### security\_week2\_2-20240910

说话人1 00:00  
In some parts of the direction back there. Good. So why do they talk to that? So substitution suffers for several 100 years with the way to include information. So in the 8th century, we had some islam culture in terms of civilization. And at the time they were very advanced in science and mathematics, it was considered to us that society i'm since topography was quite widely useful matters of state, lots of scientists, mathematicians, researchers, stupid interest in cryptography, but in making the code better and breaking the code.

So even back then, they found that security is not only one discipline, whether they did. They make like mixed different disciplines together. They could find a way to very effectively, right?

Two survivors, but they make concepts of that is the statistics and also the mistakes to do this.

So when they started, they were just takes before up. They notice that some of the letters appeared much more often than others. Right? And this holds for most languages. All right? In english, for example, the letter e and e are quite often that we never use resent for x or q right? They're offended this process as frequency analysis. If you get statistical analysis of the letters, you can find patterns and behavior that allows you to extract messages from substitution part. Right?

For example, this is the average data frequencies being engaged language from a to z you'll see that e almost 30 % operators are e 9 % of data from t as opposed to q and x and z that almost never appear. Right? This is not a hard rule. So this does not mean that he will always be the most, or they will always be the least in a sentence or a passage or whatever. Right? This is a general rule. Right? The interesting thing, they then link the substitution side areas. You have to remember that it's a later to later mapping. If we have the substitution alphabet and the e maps to ar for example, if the e is the most common letter in the flight, xr will also be the most common data in the ciphertext.

In other words, if I look at the ciphertext and there's a lot of rs, then I must think maybe this r is most likely to be an e or maybe at or maybe a right?

If there is some cyber takes letters that occur, maybe only once in a very long sequence of message, then maybe we can think that could be aq or x or az the main weakness of that identified was that the substitution suffer does not break up the pattern of the underlying context.

If you have patents and statistical behavior of the plain text, the same statistical behavior patterns will occur in the cyber text just for different data. Right? In addition, many languages, so we look at english specifically have other things. In english, the most common double letters are se e is l is o right? There are two single letter words, which is a or I there's also lots of other two letter combinations of two. It is be as so we 541, or 2 in that order of relevance of is the most common one. A is the most common one. U is the least common one, who is the least common one. All right? Then there are some three letter words. If only two 3 letter words, they are commonly d and add that opportunity to any. Readers do it like the most common feeder to versus d and add. And then there's some other funny things.

So in the english language, we have an e the letter h very often goes before that e but it will never go after it. You can manage to find out which one is the e if there is enough cipher, text letter that often goes before that plane takes e then you can have a good days. And that is probably age. Right? Another thing that could happen is if some of the plain text information, if some other message that was sent, you already know.

All right? It might be that every single morning this person sent a message and started with good morning or in the good evening or at the end says, bye bye. Then that a couple of the, you already know a little bit of the message. That means you do pretty much recover the context it is for that little biblical message. So that's a word that is called a crap. Let's look at an example. If we have this cipher decks, then we cannot obviously see what it is. All right? And we cannot vote for searches, because we know it's going to take us a very long time. Why we can use frequency analysis. If you quickly look at this message, do you see any particular subjects in there that appears quite often? All right. So one should come out as one that appears quite often. That is the later head. We use our frequency analysis, and we said the most common letter in the english operator is e we make a guess, and you can say ok the k might be an e we can map e to k we also see that there is a single letter word here.

So we know that should be an I or a so maybe we can try it as a now, we can basically do the substitution.

So now we can start again this, and it's around. There are some ease around. Right? Next step, governor along the line, you see that the cipher takes me there in appears commonly before e so why do you think the a is h because he said h have this funny pattern where it's often before e but it's not really after the end becomes h now, we started to see some work right now. We see some up here.

So what is the most common word that we're gonna start? A message with? L their idols are going to say hello. So you can say that are is r and then this is coming here. Hello. So now we have number of letters already. Right? We have some oscillators. Now. I can, for example, take this one here. We can map it out. So now it's like doing prosperity also. Right? If you have some of the letters, but he's trying to figure out what the rest of the letters are. So these guys, why do you think the word is? Example, right? Because we are doing example, actually written up here, the example as an example, we can map it up. Right? Now, what do you notice about the data is done here? What is interesting about them? They are not for medical order. Even if you only have partial of them, they're all in alphabetical order.

R is you need, we are probably dealing with shift buffer right now. We have enough information to know what the shift is. The shift is six, right? And you can have an alphabet, and we can decrypt the message. Then idc how easy it is. So we can either give it to a computer or a huge server room. And wait 4,000 years, we can use this very simple method as it is. Okay? Different methods, different resistance to different kinds of attacks, frequency analysis against substitution sites are very powerful.

So now you also know that the key is six, and you can basically equivalent message. You want going between others involved in this particular areas. Right? I think I also put on canvas, but if you really want to try this, this was fun. And you can try to purpose basis. And if you manage to do it, you can send me email and I will tell you if it's correct. Okay. We're getting ready for it is just the point. If you think this is the thing is fun, you can try to encourage this one. And see you can do it in about half an hour, I think, just using 3 . 4 systems.

Okay. How do we beat frequency analysis? Remember, we said that this is like always be like this. Ii try to protect things. People hacked it, I find it better. People say frequency analysis, simple substitution. Soccer is pretty powerful. What can we do to try and fix it? So people say we can omit the spaces, makes it easy to see where it is. We can deliver it in the spells of words. We can put in some random characters that are not in the alphabet. It doesn't have meaning. We can also use codes. So codes have turned it almost into a at two iterations suffer as we'll go through the symptoms. We are substitute code words for real word, for real words, for code words.

And then we'll do the substitution. Each time it made it a little bit more difficult to do the frequency analysis. At the moment, people figured out what was going on, people who basically just at that frequency analysis to solve the problem. So generally, people were looking for a better way of doing substitution sides. All right? So let's lead to the next big, advanced photography, which was using more than one alphabet to encrypt the same message. So in other words, having a poly alphabetting substitution, right? Around this time or around the 16th century, the near came up with the cyber.

The near was a quarry alphabetic, cyber. That means that he broke a lot of the pattern that happened with conventional substitution cycle.

So the way the veneer cycle works was you can have the same fine takes that they're mapping different socrates stickers. And you could even have a single sockets that conducted a a single good letter, I think, multiple ones. And you can have multiple ones mapping the same one. Now, you basically break up the pattern.

He uses multiple alpha bits. However, the alphabet is slowly chosen in some systematic way. Otherwise, it doesn't work. So either the near side of it, the first thing you did with a video site where you chose your key, right? Also, in this case, the alphabet is not the key, but it was the substitution cipher. You chose a key value, which is most likely a good combination of it. In this example, we have five alpha bits. Our finger is holy. What happens is now we'll have five alpha bits, and each one of the alpha bits will start with the corresponding letter of our keyword. So we have our plane takes up a bit and use it. Our first cycle takes alphabet starts with ap the a is e then we have the alphabet. Second a then one starts with au 4th one starts with an l last one starts. Good, I right? So either we do encryption, so we want to encryption message.

Hello. Right? And we're gonna use basically separate alphabet to be true. Each single one, we're gonna use the first operate to improve h second one l big one l plus one l 5th one o okay? So if the message was larger, we have five outfits. If the message was larger, then we'll just start repeating using of the outfits. Then we just extend the code word and the corresponding alpha base for longer. So we have, however, it seems, is the same thing as our keyword. So we can use every single alphabet once. We wanted to extend it to have a ball, we'll start using p and o and you to get so. Right. So encrypting, the first one would be using alphabet p and we are encrypting, the h we will have aw then for the e we're gonna use the a alphabet, so we're gonna get an e for the first owl, we're going to have the offer. They use the best if.

Then we're gonna have the second l we're gonna use l alphabet. It's gonna be w and if I is getting because aw so iec we have encrypted hello is wefw and w so immediately you can see while that happened that we had in hello before that we had in our example doesn't exist anymore. Because the first l maps to an f the 7 second l maps to aw we also see that we have three ws at the side of x then we have three different latex values, h l and o basically mapping to that w you can see that this is quite effective at breaking up statistical properties of the fight text with relation to the psychotics. Right? One of the things is we incited the two needles, lfw and we've incited hl and o all the w now you were dealing simple basic data frequency with patterns in the english language. It's going to be very difficult because you assign the text is not very effectively distorted with regards to the underlying context. Right?

So veneer you can see is slightly more complicated, because now you have to deal with multiple alphabet In a way. The key sharing is quite easy, because you have code word rather than sharing pile of it. It takes some time to encrypt and decrypt, but it's very much stronger than simple substitution. Right? So how did they break this in your cipher? Indeed? So for a very long time, for almost 300 years, the nearer seen the site of the architecture. Right? The only weakness it had was if you could figure out what the length of the code word was, because if you could figure out the length of the code word, you would know when the multiple alphabet is repeated.

And then you can figure out which characters was basically encrypted with the same alphabet. You could still apply frequency analysis, right? In 1863. So for a long time, they said it's quite difficult to choose the code word. Unless somebody leave it to you, or it was a real word, or you had some other information. So the first person that found a universal way of doing this without knowledge of the key word, but it could possibly be was this is key.

The consistent method said that inherently, just on average, we are gonna see repetitions in the psychotics. The repetition is gonna come from the fact that we just so happened to be encrypted the same plain text with this same sequence of alphabet. Right? If we, for example, have the key word, abcd we are trying to encrypt. Critter is short for cryptography as we go through the alphabet. So this plane takes with abcdabcd we see that in the cipher text, seven parts of the cipher text repeats, because this just so happens that here this right things value maps up to exactly the same alphabet used to be encrypted.

The repetition distance is 16, which means is 16 spreads 60 meters between the start of the first sequence of the same level sentence. Therefore, we know that the size of the key must be a divisor of that 16. So it could be 16, it could be a it could be four, it would be two or it could be one. In reality, there's gonna be multiple repetitions. All right? If we have multiple repetitions like in the bottom example here, the distance between the first vhvs and the second vhvs is 18. We know that the length of the code is eighteen nine six three two o one. The distances between the first quce and the second qucc is 30. So we have, 10, 65321. And we look for the union of the intersection between these. And the key size is most likely 632, or one. So this is probably for sure that the key size is most probably six. And once that the keyword is six and you're dealing with six alpha bits, then you can essentially go back to using original frequency analysis to solve the problem.

Okay? After the near was broken, the nearest type was still lived on for almost 80 years.

All right? So the last very famous version of what we can think is known as a real body of a baby cipher. Was the enigma machine? Has anyone had that e linkedin machine in the second world war? Germany was one participant in that war. And they encrypted all the communication. Using this enigma machine. The enigma was a following alphabetic substitution. It basically involved both sides of the communication having the same machine set up in the same state. Essentially, it looks like a very big old typewriter, and you have seen the typewriter. But basically, this one is opened up. So you can see the rotors and you can see some little lamps there. On the cover, it would have the alphabet printed. If you wanted to encode a message, you would turn these rotors. You would set up some permutation on the front, swapping some letters around with the table. And then you would type a message.

So let's say a you would press the a and the letter it would back to would be lit up by the little lab. It will show you the cover which way it's supposed to have two.

So I did this work. You needed an instruction book that went with it. The instruction book had a code, so that code could have been a monthly code or a weekly code or a daily code. It said that if you want to send a message on the 10th of september, you need to configure your machine in this way. Configuring the machine, as I said, on the front was a blackboard with cable.

If you wanted to swap between a and l you connected with the wire, the value a and the value l that would swap iran. At the same time, you have growth as at the top that you can turn, and they have to be saved specific values. These machines came in various regions. If I remember correctly, the only one was quite simple, so it had few returns, whereas the mainly ones that was made for boat or submarine, you vote that I regard for a long time, I had the more secure 5021.

Each one of these vectors have to be saved the same. Because and then I would say mine exactly the same as the recipient, right? About to do with the book. Look at the date, set it exactly the same. We will be able to send data to each other. This is just showing what the the platforms in front. So I just had a wire with sort of like with a stick that you stuck in and you can swap your datas around. And then these are the rotors. You can send them to a specific configuration for the right. The liquid machine had approximately 2 to the 60 feet different permutations. I have many different states. The allied forces spend a lot of time trying to do cryptanalysis on this machine.

If you have opportunity to travel to uk one day and you really like history, you can still visit a place called messy park. So messy park is where everybody went to work specifically on evening from the sheep. Right? They try to get all the smart people and mathematics in computing at the time together there to all work on decoding in different messages. Right? So they were working on ways to reverse engineer the cipher. And if they could figure out the cipher to figure out what the message was, right? They even brought these special things for bombay machines, which were essentially little computers to emulate the different machines.

Each one of these machines could simultaneously emulate 50, 60 ° machine, so they could configure it in different configurations and check or drive route for search for what specific configurations that gonna she had on that day.

So they could get the message back. Right? So some of the cryptanalysis they did was aided by very good mathematics, very good computer science. A lot of modern computer science has its origins in black teapot. The man that was most famous working there was added hearing. He's considered like the father of computer science and it drove a lot of things into article efficient machines coming up with this computer system. A couple of years later. A lot of that came from all the efforts that we could just reverse it to hearing the unknown machine.

Despite all of this very fancy stuff, some of the reasons it made easier, was it the machine? The people using it had certain operations efficiency. So one funny thing about the machine was that the same letter could never encrypted the same letter. If you press a it will never reading for food and a so that thing you can already reduce one of the possible elements. The other one was that often the people operate in the machine would be too lazy to update the code. So then a friend would just decide, well, too much effort to update the code every day. So we're just going to use the same code for like 6 months set of 1 month. You can use it for 40 that set of 1 time.

The other thing was trips that we spoke earlier about trips. Drive trips were no plane takes messages that we know her somewhere in the message. So very often these messages were sort of quite simple. Maybe it's a weather report. So we just say it is sunny today. We can also see where the basic percent from a sunny today, so then we can guess the basis probably say something about it being sunny. A lot of the time they had an operational rules, say you have to say the message even if there is nothing. So the message says nothing to report right in german. But I every single day you saw exactly the same message that said nothing to report, nothing has happened. So they already knew in some cases what the context was and that of the problem. Right? So this takes us up into the 1940s. Right? I think my machine very famous, lots of situations on, then we had a huge departure.

So up to now, since roman times, everybody was looking at substitution side, we wanted to say text messages. We will start with the roman alphabet. Everybody that worked on crypto was both sort of caught inside this box. We want to see text messages. We want to have a substitution cipher. The secret of having a good cipher is having lots of the permutations of text play off, getting writing of frequency analysis.

In the 1940s, things change a little bit. All right? So suddenly people started thinking, hey, now we have radio communication. People started thinking about how to encode data is text really the best way to encode data. What happens to the rest of the world that doesn't use the roman alphabet, right? How would they send messages? Are we going to start representing things with my free data? Right? Especially if we want to send it radio? Right? Now, how does this work? For all substitution parts? I guess we need something significant in the right at the time as we were working with now digital with the emergence of digital communication, radio, telephone is very useful. Moscow still takes place. Right? People started studying information field, right? And the biggest name in information period, the time was a man called for china. And he got interested in encryption and secrecy and confidentiality. He went at the problem from a theoretical perspective of if I could solve the perfect encryption algorithm, what would that look like? He said an encryption algorithm should provide us with when he turned to be perfect secrecy.

So the definition of perfect secrecy is even if the attacker had infinite resource at infinite five, you would not be able to break the cyprus.

In other words, even if he's able to brute forth, all the keys, all the combinations, essentially, the attacker should not have any additional information at all about what the plane takes possible to repeat. Right? China's solution for this was called the time path, the one time path.

As we look at it, it's quite a simple concept. We're gonna make bank tax. Our plane decks is gonna be binary. Now we've moved away from having alphabet to having a binary values. Our message is assumed to be binary 0 ~ 1. We are gonna generate a key. That is the same length as the plain text. We are gonna do a binary operation for the next door on it to get our side of the text. For those of you that have not had an excuse for x or so x or is a binary operation. To combine single bits together, you can almost think of an x or as a difference function. If the two binary values are the same, the answer is zero. If they are different, they are one. If you go one, x or y is zero, zero, x or zero is 01, x or zero is one. Zero is equal to one. Is that one? If you want to do, what do you think the difference is one? If they are the same? Is zero. Therefore, we can have our fate x you can have kind of alice. We can convert it to a vibrant value.

These are basically ascii codes of the extra decimal representation of each nature. We can make a random key, which is also a binary value. We don't write the binary because it will take us very long. So we write them. So basically vision, and we explore it together and we get the cipher dance. So this is essentially what we want to do. Then to decrypt, we just do exactly the same thing. We take the cipher text, we exhort it to the key that gives us. The bank takes back. Why does that work? An xo function has another special property, which comes from the fact that it's a different function. If I explore the same thing with itself, we can end up with a zero, because everything is going to be the same. Every single result is that he is zero. If ix or something to a zero, it's a space to say, because the 0 and 0 ~ 0 and 1 and 0 ~ 1.

What essentially happens here is we have latex x or with the t Which is the cipher text. We decide to text export to t it's the same as same plane takes x or p because thats the cycle things. Good. All xo to the p we get exo to p which makes it 0, 19 xo to zero. Thank you. X or is commonly used in topography for this reason, because it has a good way to go from subjects to matrix, is reversed the subjects to matrix. Yeah. Right. The only problem with this was, shannon said to be truly secure, the path, the key value should be truly random, and it must be used only once. In practice, this is quite hard. But we'll get back to that. So why is this good? Why was sometime by design group? In this case, we have the same size mistakes received earlier. This was our size that takes it back to $100. I could try a different key value, but I get a totally different message, but the different message also makes sense.

So instead of saying hello, alice, and I says, bye, bye drop.

Okay, we have the same cipher text. Somebody try to decrypt it using another key. They got a different plain text and then different plain text always makes sense. This is what makes the one time fat stronger and what shannon thought, the perfect secrecy, if an attacker would come along and decrypt the message, the attacker is so powerful that they could root for search all possible key values. Then the attacker will also find all possible answers. If the attacker finds all the possible answers, the attacker has no idea which answer is the correct one. Right? Because he's going to have every single one of the answers. He's fine. He's going to be the correct one. But how does he know? That is the correct one? Doesn't know, right? It can be equally likely as any of the other messages that the attacker found.

In other words, it gives us perfect secrecy. It's actually the only cryptographic mechanism in terms of encryption that gives us theoretical security or perfect secrecy. One contact cannot be happy. Right? Unfortunately, for us, it's also completely practical to use. Right? We mentioned on the previous slide, refer it to work properly every single time we need a key, the same length as the message. Every single time this key has to be completely new and random. If we think about practically sending information that is basically impossible, i'm trying to securely send you a message. I make a new key that's completely random, the same length of the message.

And now somehow I have to give you that key value. I can give you that bad value. How do I do that? If I had some tool that allowed me to give you secret things that are the same length as the message, I should just use that thing to send you the message. All right. That is simple as that. And the thing is, I don't have that too. Right? So managing the key for one time pad is impossible. So theoretically, this is very useful. It gives us a lot of useful knowledge. And theory of cryptography can work when we cannot practically use ￡1,000. The exception is if we use it in very specific instances, once again, almost for short text messages, one time paths were used there in the 60s and 70s, this evidence of spies using it, because they could have a code book. There could have a phone book.

So similar to the evening one machine, I basically have a book with me that said today's one time pad is, and then there would be a sequence.

And then I would make myself a short message. I can explore it to that. One contact is written in my book. I can write it on a piece of paper, hide it somewhere. The person i'm working with can come pick it up. They can then use their code book to explore the one part pad and get my message back. But it was pretty much done by hand. And we already pre distributed possible paths that we could use far in advance. And you can only really do that as the messages, once again, is very short.

So in that instance, maybe it worked. Once again, a fair operation needed. There was some people in the uk they declassify things after 30 years or 40 years. They were talking about the cold war with russia, and they said a lot of the messages they encrypted was, once again, people who were just too lazy. They were like they have these code books, but they just so much effort to look at the book every day. So they just kept using the same code word, like day after day, after day, after day, you can get some information about that. If you didn't do that, it would work as well.

So 1 . 5 theoretically secure from an information theory perspective, perfectly secure, from a usefulness perspective, not very useful. We cannot really use it. If you use some elements of it, you cannot into exactly coming back to what we had before. So we have alice involved. We have symmetric encryption, we need alice involved to have the same secret. Alice involved wants to use that secret. The same message from alice to bob. Other persons should not be able to get the key back. Other persons should not be able to get. Bank takes back, right?

So after the one time path, now we're getting into modern software close to what you are using. People when they looked at the one point piracy. Some of these properties are really good. We like the binary data, we like the x or we like the idea that we basically exploring the plane takes to random looking bit street. Right? But we know that it's not practical to use.

So how can we take one time pad and make it into something that is usable? So they come up with the idea of a streamside. It seems I would get the model that we had before alice involved. We have the same e we then would have an algorithm. We fit this key infant algorithm. This algorithm would just produce as a random binary sequence. What's it called citron sequence or history? We will then generate key stream, binary sequence of the same length as our plain text. We would take our plain text and we would export to this. And this would give us on cyber text. Bob would have the same key. Bob has the same algorithm when bobby is key, when bob basically initiate this algorithm with the same key, it starts generating random bit stream. It's generating the same random binary value.

Alice bob makes the same random value sequence. He extorts it to alice's architects. He gets the plane takes back, so we can see the similarities to one contact. Basically, we have random link, primary sequence of same link as plain text that we exhorted the plain text to make a cipher text. The recipient has the same key for the same path, and the path is take short to the cycle. It takes to get the plain text. The only difference is that the path is not truly random, but the path basically comes from a super resident on the generator that's seeded or initiated with a secret key.

So it's not truly random, but there's somebody outside of the news travel.

This makes it practical to use, because alice and bob can both have this algorithm and alice involved in a secure way and exchange the key, because the key is short. Right? The key doesn't have to be new for every single communication, or the key doesn't have to be the same length of the message as the one time pads should be your part.

So the scheme cipher is basically one time pad adapted to be used. In practice, these offer we have secret key. We have pc generator, we have generate history or cycling sequence or ivory sequence. We have generate the same length as the latex. We explore the two together to get our soccer test. Right? We can use the secret feeding to be whatever we want without having really eight birds, six birds, but it has to be sure. Right? The plain text can give any length, because we can just keep generating extreme until we have a key stream that's equal in length to the main text. Right? One additional thing that is added to here, we have this our basic model before. If we see the cycle text, you should not be able to recover the key on the main deck. So for he is stream cyprus is the same. The one additional requirement is that if the attacker sees some key stream, the attacker should not be able to predict future history. The moment an attacker is able to predict future extreme. What can happen? The attacker can then, if it knows the key stream that's gonna be used in the next message, the attackers that include his own message, because he can make his own message, explore it to that, predicted e stream and send it to god.

All right? Or if alice sends a message to god, the attacker will be able to decrypt that, because he knows the next set of extreme, then alice is going to general. Right? So wonderful, additional. Right? Let's finish off by looking at an example of this two topics. So what I would say at this moment in time, in this lecture, we look a little bit inside three different sides. Rc four is and aes in the case of a is this, especially aas it becomes a little bit complex. Right? But we are going to do it anyway. The one thing I want you to remember is that I don't expect you to know the details of what happens in the cycle. I want to give some additional information to make you appreciate how these things work, what the thinking is behind the people that voted. It also makes a good point when we compare symmetric encryption to asymmetric encryption. But I don't expect you to know every single small detail, like every single cycle, the level of detail.

And for example, experience from midterm or exam. Is it the basic little description? What is the key name? What is the basic idea that i'm hard works, but the block size? How much information we can encrypt? Not necessarily, what is the theory and the algebra that happens under these are changes. But I remind you of this, the ., in most of these cases that I think I would also like to make is that if we actually look at these things, obviously, a lot of effort went into designing it, and i'll talk a little bit about that next week. He is.

But if we actually look at how it's implemented and what the final design looks like, it's actually quite simple. It's based on very complex mathematics and competent forex. But the end result looks quite easy. It seems like it's easy to implement. As we saw last week, when we spoke about standards, that's a good thing. If we can have a good solution that people want to use, that are necessarily want to understand how it works. This when you use it, making it simple for them to use, makes it stronger, because if we make it really complicated and they make a little mistake, then it's gonna be broken. Right?

One of the more famous early stream surfers, the stream surfers are not that commonly used anymore, and you also have these days we use both into our cyprus.

So you can talk about speak to say, recently, extreme cyprus people sort of quite fast, because you are just generating history. You're not actually detropy detropy information, but subsidiary things like a is quite possible itself. So you could rather, if you really want to talk to the street side there, this means that a is actually underlying it's been generated.

One of the any examples of these are over party for it was invented by one of the best. And rc four is basically the short run code. Vision for. Rc four has two parts and initialization part, any key generation part, or sometime, arc four was considered a very secure cipher, right? So this is from the early 80s. It's not considered secure anymore. How did it do? Initialization? All right. Once again, as I just said, if we go through the slide, the idea is not ready to make you memorize this. But if you have done computer science and you have done some basic programming, you should recognize that this is actually very easy to code. As long as you have a little bit of information on what for loop is and what an array is, you can implement rc four. You cannot design rc four. That would be very hard, but to implement it would be very easy. You make three variables. The key, which is five array of size n is k which is a wide area of size twelve six, is the state variable, which is a five area of 206. Then you're gonna do a photo for initialization for I equal to 0 to 255.

Si the I position and the s area is gonna be equal to I the I value in a is gonna be the key value corresponding to I module a we just basically repetitively writing the key into the k variable, e several copies of the keys. And then we're gonna go through a second set, and we're just gonna shuffle to speak about part, which is basically shuffling it. Right? We're gonna say j is equal to j plus is I plus ki modular 56. Then we're gonna swap si value with SJ value. We're just taking the state variable, and we are shuffling the values around. That's it. At the end of the initialization, we end up with the same variable, which is an array of length. You are six. It contains basically a random permutation of the value, zero, inside.

And then the second part is e stream generation, and this is even shorter. Remember, for extreme software, we want to generate the same amount of key stream as we have software case. Right? If we have 80 bytes of fine text, we're gonna generate 80 points of history. So how do we generate one by the history? I is equal to I plus one modulated 256. Is is going to increase from 0 to 255, and then jump back to zero through j is equal to j plus si 1026. Swap si and SJ t is equal to si plus SJ one line for 56. And your p three five is in the selected list.

Basically, keep shuffling the state a little bit, and then we randomly select one of the state values could be are steam valuable PC value. Then we can generate enough history. We explore it to the plaintiffs. You may assign the end offers where we currently. So we have talked a little bit about the history of college making that LED into crypto. Lots of people are here, the substitution, cipher and variations of that for the alphabetic up until less than 7 years ago, many people started thinking about Information theory started, right? One time had, one time had sort of sales on the part of thinking for what we do in modern decade.

First, people would adapt one time PAD directly, and you start making steam cyphers. Right? The second family of street, symmetric soccer that we're going to look at from other ones, both sides.

We have to get to essentially, inside the big world of preferred systems, we have some major preferred and probably preferred. What's the matrix and asymmetric? Isometric? Is the other word, probably be right within some matrix e commerce systems. We have to be such a long cycle. So today we got this amazing piece of the system same suffer. And next time we see each other, we'll start looking at one cycle. In this character, you'll find slightly different people, soccer, steam, soccer is, like you will see what the difference between versus major encryption systems. Basically.

So that is it for today that would be finished. Remember that we still have the Orioles today, which we didn't talk last week. So what happened? Forum questions, which are on canvas. Based on what we did last week, just a reminder because sometimes I see people taking pictures of the board. You don't actually have to do that because everything on campus, all the notes, all the slides that give you even the extra discussion that the reading, all the solutions for the tutorials and everything is going to be uncanny. So you are Free to find it anytime there. We did with everyone if you want to take a picture, be happy and then go do it, but it's open to be on campus. Anyway. Thank you. I'll see you in the tutorial shortly. Thank you. Thank you.