AES, Blowfish and modes of operation (23-24)

Welcome back. In this video, we're going to be looking at two modern block ciphers. We'll be looking at Blowfish and the advanced encryption standard. If you look through the history books, you'll see that cryptography crops up every now and then.

It's a particularly fascinating topic when we look at it from a historical perspective. In the '70s, the National Institute of Standards and Technology in the USA came up with a data encryption standard called DES, for short, DES. This was secure for some time. However, with the short key size of 56 bits, this meant that actually, it didn't stay very secure for very long. So in the mid-'90s, the NIST put out a call to anyone and everyone to come up with a better encryption standard.

This competition was open to all, and anyone could submit, from big companies like IBM to researchers at academic institutions. As a result of that process, a number of different encryption standards were compared and went out to the cryptography community to perform cryptanalysis on it to see how they were in terms of security, speed, and portability in terms of putting it on the likes of microchips and so forth.

The community reviewed all of these and narrowed it down to five choices, one of which was Rijndael , and the other was an algorithm called Twofish. Both of these are symmetric block ciphers, and both have very good performance in terms of their speeds and also their security. The NIST, in 2001, announced Rijndael as the winner.

And we'll cover this cipher in this video. It's now interchangeably used with AES because, obviously, it was labelled as the advanced encryption standard. AES is the encryption which is used in SSL, secure socket layer, which allows you to use the internet in a secure way. The other entry, Twofish, was proposed by Bruce Schneier, who is a security consultant, has many books within the field and has a blog, which is really fascinating, if you want to keep up with security-related topics. Now you'll notice that Twofish is not the same as Blowfish.

Blowfish was a slightly earlier incarnation. It has a small block size, so generally, Bruce Schneier points people towards Twofish instead of Blowfish. But Blowfish is easier for us to understand, and it gets our point across without making it overly complicated. So we'll go through each of these in turn.

They are both in use today and currently are infeasible to break. So even though Schneier doesn't really point people towards Blowfish, he tends to point them towards Twofish, it is still computationally infeasible to break. So it's important to know that. Let's have a look at the advanced encryption standard structure. We start off with our plaintext, which is 128 bits long.

We split that into a 4x4 array of bytes. We can see that these are structured such that byte 1, 2, 3, and 4 are in the first column. 5, 6, 7, and 8 in the second. And 9, 10, 11, and 12 in the third. And then obviously, 13, 14, 15, and 16 in the final column. At the first stage, what we do is we take that plain text and perform an exclusive OR operation with a key generated from the main key.

We then go into a series of rounds. Within a round, we have a range of operations which have to be completed. We have sub bytes, shift rows, mixed columns, and add round key. Sub bytes is substitution operations which take the byte in a given position and substitute it with a different value.

Shift rows does as you would imagine. It shifts bytes along within a row. So the position would change. For example, we could take byte 14 and shift it along byte 2. It's cyclical, meaning that it will come back to the start and then move it in to the appropriate position. We also have mix columns. Again, mix columns works as you may expect it to.

It would take a whole column and shift that around to a different position. And then finally, we have the addition of the round key. We'll recall that our key schedule allows us to derive multiple keys from our main key. And in this instance, it will take the key for that particular round, so whether that's round 1, 2, 3, 4, or 5, et cetera, you'll have the corresponding round key, and the addition can be performed at that stage.

Having completed the first round out of 10 for the key size of 128 that we're working with here, we then loop back around to the start of that process and complete the next round. Obviously, the key changes to the appropriate sub key derived from the key schedule. So for the second round, we'll be using key 2 for the third round, key 3, and so forth.

One thing to note is that the mixed column step just here isn't completed in the final round. It doesn't really add any value at that stage. And so it isn't included for the sake of speed. Having completed the 10 rounds necessary, this key and block size, we then output our final ciphertext. So you can see that the advanced encryption standards makes use of substitutions, permutations, and also exclusive OR operations similar to the ciphers that we've seen elsewhere. And that's a quick overview of the advanced encryption standard. So that gives you an introduction to how the advanced encryption standard works, or Rijndael, if you prefer. Let's now turn our attention to Blowfish.

Blowfish has a block size of 64 bits, which, as I say, is generally seen as somewhat small for modern use. But it does have the key size of up to 256 bits, and it is using a Feistel cipher. So it gives you an example of using that kind of structure. It also has 18 subkeys generated from the primary key. So let's have a look at a little bit more detail on how Blowfish works. We have a plaintext block, which we split into our left and right half. This is in line with the Feistel structure that we've seen before. And we can see it executes this in a number of rounds.

We have 16 rounds with the Blowfish cipher, and we generate round keys from our primary key using the key schedule. So you can see the structure that we would expect to see with Feistel structure. We've got the left half being exclusive OR'd with the output of a function applied to the right half using the round key. And then we're flipping those in terms of the order.

So we flip the output of that with the right half and then move into the subsequent rounds. So in Blowfish ciphers, we do this a total of 16 times. There's then a final step before we end up with the ciphertext block. And that is to undo the last swap which happened and perform exclusive OR operations with the final two sub keys. The left half is then exclusive OR'd with sub key 18.

The right half is exclusive OR'd with the sub key K17. Those two halves are then joined back together to create our ciphertext block. Each of these octets is put through the relevant S-box, and it's the output of these S-boxes which are combined using operations such as exclusive OR and addition mod 2 to the power of 32 in order to get our final output.

The output of box S1 and S2 are combined using addition mod 2 to the power of 32. That is then exclusive OR'd with the output of S-box 3, which is then added mod 2 to the power of 32 with the output of S-box 4 to get our final encrypted value. When looking at block ciphers, one of the things that we haven't really addressed yet is how they actually operate on a set of data.

This can be referred to as its mode of operation. So far, we've primarily been working on the basis of you encrypt one block at a time and then move on. However, this can impact the security because you can start to recognise patterns in the data. So as a result, it's generally deemed more secure to use alternative approaches.

One of those alternative approaches is cipher block chaining. We'll have a look at how each of these works just now. There are a number of different types of mode of operation, but I'll cover two in this video. We'll look at electronic codebook and cipher block chaining. With electronic codebook, it's going to do each block in isolation. So it's taking the first plane block, applying the encryption with the key, and that results in the first ciphertext block and so on and so forth. However, the issue with this is that it doesn't really confuse the relationship between the cipher block and the plain block as much as it potentially could do. And if you recall, we need to try and increase the confusion and diffusion. So as a result, we have what's referred to as cipher block chain mode. In this mode, we take the output of the previous step and perform an exclusive ore operation with the next plaintext block before encrypting.

And you can see how this starts to link things together. Hence, it's called chaining. However, you may have noticed that with our first plaintext block, there is no previous output in order to perform that initial exclusive OR operation. We introduce something called the initialization vector, which helps us achieve this. The initialization vector is a pseudo-random number generator rated value. And so this increases the amount of confusion between the plaintext blocks and the cipher blocks.

So that's the two different modes of operation that I'd like to present to you today. So that's us for this video. We've examined two modern block ciphers, one of which uses the Feistel structure and the other one doesn't. Both are computationally infeasible to break, and AES is used in the majority of modern browsers for secure socket layer communication.

Blowfish is used in password managers, as well. So they are both in modern-day use. One of the key things that I want you to take away here is that it's important not to implement your own security from first principles. I would definitely not advise trying to implement something like AES unless it's an intellectual exercise. There are many open source implementations available, such as OpenSSL, which allow you to achieve this without having to implement it yourself. Often when such a thing is broken, it tends to be because of the implementation, and it's very difficult to get that right.

So it's better to use something which has been more widely used. Even then, things can go wrong in the implementation. For example, the Heartbleed bug showed that OpenSSL had an issue with it in terms of its implementation and was potentially leaking passwords for quite some time. Of course, OpenSSL has been patched since then, but it just goes to show that there's no bulletproof approach to this. So that's us for this video. I hope you've enjoyed it, and I'll see you next time.