength of the asymmetric ciphers and cryptologic hashes comes from the fact that while they can be cracked, it can take hundreds or thousands of years to accomplish. You'll learn the details of this later in the book, but you can see this for yourself by checking different passwords at some of the online sites

that check your password strength. Weak and short passwords can be cracked in a matter of seconds, but longer passwords can easily take hundreds or thousands of years. While it's reassuring that it can take thousands of years, notice that the password can still be cracked.

The last point you need to be aware of is that current cryptographic systems are great compared to previous systems, but they are not guaranteed to be unbreakable forever. In fact, every previous "unbreakable" cryptographic has been broken. Modern ciphers are based on some fairly complex math but periodically some genius, literal genius, figures out a way to break some of the encryption schemes that were previously thought to be unbreakable. If we learn anything from history it's that even the current schemes will probably fall someday.

In fact, it's predicted that the encryption and password schemes we currently use could be easily cracked by quantum computers. At this point in time, fall of 2020, there are quantum computers that can solve very small problems almost instantaneously. The limiting factor is the number of qubits that can be processed, but if this problem is ever solved then encryption performed on classic computers, which are non-quantum computers will become obsolete. Don't panic though. Cryptologists have already come up with schemes for performing encryption with quantum computers, which should also be unbreakable … until someone figures out a way to break it.

Pre-requisite knowledge

Modern cryptology is an interesting mix of history, linguistics, logic and statistics, computing, and math, lots of math. While I do my best to explain the important theoretical concepts in easy to understand terms, I also assume that you already have a base knowledge in these prerequisite subjects:

1. ASCII Encoding – You should be familiar with the ASCII table and be able to convert text characters to their ASCII value and vice versa.

2. Math – You should be familiar with binary numbers. You should be able to convert between decimal and binary, for binary numbers of 8 digits or less. You should also be able to do basic binary math operations such as adding two binary numbers or multiplying two binary numbers. It will also be helpful if you can convert between hexadecimal and binary.

3. Math - You should be able to divide two numbers and calculate the whole number remainder. For example, 17/5 has a remainder of 2. You may know this as the Modulus or Mod function, but no matter what it's called you need to understand this operation and be able to calculate remainders.

4. Boolean Math – You should understand how the AND and OR operators work in Boolean Math.

5. Math - You should be able to determine whether a number is a prime number or not and be able to factor small numbers to their prime number constituents. For example, if asked to factor 39 and find the prime constituents you should be able to determine that 39=3x13

6. Oracle VirtualBox and Linux - You should be able to build a Linux VM in Oracle VirtualBox and install and configure a Linux service.

7. Programming - You should be able to understand, and hopefully write, a simple program in Python, or one of the languages in the C family such as C, C++, C#, etc.

The reason I say that you need to know about these items is that I'll be using them to explain various concepts later in the book. If you don't have a passing understanding of any of the pre-requisite subjects, I suggest you do some research on your own until you gain a decent understanding. (I have links at my website, TonySako.com, to videos that explain many of these subjects in a fair amount of detail.) I could take the time to also explain these concepts, but it would make the book a lot longer. So rather than do that I'll assume that you're up to speed on the pre-requisite subjects.

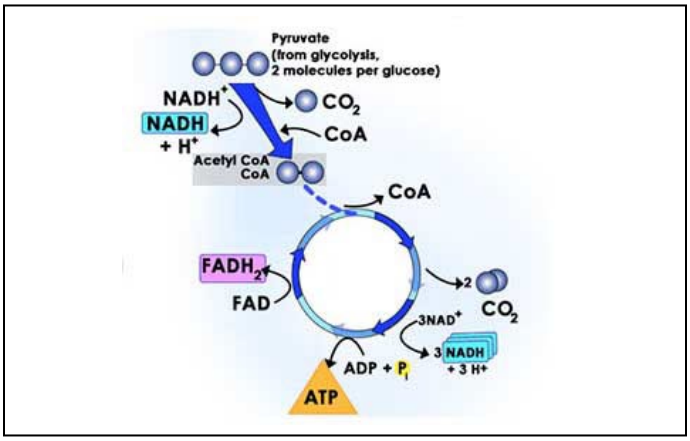
Isn't Cryptology All Math?

You've probably gotten the message that modern cryptology is based on math, which intimidates a lot of people, especially if they've struggled with math in the past. It can get even more worrisome if you look at any cryptography web sites or most college textbooks on cryptography which are chock full of math. The level of math used by professional cryptographers might seem simple to those who eat, sleep and dream in math, but for the normal person the math can seem complicated and incomprehensible. Since the math can appear intimidating, I'd like to offer you some reassurance that you can learn the necessary theory without the tortuous math.

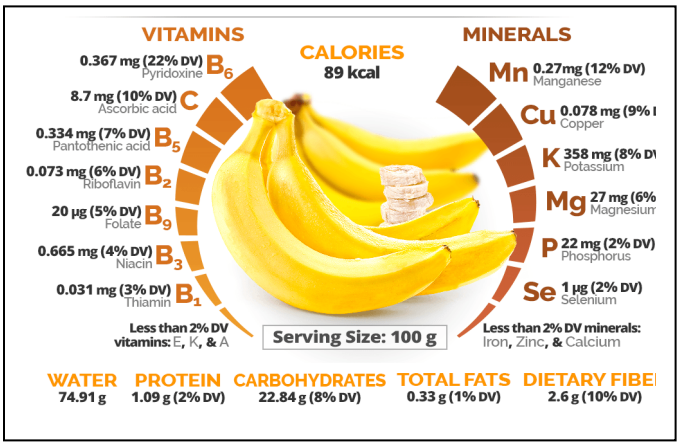
Here's another analogy that helps explain why professional cryptographers use math, and how you can still learn what you need to know without spending years in math classes.

Cryptology is one of those subjects that can be studied from three separate views, each with their own emphasis and their own distinct goals. The analogy that we'll use is comparing cryptography with human nutrition and cooking. When it comes to eating healthy there's a lot to know and studying healthy meals and human nutrition can also be done at three different levels, each with their own emphasis and goals.

At one level there are people who study human cellular biology and how macronutrients are absorbed and used by the different cells within our bodies. The people working at this level need to understand a lot of theory in biochemistry and cellular biology. Their goal is typically to use their knowledge to advance the science and gain a deeper understanding of how macronutrients are used by human cells or bacteria in the human biome. When cellular biologists communicate with their colleagues, they use the languages and symbols of biology and chemistry. Using these symbols and equations allow them to clearly communicate complex ideas quickly and efficiently. If you're not a cellular biologist this language will be hard to understand, but it makes perfect sense to other people working in the same field.



At the next level are the nutritionists, who take the theoretical knowledge produced by the people in the first level and use it to determine how humans will use various foods. They need to know a little of the cellular biology theory, but what they really need to know is how to apply that theory to decide the nutritional value of different types of foods. Their goal is not to advance the science so much as it is to determine what types of foods need to be consumed to achieve a healthy diet. These individuals make use of the theoretical studies produced by the cellular biologists, by putting this theory into action to determine what we should eat, and what we should avoid eating.



The third group of individuals would be those that actually cook food and prepare meals. They take the dietary recommendations produced by the second group and use them to develop entire meals. But the main concern at this level isn't how macronutrients are going to be used in different chemical processes in your cells, its cooking meals that are nutritious, healthy, and delicious. When planning and preparing meals, and trying

to ensure they're healthy, it's helpful to understand some nutritional theory. But the goal is not to advance the science or determine the nutritional value of each item on the menu. The main goal is to create meals that people can actually eat, and hopefully enjoy.

With modern cryptology the equivalent of the researchers at the cellular biology level are the individuals who use number theory and math to describe new algorithms for use in cryptography or cryptanalysis. Like cellular biologists, the main goal of cryptologists at this level is to advance the science. And like cellular biologists that use chemical symbols and notations to describe their work, cryptologists at this level use the language of math. The reason that math is the language of cryptology is the use of math symbols and formulas provides a succinct way to explain complex ideas and concepts. Since cryptologists working at this level eat, play, and dream in the world of math, they are all fluent and have no trouble reading and comprehending the meaning of complex formulas. Anyone *could* try to study cryptology from this perspective, but unless you're extremely proficient in math it's a difficult way to proceed.

At the next level of study in cryptology are the individuals that take the work produced by the theorists and convert it to computer code. This isn't as straightforward as it may seem, as computers have limits that don't exist in the theoretical world, and these limits must be accounted for. For example, the theory may call for a string of random numbers which is easy to produce in the theoretical world. But as you'll learn, it's fairly challenging to obtain truly random numbers on a computer. In any case, the goal of individuals working at this level isn't to advance the science and come up with new algorithms. Their goal is to produce functions or programs that implement the algorithms developed by the theoreticians. The programmers also use a specialized language to describe their work, although as there are several specific languages such as C++ or Java or Assembly Code that could be used. We could also try and learn cryptology from this perspective. Experienced programmers can probably understand the basics of the lines of code in each cryptographic algorithm, but they may not be able to see the bigger picture of what the code is accomplishing. That is, they might see the trees but miss the forest.

For example, the following Python code is used to implement the key scheduling algorithm (KSA) for RC4. You can probably understand what each line of code is meant to accomplish. But understanding each line doesn't shed any light on the overall purpose of the code:

```
j = 0
# set up an identity array S where S[0]=0, S[1]=1  from 0 to 255
S = range(256)
# walk thru S and shuffle the values
for i in range(256):
  j = (j+S[i] + K[i%n])%256
  S[i], S[j] =   S[j], S[i]
```

The last level of study in cryptology is where we learn to apply the theory and code. Like the meal planners and cooks, at this level the goal isn't to advance the science, it's to provide solutions that people can use. To be effective at this level you have to learn a little about the

theory behind the algorithms, and the limits of the code produced by the programmers. But you only need to know enough to produce a delicious and nutritious meal, or in our case use the cryptographic functions and programs that are built into every system to provide secure solutions.

I guess this is a long way to explain that you won't need a Master's Degree or PhD in math or be an experienced programmer to use this book to learn about cryptology. You will see some math and some code, but it will always be accompanied by an explanation in plain English. For example, here's an explanation of the RC4 KSA code. This code is essentially shuffling a set of numbers that range from 0 to 255 to try and randomize them. But it shuffles them in a way that can be reproduced in case we ever need the same sequence of random numbers again. Here's how it works:

1. It helps to think of the array as a deck or stack of numbered cards, which we'll call S Each card is numbered, from 0 to 255. We always start with this same stack of cards, but after they're shuffled, they will act as our random numbers.

2. This isn't in the code shown above, but you also have to get another stack of cards that we'll call the K stack. The key stack or K is the array that's referenced in line 6 of the code. This is a set of numbers that is supplied by the user and called a key. It's much different than the S stack, as the numbers won't be the same every time, unless the user provides the same numbers.

3. You start by stacking the cards in the S stack in order.

4. You're going to swap every card in the S stack with another card from the S stack, starting with the top card and working your way to the last card in the stack. To do this you do the following for each card in the S stack:

    a. Take the card from the S stack.

    b. Take a card from the K stack.

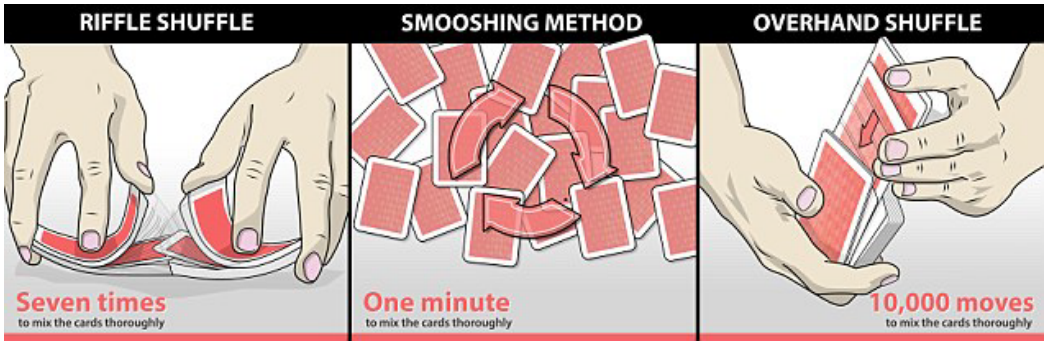
    c. Decide which other card from the S stack to swap with. This is done by taking the value of the S card and adding it to the value of the K card. Add the two numbers together. If the result is less than 256, then that's the position of the card you'll swap with. If the result is greater than 256, subtract 256 from the result. For example, if the result is 306, then you would calculate 306-256 equals 50, so 50 would be the position of the second S card. These calculations are guaranteed to always point to a second S card position between 0 and 256.

    d. Swap the two cards from the S stack. That is, take the card from Step a, and swap it with card whose position you calculated in Step c.

e. Put the K stack card back on the bottom of the pile.

f. Move to the next card in the S Stack until you hit the bottom of the stack. If the K stack runs out of cards before you hit the last card in the S stack, just recycle the cards in the K stack and use them again.

This is essentially a mechanical way to shuffle cards, and it would be really slow if you had to do it by hand. But it has a couple of important features. The first is that if you work your way from the first card in the S stack to the last card in the S stack, then you're guaranteed that each card will be moved at least once. The second is that the K or key stack controls the shuffle. If you provide the same key, you'll get the same shuffle. But if you use a different key it will result in a completely different shuffle.



Ok, that's an example of what I mean by a plain English explanation. The hope is that you understand the plain English explanation, or that at least it makes more sense than what you got from the Python code. And don't worry if you don't completely understand the explanation. It's just an example of the approach we're going to take to try and use plain English instead of relying exclusively on math or code for explaining the various algorithms. You'll run into key scheduling algorithms later in the book, and they'll make more sense when you see them in context.

Just in case you're curious about learning cryptology from the math perspective, here's a sample of the math and number theory you'll be missing. This is an excerpt from **Understanding Cryptography**, which is an excellent and highly respected book by Christof Paar and Jan Pelzl.

> **Why Are Encryption and Decryption the Same Function?**
> The reason for the similarity of the encryption and decryption function can easily be shown. We must prove that the decryption function actually produces the plaintext bit x_i again. We know that ciphertext bit y_i was computed using the encryption function $y_i \equiv x_i + s_i \bmod 2$. We insert this encryption expression in the decryption function:
>
> $d_{s_i}(y_i) \equiv y_i + s_i \bmod 2$
>
> $\equiv (x_i + s_i) + s_i \bmod 2$

$\equiv x_i+s_i+s_i \bmod 2$

$\equiv x_i+2s_i \bmod 2$

$\equiv x_i+0 \bmod 2$

$\equiv x_i \bmod 2$ *Q.E.D.*

The trick here is that the expression $(2\ s_i \bmod 2)$ has always the value zero since $2 \equiv 0 \bmod 2$. Another way of understanding this is as follows: If s_i has either the value 0, in which case $2s_i = 2 \cdot 0 \equiv 0 \bmod 2$. If $s_i = 1$, we have $2s_i = 2 \cdot 1 = 2 \equiv 0 \bmod 2$.

You can also watch hours and hours of Dr. Paar's lecture videos at his channel:

https://www.youtube.com/channel/UC1usFRN4LCMcfIV7UjHNuQg

Once again, math is the language that's used to describe what different cryptologic methods are doing, and to provide proof that the methods are sound. This is great if you can understand the language of math, and the details of the math being used. But if you aren't intimately familiar with math then the mathematical descriptions and proofs will be impenetrable, and any explanation they are meant to provide will be lost. It's like trying to read something that's been written in Klingon. It makes perfect sense if you can read Klingon, but if you don't read Klingon then you will have no idea what's being said. For example, do you know what the following sentence means? Here's a small hint, in case you couldn't guess, it's written in Klingon.

Any idea what it means? (I actually hope the answer is that you have no idea. I'm a little worried/impressed by anyone trekkie enough to take the time and effort to learn Klingon.) I'm sure you're curious about the sentence, so here's the same phrase written phonetically using the Latin character set:

```
loS poH pagh. ghaH'e' SoHbe'chugh loD Chuck Norris.
```

And the same words of wisdom translated from Klingon to English:

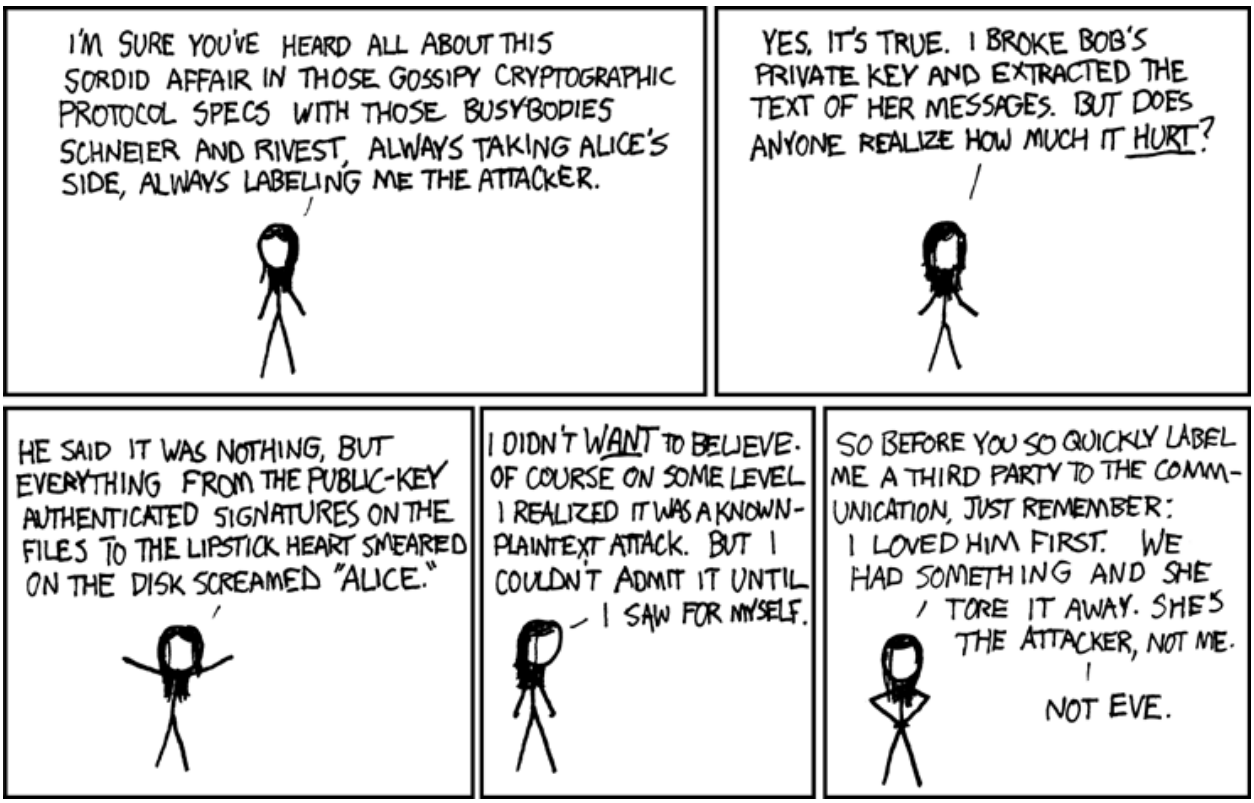
```
Time waits for no man. Unless that man is Chuck Norris.
```

The point of all this is that if you're a mathematician who wants to understand cryptology, and possibly advance the science of cryptology there are several books available to help you. In fact, almost every college textbook on cryptology/cryptography is based on the mathematical principles. But if you're not a mathematician, you need a book like this, a book that doesn't rely solely on math to explain the cryptologic algorithms.

Alice, Bob and Eve

Another introductory item you need to learn about are three people, Alice, Bob and Eve[2]. The same three people, Alice, Bob and Eve, pop up almost every time there's a discussion about a cryptographic system. Much like the way that most fairy tales start with "Once upon a time … " most classic explanations of a cryptographic system start with Alice, Bob and Eve in a bed. Alice is on one side of the bed, Bob is on the other, and Eve is in the middle. Alice wants to tell Bob something private but since they're all in the same bed Eve will hear everything Alice or Bob says.

Alice, Bob, and Eve aren't actual people, at least not the Alice and Bob we talk about in Cryptology. These names are just used because the first letters of the names Alice and Bob are A and B. And in most examples, it's easier to relate to Alice and Bob trying to have a private conversation than it is to relate to trying to send data between network nodes A and B. Eve is short for eavesdropper (Eve's dropper … get it?), and she's the person that's always sitting between Alice and Bob, trying to listen in on the conversation.



Cartoon from xkcd https://xkcd.com/177/

It always seemed weird to me that three people were in the same bed, a little Willy Wonka'ish. But that scenario was used to show that Eve could hear everything that Alice or Bob said. I think

[2] http://cryptocouple.com/

in most current examples that use Alice, Bob, and Eve they've found their way out of bed. But in any case, you will see these three names pop up a lot. And I imagine that these days the names might seem culturally insensitive, so feel free to use different names if you want. However, I'm going to use these names to stay consistent with the classic cryptography books.

Control of Cryptology

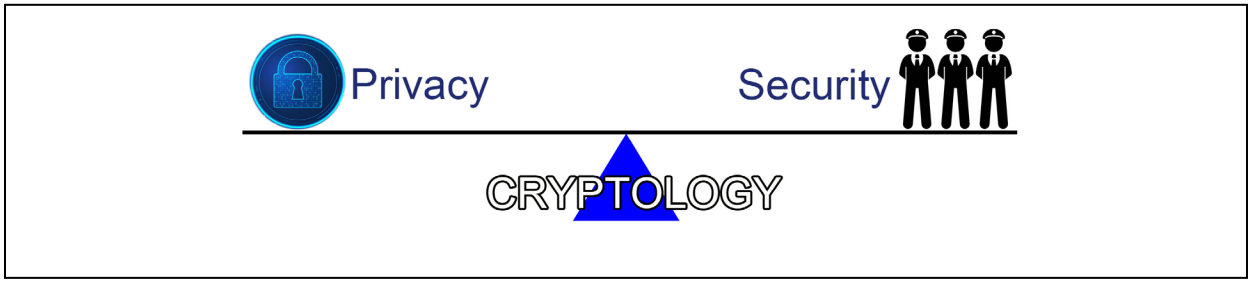
As you learn about the different types of encryption and hashing algorithms, you'll see that there are dozens to choose from. But even though there are dozens of available algorithms your choices are probably going to be restricted. The first restriction will be legal restrictions as there are laws that restrict which algorithms can be used outside the US. You should be aware of these laws and restrictions, but if the organization or company you work for has no foreign offices you won't have to worry about them too much.

The second set of restrictions are on a practical level, where you'll need to choose algorithms that work and interoperate with existing systems. In other words, you'll need to know what are the standard algorithms being used by other systems. There's one set of standards for the Internet standards, and a subset of these standards for working with the US Government.

In this section you'll learn who is in control of the laws and standards related to cryptology. You'll learn about two different aspects of control. That is, who has legal control over cryptology, and who controls the technical standards.

Legal Control

One of the purposes of this class is to help you become aware of social issues associated with cryptology. The issues involving cryptology revolve around society's need and desire to find the proper balance between the right to privacy and security. These issues actually might be more appropriate for a Sociology class, but since they deal with cryptology, they're very pertinent to this class. It's also pertinent at this point in time in modern society, both in the United States and in other countries, as we try to find the balance between our right to privacy and the need to protect our nation and our communities. You should be aware of these issues and the laws surrounding them, not only because they affect any work you'll do that involves cryptology, but they also have a tremendous impact on your personal life and your right to privacy.



On the privacy side, cryptology is a benefit because it allows people to protect their data and keep communications private which in the United States is an right granted to all individuals by the Fourth Amendment of the US Constitution. Ok, for you lawyers in the crowd, I know the Constitution doesn't explicitly say the words "right to privacy", but this phrase is commonly used as it's short and to the point. In any case, the fact that the US Constitution grants the right to privacy seems to indicate that anyone should be able to use cryptology to ensure their data and their communications remain private. And I believe most people believe that some of our data should be private, as we all know the dangers of sharing sensitive data like our birthdays and social security numbers.

On the flip side there are some secrets that are not "good", and the ability to keep things secret has enabled people to do some terrible things such as selling illegal drugs, exchanging child pornography, hiring people to commit crimes like arson and murder, or planning and mounting terrorist attacks. The people planning and committing these offenses can use cryptology to keep their actions and plans secret and thwart the organizations and agents of law enforcement tasked with stopping them. If there were no way to keep secrets, or no cryptology, it would make it much easier to detect and prevent crime and attacks on society. From the perspective of stopping crime, cryptology is a negative as it makes it more difficult to maintain security.

There are a few questions regarding cryptology that are at the heart of these issues of privacy and security. Should everyone have access to encryption and have the ability to keep their data or messages private? Or should access to cryptology be controlled? That is, is cryptology part of the right to privacy, or is it more like a weapon that needs to be controlled?

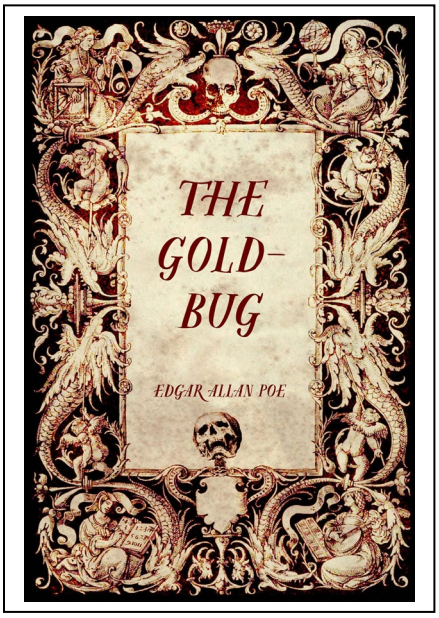
The answers to these questions could be either extreme, or somewhere in the middle. At one extreme, everyone would have unfettered access to cryptology. And at the other extreme only the government would have access to cryptology, it would be illegal for individuals, businesses and other groups or organizations to use cryptology.

Or, the decision could be somewhere between the extremes, where the public could use parts of cryptology or use all of it in certain circumstances. But if the decision is somewhere between the two extremes it brings up another question, which is who should be given the authority to decide the limits and restrictions on public use of cryptology?

These aren't easy questions to answer. In fact, they're quite difficult. And like any social issue there are consequences to any decisions. But decisions have been made, and laws passed that answer these questions for citizens of the United States.

We could just cite these laws, but I think you'll get a better perspective if you read the history of how the laws came to be. This history is a little convoluted, as even within the federal government there is tension between different branches and agencies of government. Some agencies are worried about protecting individual privacy and favor less government control

over cryptology, while agencies tasked with protecting society want more government control. The full story of cryptology in the US in the last century is pretty long, but it's important if you want to understand the relationships between the federal agencies. (You'll get a good portion of the story later in the class when you learn about asymmetric encryption and Public Key Infrastructure (PKI)).



If you go all the way back to the first recorded uses of cryptology, you'll see that it was almost exclusively used by kings and queens, or their military. A large reason for this was that most people couldn't read or write, and the rulers needed ways to communicate in secret. Through the centuries the general populace became more educated, and more people learned to read and write. General knowledge, and knowledge of some of the text-based ciphers became more widespread and during the 1800's the use of encryption made its way into several popular stories such as Edgar Allen Poe's The Gold Bug or Sir Arthur Conan Doyle's Sherlock Holmes and the case of the Dancing Men. Ciphers were used by the general populace as well, both for entertainment and for protecting sensitive information. There are several interesting stories with unsolved ciphers from this period, including the Beale Ciphers[3], and the Peralta Stones[4], although the Peralta Stones may be a fake.

During World War I and World War II new electro-mechanical systems were developed that significantly sped up the process of both scrambling and reading messages. These machines changed the course of the war, as they gave the Allies the ability to read the Nazi's encrypted transmissions, while at the same time protecting their own communications. However, the complexity and cost of the machines meant that the cryptographic machines were still mainly used by governments for diplomatic and military purposes.

After World War II the National Security Agency (NSA)[5] was created to continue the advances in cryptology by handling the development of encryption systems for the US government and military, as well as figuring out methods for breaking encryption schemes used by other countries. The NSA was famous for hiring the best and brightest minds in cryptology to keep the US at the forefront. The NSA controlled all cryptology in

[3] https://www.nsa.gov/Helpful-Links/NSA-FOIA/Declassification-Transparency-Initiatives/Historical-Releases/Beale-Papers/
[4] https://en.wikipedia.org/wiki/Peralta_Stones
[5] https://www.nsa.gov/portals/75/documents/about/cryptologic-heritage/historical-figures-publications/publications/misc/origins_of_nsa.pdf

the US by invoking the International Traffic in Arms Regulation (ITAR)[6] which is now called the Defense Trade Regulations. These regulations were meant to control the sale of weapons to other countries, but the NSA successfully argued that cryptology could also be used as a weapon since the ability to maintain secure communication was essential to our country's security.

The NSA also successfully argued that they needed to control dissemination of knowledge about cryptography to prevent it from being used by any other countries. That is, just writing about a new algorithm or speaking about it at a conference might make the idea available to people in other countries and impact the NSA's ability to read their communications. The NSA became infamous for their ability able to prevent anyone from releasing knowledge about cryptology into the public sphere by simply writing about it, or in some cases even publicly speaking about new or advanced methods in cryptology. When they found anyone with a new idea the NSA offered them a job, and if that failed the NSA threatened them with criminal prosecution if they ever published or spoke about their ideas.

So, at that point, the balance between security and privacy was at the extreme security end of the scale, at least where cryptology is concerned.

Even though the NSA had stranglehold on cryptology it wasn't much of an issue for most people in the 1950's. Up until the 1960's there wasn't much interest in strong cryptography outside of the government and military. But as computers became more powerful and more widely used the business community started doing their own research on ways to protect their data and communication. In addition to looking for ways to secure inter-company communication, some of these companies also wanted and or needed to build systems that could securely communicate with systems run by the government or military. This occurred when a business won a military or government contract for "secret" projects, and they needed to communicate with their contacts in the government.

And on another front, the government itself was looking for standardized cryptographic algorithms for their in-house work. When the government wants to purchase or build large systems of any type, they want to ensure that the components meet some standard so they won't be locked into a proprietary solution that forces them to buy everything from one company. So, both business and government were asking the NSA to come up with some standard encryption that could both use.

The government agency that handles setting these types of standards is the National Institute of Standards and Technology or NIST[7]. NIST put out a request for proposals (RFP) for an encryption standard and finally selected an encryption scheme known as the Data Encryption

---

[6] https://www.pmddtc.state.gov/ddtc_public?id=ddtc_kb_article_page&sys_id=%2024d528fddbfc930044f9ff621f961987

[7] https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines

Standard (DES). Many people have noted the suspicion that the NSA had a hand in NIST's selection of DES, and possibly forced NIST to select it because they were able to crack it. You'll learn all the excruciating details about DES later in the class, but for now the important thing to note is that it was the first encryption standard approved for use by the US Government.

During the 1970's and 1980's the computing landscape changed dramatically with the advent of the Internet and the use of personal computers. Computers were no longer large, expensive items that were only owned and used by businesses. They were now owned by everyday people, and many of these people wanted to use the Internet to make secure purchases or send private communications. The desire to shop securely on the Internet may seem laughably quaint now, but in the early 80's it was impossible to do. Even though NIST allowed public use of cryptography via the DES standard, it was impractical for home use. This is because DES is something called a symmetric cipher which requires the sender and receiver to exchange an encryption key for each transmission, and the key exchange itself is insecure unless it is also encrypted. (You'll learn about symmetric encryption and the key exchange problem later in the class.)

During this time period there were a few individuals outside of the NSA who felt that everyone should have access to cryptology. A few people in this crowd spent a lot of time and brain power coming up with another type of encryption, called asymmetric encryption, which solves the key exchange problem of DES and symmetric cryptography. The methods they developed had the ability to bring the protection of encryption to the general computer using public.

Even though asymmetric encryption had the promise of allowing the general public to encrypt their data and communications it had one big problem, The NSA tried to prevent asymmetric encryption from becoming public because it was so good that they couldn't crack it. From the NSA's point of view, the danger of asymmetric encryptions was that if other countries started using it, they would no longer be able to read intercepted communications. The loss of this ability would essentially blind them and be a huge blow to national security. The NSA was able delay the public release of asymmetric encryption for several years, but despite their efforts the new system was eventually released. The systems that use asymmetric encryption have evolved into what we now call Public Key Encryption (PKE) and the Public Key Infrastructure (PKI) which among other things allows you to use HTTPS to jump online and buy that new thing you need from Amazon and get it shipped to your house tomorrow. Even though the NSA lost their fight to ban public use of asymmetric encryption, they were able to place restrictions on the key size for any software that used asymmetric encryption that was to be exported or used outside of the US. (As you'll soon learn, smaller keys make messages easier to break, and larger keys make it more difficult.)

The adoption of asymmetric encryption caused a major shift in the balance between privacy and national security. It went from being almost totally in favor of security to the other end of the scale and being almost totally in favor of privacy.

From the early 1980's to the present the NSA essentially lost its ability to control the use and dissemination of encryption. Although the NSA still has the legal right to control cryptology, the ability to download software on the Internet makes it almost unenforceable. All you need to do is post code on some Internet server and it can be downloaded and copied to computers around the world, so there's no way to control dissemination. Currently the NSA can still use the ITAR to enforce some restrictions on encryption, especially for products that will be used outside of the US, but it's essentially given up trying to restrict use of cryptology in the United States.

And in essence this means that the right to privacy advocates won the battle over the use of cryptology. At least to this point.

Of course, the right for anyone to use cryptography hasn't come without consequences. The widespread use of encryption has made it more difficult for law enforcement or security agencies such as the NSA, FBI, CIA, etc. to perform their jobs effectively. None of the law enforcement or security agencies were happy about the new developments. And there have been many high-profile cases where the inability to read emails or text messages have come at the cost of human life.

The law enforcement agencies periodically make attempts to rein things back in. There have been several laws proposed that provide back door access with the use of systems like the Clipper chip or requiring vendors like Apple or the phone companies to provide them access to user passwords or user data. But so far, at least to this point in 2024, none of these laws has passed in the United States.

While the balance between privacy and security currently remains in favor of privacy, I predict there always will be a fight to find the correct balance between individual privacy and the ability to secure and protect society. And outside of the United States, there are several countries such as China and Iran that have found ways to prevent public use of cryptology in the name of state security.

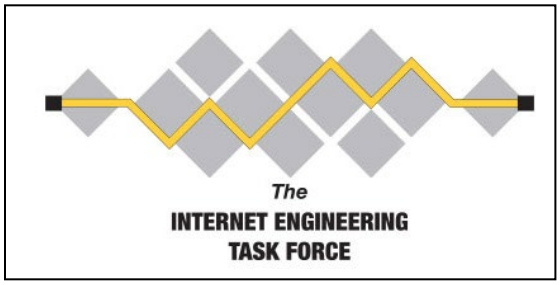
If you want to read more about cryptology and privacy you can check out the following web sites, or read Steven Levy's *Crypto*, which is an excellent book on the subject of how encryption had to pulled out of the hands of the NSA.

https://www.oreilly.com/library/view/web-security-privacy/0596000456/ch04s04.html
https://www.technologystories.org/regulating-contested-reality/
https://www.nap.edu/read/5131/chapter/19#418

Technical Control

In addition to legal control of cryptology, there are a couple of groups controlling the technical standards that you should be familiar with.

IETF



The first group that sets and manages cryptographic standards is the Internet Engineering Task Force (IETF), which decides which technologies will be Internet standards[8]. The IETF is the group that sets and manages standards for the Internet. For example, the IETF selected TCP/IP as the main network protocol and addressing scheme for the Internet, or HTTP and HTTPS for web browsing. The IETF isn't dictatorial about their standards, but following their standards is the only way you can be guaranteed that your network traffic will work with the rest of the Internet. Being a published Internet standard has a lot of weight with everyone on the Internet, which means there's an excellent chance that it will become widely adopted.

Setting the encryption standards on the Internet has worked the same way. The best example is with HTTPS, SSL, TLS, and PKI which allows secure communication between web browsers and web servers. You'll learn the details about these technologies later, but the important concept to take away at this point is that everyone involved with the Internet and encryption has agreed to use these standards. This includes every Operating System, every browser, and every web site. I don't know if you can imagine the chaos that would ensue if each browser vendor, each web server, and each Internet user got to choose their own method of encrypting data. There could be hundreds or thousands of different encryption schemes to choose from, and each time you made a secure connection to a web server you'd have to select or negotiate the correct scheme. The actual encryption methods we use today were proposed to the IETF, which selected the best technical solutions and then published them as standards. Once again, using these standards isn't a requirement, it's just a suggestion. But if you want your systems to connect to other systems your best bet is to use the standards.

The IETF publishes their standards in documents known as RFCs. This stands for Request For Comment, which may seem like strange name for a standards document. But the first RFC was written by a man named Steve Crocker who chose the name carefully[9]. He was part of a group of younger people who documenting the technologies that should be used to connect to the original small network that eventually grew into the Internet. They needed cooperation from a group of older, established operations, and wanted to avoid sounding dictatorial and possibly discouraging cooperation. So instead of titling their documents Standards, they called them RFCs. In any case, there are RFCs for every Internet standard and each RFC has a unique number. For example, the RFC for FTP is RFC 959 and the RFC for HTTP is RFC 7231. One last thing to note about RFCs is that the information in an RFC isn't meant to be used as a general tutorial or general explanation. Each RFC contains the technical details required for implementing the standard and the information can be a little difficult to digest. That is, each

---

[8] https://www.ietf.org/standards/

[9] https://www.wired.com/2012/05/steve-crocker/

RFC is more like the schematics for a car engine, which isn't super helpful if you're looking for something like a driver's manual.

NIST

The second group charged with setting standard in cryptology is the National Institute of Standards and Technology or NIST. As mentioned above, NIST in charge of setting standards for the US Government. And in the case of cryptography, NIST decides which encryption algorithms will be the standard for computers used by any government agencies, or computers or networks that communicate with a government agency. This includes any systems that interact with the financial and banking system, so it covers quite a few systems in addition to the obvious ones. While some people are suspicious of NIST as it's a government agency, NIST actually provides quite a remarkable service when setting cryptologic standards.

In the case of cryptology, NIST doesn't write any of the cryptologic algorithms themselves. Back in the 1960s and 1970s NIST asked the NSA to develop standards or review proposed standards, which led to distrust and the suspicion that the NSA gave the NIST algorithms that contained a backdoor[10]. To allay these suspicions, NIST moved from asking the NSA for help with algorithms to asking the cryptologic community to help develop standards. NIST starts this process by publishing something called a Request For Proposal (RFP) which has the specifications of the item they're looking for. Interested individuals, members of business and industry, and individuals from academia can respond to the RFPs by submitting different algorithms for consideration. NIST then evaluates each proposal by letting anyone and everyone inspect and attack the proposed algorithms. During this process, which can take months or even years, the experts will look for flaws and try to find weaknesses or ways to break the algorithms. This results in NIST standards that contain battle tested algorithms, and they're tested at a level that would be almost impossible to recreate in any other way.

Once NIST settles on a standard, they publish the technical specifications for the standards in documents called FIPS (Federal Information Processing Standards). The FIPS document contain background on the standard as well technical details about any associated algorithms. NIST also publishes documents called SP, for Special Publications, which contain technical details about related algorithms or technologies.

You can find a list of the NIST Cryptographic Standards and Guidelines at:
https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines

---

10 https://en.wikipedia.org/wiki/Dual_EC_DRBG#:~:text=In%202013%2C%20after%20the%20New,their%20customers%20to%20switch%20CSPRNG.

Summary

Here are the important points you should take away from the discussion on who controls cryptology:

1. The NSA has the legal authority, through ITAR, to control the use of cryptology in the United States. Currently, they have removed all restrictions on encryption for domestic use, but they still have restrictions on encryption systems that may be exported outside of the United States. However, controlling the export of ideas is extremely difficult as the Internet makes disseminating information as easy as clicking a link.

2. The IETF controls the Internet standards. Any cryptologic system that is expected to work on the Internet should follow these standards to ensure interoperability. The cryptologic systems approved by the IETF have been well tested, and all the various components in the PKI system function together to provide end to end security.

3. NIST controls the US Federal Government standards. Any system that communicates with the Government, or by extension with the financial network must follow these standards. This may restrict your choice of cryptologic solutions to a subset of the standard solutions approved by the IETF.

How the Book Is Organized

Now that you know why cryptology is an important piece of anyone's cybersecurity education let's talk about how we're going to approach learning about cryptology and how the book is organized.

We could just jump in and start studying modern cryptology and looking at the encryption schemes currently in use. In fact, most of the college textbooks that I've read do this. We could follow suit and just jump right in and start talking about XORing bits and Elliptic Curves. But as it turns out the modern algorithms are the result of a straightforward evolution. The evolution started with ciphers that are easy to understand from a human perspective, then became a little more complicated with each step. So instead of jumping to the current step in the evolution, we'll look at the cryptographic ciphers and algorithms at each of the step in the evolution. This incremental approach makes it much easier to understand what is going on with today's encryption algorithms.

Another benefit of taking this approach is that it allows us to look at some of the historical events that have been impacted by cryptology. The history of cryptology is different than any other subject in cyber security because humans have been using cryptology for thousands of years, as opposed to the rest of cyber security which has only been around for the last 40 or so

years. And cryptology has been involved in many interesting historical events, changing the course of wars, and bringing about the downfall of historical figures.

There's no official timeline for cryptology, but I've broken it down into the following sections for this book. While we'll use the historical approach to make the current methods easier to understand, the majority of the class will be used to cover current methods.

1. Historical manual methods for hiding data.
2. Early Electromechanical
3. Cryptography goes Digital
4. Modern – Symmetric Stream Ciphers
5. Modern – Symmetric Block Ciphers
6. Modern - Asymmetric
7. Modern Hashing
8. Modern PKI
9. Current/Future?

Optional Reading

If you want to go further into any of the subjects in this chapter here are some web sites you might want to look at. You can obviously do your own searching as well … But, just be careful not to go too far down any rabbit holes.

https://simonsingh.net/The_Black_Chamber/index.html This is the web site of the author of the book. Like the book, it contains a lot of entertaining information.

https://www.britannica.com/topic/cryptology
https://crypto.interactive-maths.com/introduction-to-cryptography.html
https://crypto.interactive-maths.com/steganography.html
https://crypto.interactive-maths.com/codes-and-ciphers.html
https://www.sciencemag.org/news/2020/07/biggest-flipping-challenge-quantum-computing

https://www.khanacademy.org/computing/computer-science/cryptography - Khan Academy
https://en.wikibooks.org/wiki/Cryptography - wikibook

https://www.youtube.com/watch?v=yy6TV9Dntlw - Quantum computing explained with a deck of cards

https://www.engadget.com/2019/07/31/how-ag-barr-is-going-to-get-encryption-backdoors/ - Should the government control encryption?

https://www.forbes.com/sites/kalevleetaru/2019/07/26/the-encryption-debate-is-over-dead-at-the-hands-of-facebook/#1e8c862d5362 - Grabbing data before/after it's encrypted