### Security\_week6\_tu5-20241015

说话人1 00:00  
Is a sense, say that if we had given y and the hash of y we should not be able to find it out the message x the hash of y is equal to the attributes. Basically, in two b we show that it's very easy to find another x that the hash of x is equal to the hash of one. In other words, this is not a good hash function. It's a good wide wave function, but it's not a good hash function, because it provides one way function operation, but it's not saying create its resistance. Then it's not even be amazing is that he got to be different. It gives one of the three security properties you need for action, right? In the answer. It's a bad aspect. Okay? For question three, first, we use the hash function h of x or output length 128 bits, which extends input in box of 60 characters, meaning that a message is always split into box of 16 characters and include into the hash function. Right? We want to hash a message. The message is the message that has eight characters, because we can only have 16 characters at a time.

Remember, the function is similar to draw function. They have a block size. You cannot just arrive at the block size. Now we added with 60. This is the rule that we decide on in our system. Show that if we are using this pattern scheme, it's trivial to come up with different messages which produce the same aspect. Okay? So different messages don't necessarily need to be logical messages, right? It could be any different messages than the message that will give us the same hash value.

And then the first trick for the first thing that we need to remember is that we're not actually hashing the message. What actually goes into the hash function is then as each followed by six zeros, the simplest solution here is we have a message pattern with six zeros. And it can be any message, and it doesn't need to make sense.

So we can use all the following messages. We can use the message zero, which will be padded with five zeros, the message 00, which will have four zeros. The message 000, which will have three zeros. The message 0000, we will have two patterns, zeros, and then they see 00000. We shall have one missing 10 pattern. Right? After we had all these different messages, they all correspond to the base. Each followed by six zeros, vary into the hash function. If we have the same thing, we gonna get the same as result. Right? All of these messages will end up hashing. Do the same thing. The following question is, can we have a bigger planning scheme? It doesn't allow this to happen.

What we can do is we can change spanish scheme as follows. We can say, rather than just panic with all zeros, we can tag with one, followed by zero. If an attacker now tries to send scheme and they change it to the basis one, the actual basis is the basis y when y followed by 1000, which would be different. There's only one special case here. That's if the message is actually only 14 paragraph law, right? So then we can have abcde and ghi the government. We've had it with 10, and we can choose another message. The 40 characters followed by 10. That could be the legitimate message. So legitimate message could be 60 characters long with the same 14 initial values as us.

And then with 10, if we had this, they will end up to be exactly the same as. Right? So why we will have to have one special rule that says, even if a message is 16 characters long, we are then going to have higher additional block, right? And then encrypt that form. As that block is as both blocks. This is actually quite common. If you have access to some kind of the next machine, and you want to play with open ssl if you're going to open it yourself, and you actually try to do encryption and you try to do maths, right? You will often see that if your message that you put into it is the exactly the same size as a block, it will actually encrypt one extra block. Right? And it's basically trying to protect against different issues arising with the panel, right? With the banking scheme. Remember, the case is in those cases, if you pay bytes, wifi missing is 01, two fives missing is 0202.

In this case. So if we did the same thing here, basically, what would happen is, depending instead of fighting with zeros, you go back with school success, it will be the message six. In this case, we followed the pattern will be the message 05. It would be 004. That's why it would be 00033300000000001. That would be the real magic scheme that people would use. Right? You would still get into trouble if the message is exactly the same thing. Then you might have such right? Because it might be that I always advise of other messages. So normally, people have an environment extra long to make sure. Okay. Okay, so then the final one, assuming that the panic scheme is secure, what is the estimated computation? If they need to find a collision on the hash? Remember, we did the experiment with birthday. We said that the effort to find a hash, a on aa position on the hash is dependent on the size of the output of the hash function. If it has an input output, the provision, if it is two to the n divided by two, if our output is 128 bits, then basically, the effort is two to the hundred and twenty eight ÷ 2, which is to do this.

So we can estimate if it that we can find for the difference reality, this hash function is probably a little bit short. The shortest we should have for a hash function is 160. And on 64 is getting a bit essential. The last question, so we also looked at this in the in the slides, we said if we simply use cpc mode encryption, as is to vote cpc max, the following issue can occur. We can basically stage multiple messages together. In this case, we have two messages in zero, in one, and the second message in two, in three, in four. The question is, can you create a new message which is in zero in one while part block? So you can choose as long to be anything, and then basically three and then it got four with where the map tag of this message is the mac tag of the second message here.

In other words, we can stitch the two blocks together at the back for the new provide message will remain the back for the same admission for for purpose of calculating, we can use the iv equals zero.

We can send it back and be constructed from rock cipher from cdc mode. Id is equals zero is simply what we're going to do is for a mac, cdc mac, we just encrypt everything with cdc mode, and then we throw away all the intermediate context. We only keep the last type of days long as our backlog.

What happens here is we can do the following. We can first calculate what the map is for message one and message two, message one being message zero, message one, and message two, being message two, message three, and message four.

So what we see is that for the first message, the tag is equal to c one, which is the encryption of c zero, xoun one. For t two, the tag is equal to this c two, which is the encryption of cyxon three, which is also the encryption of c zero xon three four, which is the encryption of n two export to n three, all encrypted extort on n three, all encrypted. What he wants her in this is good. It's the incursive in zero, picture in one.

Or if we choose our message in the following way, we have n zero in one into x or t dash, the tag for the first message. And n three and n four. If we now calculate the hash of this message, c zero is equal to the encryption of n zero, x source of id which is the encryption of n zero. C one is the encryption of n one, x or the encryption of n zero for this encryption of n zero, c zero, which is equal to the first mac back, which is equal to the value here. C two is then equal to the encryption of c one, x or to m two x or to t dash.

But remember that this cy is equal to d dash, we choose the next plain text block to be into a sort of t dash. We basically cancel out the cy we are only here with the encryption of m two.

By choosing this block, in this way, we basically erased from the system, all memory of what has gone before of c zero and c one. Essentially, we're forcing the system is to restart, re encrypted n two, n three, nn four. All right? And so it encrypts only in two. They didn't increase in three, and they didn't increase in four. And basically, that would be the same as if we just calculated the mac tag for the saving basic. Because if we do the cbc by encryption of in creating for, we get max at two. All right? And basically, we're doing exactly the same thing here, where eliminating the effect of cy and c zero of the system of n zero and ny of the system. We forcing the cdc mode encryption that we are using before the mac tag with to restart with n two, n three, and n four. The back tag for this entire message is therefore only effectively the mac tag, on two, n three and four, which was the same as the mac tag or message in double dash. We expect a set of n two, n three.

Okay? That is why we can if we use cbc by early people, the the cbc back, then we can simply stitch the values. You get it right. Then it's quite as interesting as that. You can also actually try it for real time with something like open itself. And you can see that it actually, this is the end of the tutorials off with the tutorial solutions, also on campus.

So with the output, I went with the notes for the education next year. If you have been to get, you don't need to study for the exam. I will put on the solutions for the tutorial. And later in the week, i'll put on the tutorial. We have the problem set solutions. I'll give you the problem set solutions. But i'll only post it later. I'll probably post it friday, maybe from saturday. I my post to do some markings, make sure the solutions are correct. Before I post it online, before I do something that's wrong. So they become prevented. So everything else will be on there.

And then the solutions for the problem set will be on there, on friday evening or saturday morning. But you can study everything out so long. Remember that we have the quiz next week. The quiz is the only thing we are having next week. If you are finished with the quiz, then you can go to anything on schedule next week. If it works. There is no make sure. It's not important. I I it was. Okay. If you have any questions during this week, you'll be, let me know, okay, so good time for the study. Then i'll see you with a quiz. Makes me. Thank you.