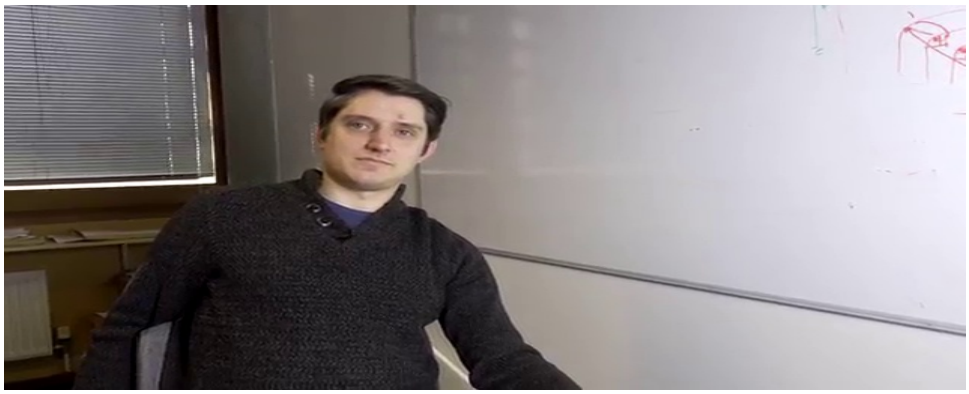
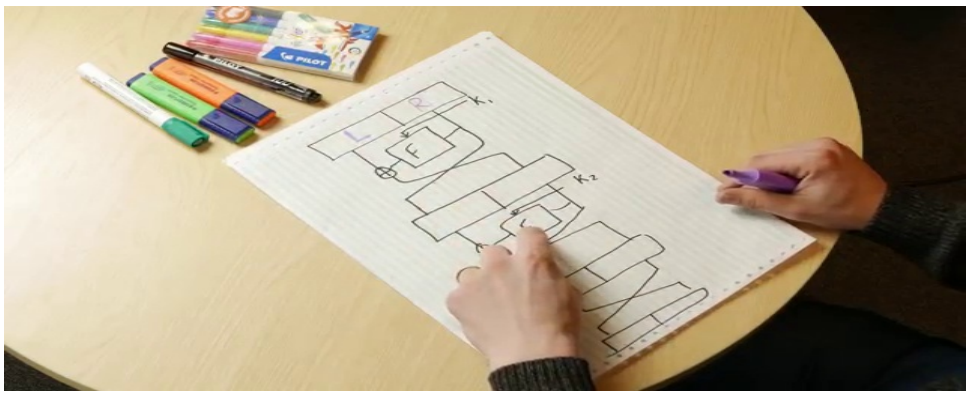
And yet, you just think that's that's great.The fossil network was designed roughly around the 70s, when horse Firestore, who worked for IBM, who was a German physicist with the NSA, he helped develop the data encryption standard death. Right now Dez was, as we know from previous video we replaced by AES eventually, mostly because of its short key length, but Dez, the structure of Dez is something called a Feistel cipher or a Feistel network, and there are few of these around 2 fish, for example, is a fossil cipher Faisal cipher. See use in padding schemes.Like the padding scheme used for digital signatures on certificates, Feistel cipher's are used for key schedules. It's a structure and then you put in some encryption rounds and a key and things like this and then it turns it into a cipher for you and it has some really neat properties.So I'm going to draw it out and then we'll talk about the interesting properties that it has. So you start with a block and we're gonna split that block into and then we're gonna take this as a right hand side. And this is the left hand side. We're gonna take the right hand side down. We're going to put it through some kind of function, right? Which is going to be some kind of pseudorandom function, like a hash or encryption round or something like this. We're gonna take it out here. We're gonna explore it with the left, and then we're going to bring the left down and we're going to bring the right down here.

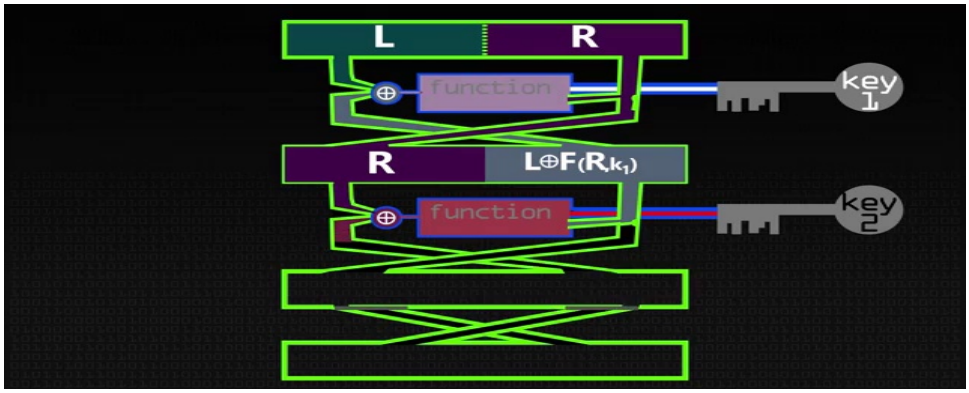


Half very good question. So in this case, yes, but in general no. You have unbalanced Feistel ciphers wherever left, and the right are different sizes for for this demonstration, they're going to be the same size as as as near as I can draw it. I'm not very good at drawing.So the next round is exactly the same we take whatever this new right is we bring it round. We go through F we X or it with the left and we come down here like this and so on. And you can repeat this process as many times as you like for how many rounds and then at the very end after the last round you flip the output like this.About this encryption algorithm is how you decrypt it to decrypt using the same file stuff you encrypt by putting your block in. Here it goes through. And even if this F is a one way hash function, that can't be reversed, that still decrypts it.I mean, I like bases bases, wildlife, faster ciphers, so we're just gonna do it. We're going to start with L&R; and we're going to work through and we'll see that it does actually reverse itself, right? Which is just amazing. I mean, maybe you've seen it fast and I thought maybe you know this happens, right? But I think when the first time I learned about this, I thought that is that is awesome.

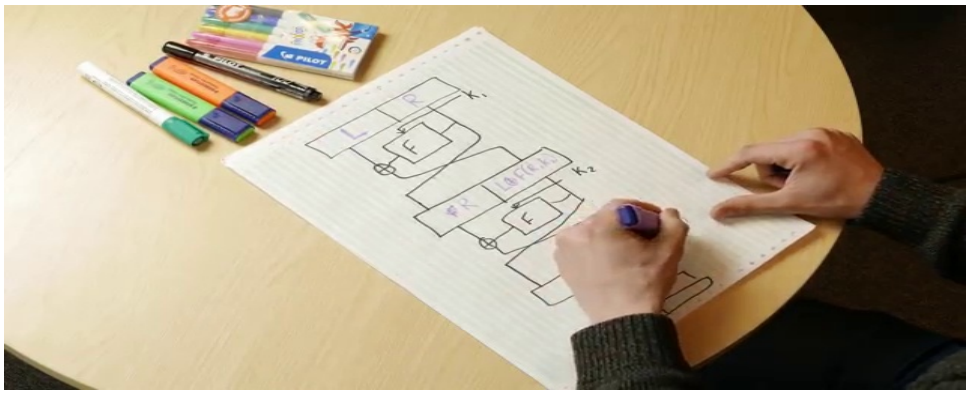
So let's do this.

This is the left. This is the right now for the sake of argument, they're the same size in this one, the right is going to come down here, and it's going to go through this function. So we're going to put in a key into this F and.There's a lot more secure to make these subkeys, so lots of different keys for each round, so it's gonna be key one. This is going to be key to like this.

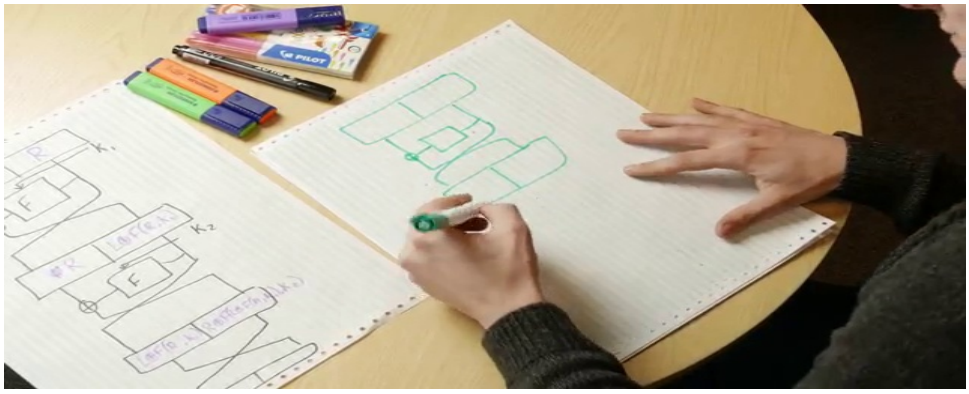




It's going to be X sword with this are so this output here is our X or F of this which is L. I'm going to run out of space LX or F of R K1 of K2 like that does that is that right? I think it's right. Maybe it's a good thing that we didn't do 3 rounds or 4 rounds of this 'cause this could take me quite awhile by hand.

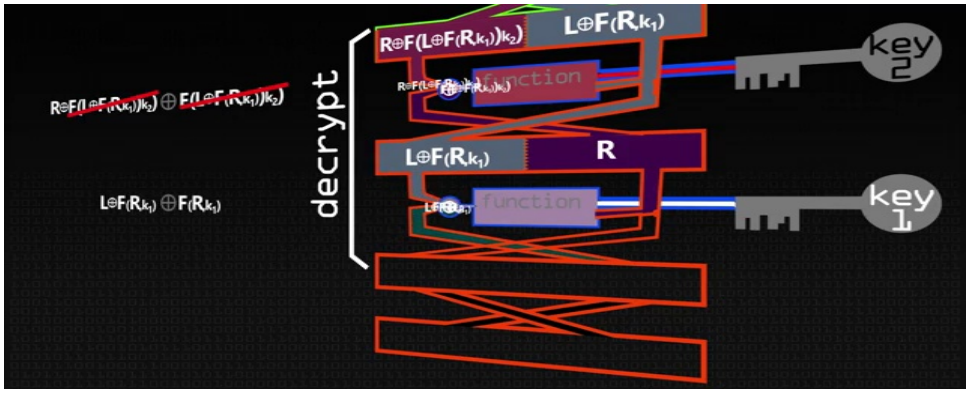


This one is going to get copied down so this is going to be LX or.R and K1 now both of these will look like gibberish. We're going to switch them around here. I won't draw them in quite yet, right? But this one comes down here, and this one comes down here. Feel free to animate it, Sean, thanks for that, right? So now we're going to see how we can decrypt this back to Eleanor and all we have to do is take this, put it in the top and we have to swap our subkeys around. Because of course the rounds are happening in a different order, so I'm going to draw this exact same structure again on the next piece of paper so that we have something new to work on. Otherwise, I'm going to get very confused. Yeah, alright, let's go again.



I think that's right, you know. So the keys need to be in a different order, so this is going to be K2 coming in here and this is going to be K1 coming in here. Reversing these keys algorithm not too much of a problem that will be a list of data or something like that. Very straightforward, all right? So let's put.

This gets copied down here. So LX or F of R K1. So let's go again.R comes in here.



It becomes F of R K1 excel with this.And it's going to be a combination of all of these, so this one times this one.

Plus this one times this one. Plus this one times this one plus this one times this one. And then we repeat this process for each of the values. So we're taking bits and bytes from all of these in this column, jumbling them up, moving them around, shifting them.