Symmetric Ciphers

Description: Steve explains, very carefully and clearly this time, why and how multiple encryption increases security. Steve also carefully and in full details explains the operation of the new global encryption AES cipher: Rijndael.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-125.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-125-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 125 for January 3, 2008: Symmetric Ciphers. This show and the entire TWiT broadcast network is brought to you by donations from listeners like you. Thanks.

This is Security Now!. And here he is, ladies and gentlemen, ready for a new year of security, Steve Gibson. Happy New Year, Steve.

Steve Gibson: Hey, Leo. Happy New Year to you. And we're going to kick off the new year of Security Now! with a - well, I'm really excited about this. I've been doing extra research and getting my notes together. And I'm going to tackle something that I think people are really going to like. And it comes from really that bungling attempt I made a couple weeks ago when, well, it was the issue of double encryption, the question we answered several Q&As ago where some guy said hey, you know, what if I encrypt something with one key, then I encrypt it again with another key? Isn't that, like, much better than encrypting it just once? And I absolutely know that it is, and I know why it is. And I realized that in the several years intervening from the time I first talked about symmetric cryptography and symmetric ciphers, I've written a bunch. I mean, I've implemented Rijndael, which is the AES standard, in Assembly language. I know exactly how it works. And then when I was trying to explain subsequently in the next Q&A why that was the case, I got into all this 2 to the power of 128 factorial stuff, and everyone's eyes crossed, and you were saying, okay, Steve, okay.

Leo: We'll take your word for it, Steve.

Steve: So first of all it bugged me that - well, and actually people didn't know that you and I were under time pressure for that, too. Your schedule had been packed here toward the end of

last year. So I was feeling like, oh, I can't really just, like, stop and really explain this.

Leo: Well, we've done, I think, three or four shows in a couple of days, so...

Steve: Yeah, actually we did five shows in eight days.

Leo: Geez. Including this one.

Steve: Two the prior week, two yesterday, and then one today. So...

Leo: I'm sorry, Steve, but you get six weeks or five or something off. You get some time off, anyway.

Steve: Exactly. I'm going to forget the name of our podcast by next time.

Leo: And by the way, I want everybody to know, I said this to you, oh, good, we don't have to do any shows, we're going to take a couple of weeks off, the whole network's going to shut down for December. And he said, "Over my dead body. No way. I want to catch up with TWiT." And you have. Congratulations.

Steve: We've passed up TWiT. I'll never catch up with the Giz Wiz because you spit those things out one a day, so...

Leo: We even took a week off with the Giz Wiz, and it's still, like, we're close to Episode 500.

Steve: Yeah, okay. Well, I'm just not going to...

Leo: Don't worry about it. But for the weekly shows, you are now number one. For the once-a-week shows.

Steve: Yay. I like number one. Number one's my favorite number. So anyway, I don't know what exactly are we going to call this? Anatomy of a Symmetric Cipher, or...

Leo: Yeah, just Symmetric Ciphers. I think people are very - first of all, we're going to get into this in detail in just a second. But first, can you kind of distinguish what symmetric cipher is? What is it we're going to be talking about? I mean, I know ciphers are codes. And we use encryption all the time.

Steve: Right. And the reason, okay, there are a couple reasons that symmetric ciphers are cool and powerful and neat. First of all, symmetric ciphers are the workhorse of encryption. Symmetric ciphers do all of the heavy lifting. They are sort of the bulk encryption technology which fundamentally protects everything. That is, for example, we've talked about asymmetric

ciphers, which means you have one key to encrypt and one key to decrypt, and those keys are different. Well, those are cool because that's where you get so-called public key encryption, that is to say where one of those keys might be kept secret. Doesn't have to be, but typically it is. So the idea being you can publish one of the keys, which could be used to encrypt the contents, and you keep the other key private. Well, only the matching key to the one that you exposed can be used to decrypt the contents. And so there are all kinds of things you can do with this.

The problem with that is it is excruciatingly math number crunchy intensive just to do a little bit of encryption with an asymmetric cipher. So nobody uses an asymmetric cipher for bulk encryption. You just can't. So what instead they do is they use the asymmetric cipher, which is so time consuming, just to encrypt the key, that is a symmetric cipher key. So you use the time-consuming but very powerful public key crypto to encrypt the private key. I mean, I'm sorry, to encrypt the symmetric key. And then you use a symmetric cipher which runs really fast. And so really you still depend upon both. But the symmetric cipher performance is super high, but that's really what we're depending upon. So if there's, like, a blob of data, like for example we were talking about Jungle Disk, Jungle Disk uses Rijndael, the Rijndael cipher, to encrypt all of the data that it sends up to Amazon's S3 servers when you give it a key that you want to use. So there's all that bulk data sitting there. And it is the strength of the symmetric cipher which is protecting it, the fact that there just is no practical way to decrypt it. So, for example - well, okay. So there's why symmetric ciphers are cool.

Leo: Right. So now that we understand what symmetric ciphers are, are you going to - you're going to actually - now, Steve, before we got started said "I'm going to explain it so that you understand it, Leo. By the end of the show, you will go, oh, that's easy. That's simple." I said, "Steve, you're a little ambitious. You're giving me a lot more credit than I deserve."

Steve: We don't have any errata this week, since we've been cranking out shows so fast there's been no chance for any errata to sneak in.

Leo: Oh, and I thought it was just because we were perfect.

Steve: There's been no chance for any feedback. But I did have one really nice sort of Christmas story relating to SpinRite that I wanted to share real quickly. This was Scott Keoseyan. He wrote to us and said, "SpinRite = Success," was his - it was a little formula with an equal sign there. He said, "Steve, I just wanted to share a SpinRite story. My 12-year-old son came to me Saturday evening with his laptop, parens, my old laptop, a Dell Inspiron 8600, that would no longer boot. On boot it would bluescreen, saying that the primary partition was unmountable due to errors." He said, "I had purchased SpinRite a couple of months previous to fix another machine, but it turned out the issue with that machine was with the main board on the machine, so I was unable to really put SpinRite to the test. Anyway, I went ahead and popped the disk into my son's computer and let it run overnight. When I came back the next morning, I rebooted the machine, and all was well again. While my son doesn't keep anything irreplaceable on the machine, having to rebuild his machine from scratch would have been a four-to-six-hour ordeal. My time is definitely worth more than the $89 I spent on SpinRite. And just having to rebuild that machine would have ruined my whole weekend. I was very glad to have access to SpinRite and that it worked so well. Thanks," signed Scott.

Leo: Isn't that great.

Steve: So, yeah. And you know, I have to say, all these testimonials and feedback that we

receive on SpinRite, it's so often about machines that no longer boot because the system just keeps on going until it finally collapses to the point...

Leo: It just can't - yeah.

Steve: ...that it just can't get going.

Leo: It can't go on. And that's - it's sad because people really don't - they defer everything. And it's like with the house or anything else. You defer maintenance on your car until it stops running. And then you go, uh-uh.

Steve: That's exactly right. And again, I know it's - SpinRite's $89. It's not something that people think of, like oh, look, I've got an extra $89. But the fact is, if SpinRite were used prior to systems finally collapsing to this level where they just bluescreen, then these problems would be prevented. So for what it's worth...

Leo: Well, I think people look at 89 bucks, and they say, well, I can get a new hard drive for that. They forget, we're not talking about the hard drive, we're talking about your data.

Steve: Precisely. And the point Scott makes is a good one, too. I mean, I sort of enjoy setting up machines because it's like, okay, it's interesting for me. But I find myself now trying to do multiple things at once because it takes so long to get Windows XP installed and then update all the patches and get it going and then tuned, and load your apps and get them registered and all that. It's like, okay, really, what's your time worth?

Leo: Exactly. What's your data worth?

Steve: Yeah, that, too.

Leo: All right.

Steve: So you're right. Here's the goal that I have for our listeners and for you. Symmetric ciphers are incredibly cool. And I realized I have a really much better handle on the internals of them as a consequence of having written some, implemented some that of course have been designed by crypto geniuses. And I'm not a crypto designer, but I've implemented the actual code. And...

Leo: So how do you do that? Is there a spec? Do you work from an algorithm list?

Steve: I'm actually, in this episode, we're going to understand the new AES standard, Rijndael. I can explain what's inside so that afterwards people are going to be going, oh, is that all? That's all there is? And that's so bulletproof and strong and amazing? It's like, yes. I mean, it turns out it's not that hard, and it can be explained in a way that's pretty simple. But I want to, also in the process, I want to answer the question that so many people had. I mean, it's amazing. I'm looking at another that I was going to sort of use as an intro to this. Adam in Ottawa, Ontario, he said, "Steve, I've been a listener for a long time. And as a graduate from a

computer science program with an interest in security, I enjoy listening to the show. In the past I've disagreed with you on some topics, but never so much that I felt compelled to write in. Now I am."

Leo: Uh-oh.

Steve: "Last week, Episode 120" - so it's last week for him when he sent this - "you answered a question about whether encrypting something twice was more secure than encrypting it just once. You said that it was, but I'm not so sure. If good encryption maps plaintext, something intelligible, to something completely random, using a key, wouldn't mapping plaintext to random, then to random again, be the same as just mapping to random once using a different key?" And of course that's what everyone has thought. I mean, so many people have thought this. So I'm jazzed at the fact that people are taking the time to think about this and scratch their heads and write in and tell me that they think I'm wrong, in fact, which is totally cool.

And so he goes on, he says, "What I mean is, if I use key K1 to encrypt my plaintext to something random, then use key K2 to encrypt that into something random again, isn't that the same as using some unknown key, K3, to encrypt my plain text just once?" And then he goes on to more detail, but we understand that this, I mean, this is really what people have been talking about. And I want to explain why it's not the case, and in the process what I was trying to explain when I got us all mumble-jumbled up in that 2 to the power of 128 factorial stuff.

Leo: So they just think, and I guess this is reasonable, that instead of two keys there must just be one magic key that would take the final scrambled version all the way back to text.

Steve: Correct.

Leo: And I don't know, I'm not sure why they assume that. I'm not sure I would leap to that. But I guess it makes sense.

Steve: Well, okay. It's a really - to answer that particular question, it's a matter of scale. So let's, instead of talking about Rijndael, that is a 128-bit block cipher with a 256-bit key, let's create a dumb, simple little cipher that's much easier to talk about.

Leo: Perfect for me, okay.

Steve: So this is...

Leo: You know, a simple little host.

Steve: This is the Romper Room cipher.

Leo: Okay.

Steve: It's just 4 bits. That is, it takes 4 bits at a time, and it encrypts it into a different 4 bits,

and it has a 1-byte key. So an 8-bit key. So it's similar to what we were talking about inasmuch as the Rijndael cipher, for example, that Perfect Paper Passwords uses, and that is used often, is a 128-bit block with a 256-bit key. So the key is twice as long as the block that you encipher each time. So we've got that. Now, in the Romper Room cipher we've got a 4-bit block that's controlled by a 256 - I'm sorry, by an 8-bit key. Okay. So first of all, if we're going to encrypt a 4-bit block, we're going to take 4 binary bits, which we know has 16 possible combinations. You can have 0000, 1000, 0100 and so on, until you get 1111. So there's 16 combinations, 16 possible combinations of 4 binary bits. This little Romper Room cipher will take any 4 of those bits we put in and translate them to a different 4 bits out, under the influence of this key.

So, okay, now the problem - there are several problems with the Romper Room cipher. I mean, no matter how good the cipher is, the problems are that there just aren't enough bits. That is to say, if you didn't know what the key was, and you didn't really even care what the key was, you wanted just to map the cipher, all you would have to do would be, if you had access to the plaintext, that is, what's not encrypted - and many times people do have access. For example, you could encrypt your own data through the cipher and get out the cipher text. Well, since there's only 16 combinations of 4 bits, you could just write down - you can create a little table. You just write down, okay, 0000 is turned into 0100, 0001 is turned into - since there's only 4 bits, there's only 16 possible inputs, and so it's easy to create a table of the 16 corresponding outputs. You don't have to care what the key is. You don't have to even look at it because you're able to just build, write down what the cipher does. And obviously, looking at that table, you could then take other encrypted text and decrypt it. You just use the table backwards.

Leo: It'd be like a transposition cipher where you'd have an alphabet, and you'd say A=Z, B=A, C=B...

Steve: Ah, that's a very good point because a transposition cipher that you were just mentioning, that would be adding a constant to each of them.

Leo: Ah, it's not as random, okay.

Steve: Exactly. So, for example, if you just added 4, you're right that 0 would become 4, 1 would become 5, 2 would become 6 and so on. And when you get to the end, you wrap around.

Leo: So this is a little better. So it'd be like randomly A=Z, B=Y, C=F.

Steve: Exactly. Now, that leads me to the perfect next question, which is, okay, one of the reasons this thing is dumb - it's the Romper Room cipher, after all - is that there's only 16 possibilities. But how many ciphers are possible? That is, how many possible tables - remember we just talked about a table where we just write down, you know, we have 16 entries in this table with each one of the possible inputs and the corresponding output. How many tables could there be?

Leo: Well, you said there was a 1-byte key, so I'm going to guess 256.

Steve: Well, exactly. Now a 1-byte key going into the cipher, as we know, 8 bits gives us 256 possible combinations. So the Romper Room cipher that operates under the influence of a 1-byte key, as you said, Leo, can have 256 tables. But the question I was asking is a little different, and this is why it's so important. How many possible mappings are there? That is, we can access 256 of those with the key. So we know there's at least 256. Well, we hope there

are, otherwise some keys would give us the same cipher. But the question is, how many possible mappings could there be? So think about it this way. If we put in a 0 as our first test, we could get out 1 of 16 possibilities. We could get out 0, 1, 2, 3, 4, 5, 6, all the way up - I'm sorry, 0 through 15, which is 16 possibilities. So in terms of, like, the possible combinations, we put in 0. For a given, say that we have a given cipher, we put in 0, we could get out 1 of 16. Now we put in 1. Well...

Leo: So it's 16 squared.

Steve: No.

Leo: Oh, I'm so slow.

Steve: That's what's so cool is we put in 1. Now we've used up one of our outputs when we put in 0. So we've got 15 left. So that is when we put in 1, we could get 1 of 15 remaining outputs.

Leo: Ah, it's a factorial.

Steve: There you go. When we put in 2, we could get one of 14 remaining outputs because we're consuming one each time. So it is 16 factorial.

Leo: That makes sense. Okay.

Steve: So I want to make sure everyone gets that, that is, the total number of possible ciphers is 16 factorial.

Leo: 16, that's 15 times 14 times 13 times 12 times 11 and on.

Steve: Exactly. So as we're doing this, when we get all the way down to putting in - we started at 0, so the last one would be 15. We've used up all the others. There's only 1 left because the other ones use it up. That's the total number of possible mappings of just 4 bits into 4 bits. And what's so counterintuitive, 16 factorial is 20 trillion, 922 billion, 789 million, 888 thousand...

Leo: Wow, I had no idea...

Steve: I know. Factorials are huge. So it's 20,922,789,888,000. That's the number you get if you just multiply 16 times 15 times 14 times 13 times 12 times...

Leo: Okay, you can stop. It's not the 12 Days of Christmas. We get it.

Steve: Okay, okay. So my point is 20,922,789,888,000 possible mappings. Now, how many of those can we access? 256.

Leo: Ah, right. So that's - we need a bigger key.

Steve: Well, exactly. So my point is that, even though it's counterintuitive, until it hits you, the size of these numbers, it's like, wait a minute, I'll just find a key that gives me the mapping that I want. Well, good luck, because - and my point is, most of the mappings are not available.

Leo: Are not available, you only get a few.

Steve: With just 4 bits, if there's 20-trillion-plus possible mappings, but with an 8-bit key we can only get to 256 of them. Okay. So what that means is, now let's stop here and now switch to Rijndael. Let me tell you, Leo, it was hard to find out what the factorial would be for 2 to the 128. So remember with real Rijndael, with a real cipher, not the Romper Room cipher but with a real cipher, we don't have 16 bits. We've got 128 bits. Okay? So we're talking about the total, what we're looking at, remember is the reason we were doing on the Romper Room cipher 16 was it was 2 to the power of 4. 2 to the power of 4, and you take that, and you take the factorial of that. So 2 to the 4, of course, is 16. That's why we were doing 16 factorial. We know that's 20 trillion. Okay. So that means that the number of possible mappings on a real cipher, an industrial-strength cipher like Rijndael with 128 bits, is 2 to the 128 power. You take that and do the factorial of it.

Leo: Okay. Big number.

Steve: Now, Leo, I found some math genius somewhere on the Internet who she spent her whole life coming up with cool ways to do factorials. I have the size of that number, thanks to her. It is - okay. It is a number, I can't even say the number. I don't think there's a possibility to say it. It is a number with 1,296 billion billion billion billion digits.

Leo: Digits.

Steve: Digits.

Leo: Okay. It's a big number.

Steve: 1,296 billion billion billion billion digits. That's how many possible mappings...

Leo: I never heard of such a big number. That's 10 to the - geez.

Steve: This is the biggest number that's ever been written. It blew up three computers when I was trying - no. And so this is what I understood, but I didn't explain it properly a couple weeks ago, that is, that's the number of possible mappings that a 128-bit block symmetric cipher could have is a number that has 1,296 billion billion billion billion digits.

Leo: Now, you only get to pick from a subset of that, though, because of the key size.

Steve: And that's exactly my point. And that is, okay. So Rijndael can operate at different key lengths. You can use it with a 128-bit key, a 192 or 196, I can't think of which...

Leo: 196, yeah, and then 256, and then...

Steve: Yes. So, okay. So a number like 2 to the 256 is the number of keys that Rijndael can have. Now, that's a big number also, but it's not anywhere nearly as big. I'm looking at it here, and it's...

Leo: How are the keys selected from the superset?

Steve: Well, that's what we're about to do. That's the next part of this podcast is how does Rijndael itself actually work? But what I wanted to explain was that 2 to the 256 is not a huge number. I mean, it's a big number, and it needs to be big because, if we look back at the Romper Room cipher, the other problem with it, not only was there a problem that the block length was too short, it was only 4 bits. So we could simply write down a table to figure out all the mappings. But the Romper Room used a 1-byte key, which as you pointed out there are only 256 of them. Which means the other way to crack the cipher, if you didn't want to just - if, for example, if you had no access to the plaintext, okay, remember that if we had the plaintext we could just run it through the Romper Room cipher, not caring what the key is, and create a simple little 16-line table that shows all the mappings. Well, you cannot do that with a 128-bit cipher because it's got 2 to the 128-bit possible combinations.

Leo: Big table, yeah.

Steve: Big table. And so in fact that's even why - that's why short block-length ciphers like DES, the prior standard, was a 64-bit cipher. And so it would encipher 64 bits at a time. Well, that's strong. But it turns out that people like the EFF - the EFF created a blob of hardware that was able to crack DES, any given instance of DES, in a day or two.

Leo: You could map a table 2 to 64.

Steve: Yeah. You wouldn't want to. I mean, that's still a lot.

Leo: Yeah, but a computer can do it.

Steve: Big computers. Certainly there's enough data around on the Internet to do that. But it's still - it's big, but it's not too big. But just remember, when we double the bit length, we're not doubling the table size. We're squaring the table size. So 2 to the 128 is - you cannot do a table that's that big. Okay. So in fact it is - do I have it in front of me? It's roughly 3 times 10 to the 38. That would be the length of the table. So that would be 3 and, well, 3 and 39 - or a total of 39 digits. So that's a really long number. So, but the problem with the Romper Room cipher with an 8-bit key was a brute-force attack. We've talked about that a lot before. If you only had the ciphered text, and it was all gibberish, you could try using key 0, then key 1, then key 2. Well, you've only got 256 of them. So in a short time you could figure out what that was. And in fact, that's the same key length, if you want to call it that, used on the wireless keyboards that we talked about. There's only 256 possible bytes that could be XORed with the keystroke data. So it's trivial to try them all.

So that's the problem with the Romper Room cipher is the key is too small. And that was the problem with DES was that a DES key was only 56 bits. And so that was, like, way too small because equipment really got much faster much more quickly, and DES just ran out of steam because the key wasn't long enough to prevent a brute-force attack. So Rijndael, that runs with 128-bit key minimum, or 196, or 256, those keys are so big that it's just - it's absolutely impractical to try them all, which is the way a brute-force attack works. So, okay, so I just wanted to put to rest this - or to give people a sense for the size of the total number of possible mappings that a symmetric block cipher offers. So we've got that.

Now let's talk about how Rijndael works, that is, up to this point in all of our podcasts we've talked about this as a black box. It's a black box, and that's true of all of these ciphers, although we have explained, for example, the inner workings of public key crypto. Remember we talked about exponentiation and how it's possible to do an exact exponentiation, but incredibly time consuming to reverse that, to do an exact logarithm in order to reverse that. And public key crypto uses the fact that it's a so-called one-way function, or at least one way easy, the other way really, really hard, and no one's ever figured out how to do that. And prime factorization is the other thing. It's very - it's trivial to multiply two big primes. It turns out that, if you don't know what they are, and you only have the product of two primes, it's incredibly time consuming to go the other direction, to do a prime factorization of that number. So public key crypto uses the concept that it's easy to go one way but not the other in order to create its security.

So but in terms of a symmetric block cipher, we've never looked inside one. And that's what we're going to do for the rest of this podcast is understand how Rijndael functions, that is, actually how does it encrypt data. And what's a little distressing is that it's really not that complex. All of these symmetric ciphers use the notion of a round, that is, they're iterative. You put data in, and they do something to it a number of times. And the strength of the cipher is based on how good what you do to it is and how many times you do it because basically it's making it much more difficult for anyone on the outside of this black box, which is what a cryptanalysis is, it's like trying to figure out something about his that will find some weakness in it.

Rijndael won in the competition because it was very clear and a very clean algorithm. The designers were able to explain every aspect of it and why they made the decisions they did. And they deliberately designed it to be efficient on 8-bit processors, which are the very slow and on low-power processors, for example, used in smart cards. So they wanted a cipher that you could express the algorithm efficiently in an 8-bit processor. So things needed to be able to be done easily with just 8 bits. And they also wanted it to be fast on 32-bit architecture, the architectures of the day.

So what happens is it starts with the key. We take the key, and there's something called the "key schedule." In symmetric ciphers a key schedule is the algorithm used to expand the size of the key to create much more material which will be used throughout the encryption process. So the idea is, essentially, you have a bunch of carefully chosen random data, and the key is used, mixed with and to select from a pool of random data. And this is, it's random, but it's always the same. So that part of the specification of the Rijndael cipher and many similar ciphers is blocks of data that, when you look at it, it's like, okay, I hope I don't have a typo in here anywhere because you've got to get it exactly right. So the key is, through an algorithm that is specified, is mixed with this random data in order to create a much longer - essentially it's like an internal key. It's called "key expansion."

The way Rijndael works is that the block length, for example, is - there are various block lengths of Rijndael. You can actually implement Rijndael in a block length that's any multiple of 32 bits. But the standard that was chosen is 128 bits because that's enough that you could make a 256-bit block Rijndael. But it's like, okay, 128 is already so many in terms of the number of combinations, there just isn't a practical need to go any further. So the standard was set at 128-bit block length. The way Rijndael works is the key is expanded to a length which is a multiple of the block length. And this expanded key, different pieces of it are used for each of

the internal cycles, or rounds, of the cipher.

So say that we had a cipher with 14 rounds, that is, we were using Rijndael with 14 rounds, which is the number of internal cycles Rijndael uses if you use a 256-bit key and the standard 128-bit block. If you use a half-size key, that is, 128-bit key and a 128-bit block, then you only need 10 rounds. For a longer key you need more in order to get the equivalent protection. So the 256-bit key Rijndael uses 14 rounds. It uses one more set of key material than rounds. That is to say, so 15 sets of key material. So the key expansion expands this 256-bit key that we give it to 15 sets of 128 bits. So we've got 15, essentially 15 block widths of stuff, which is created using this pool of randomness that is part of the cipher.

So the data comes in from the outside of the cipher, and it's XORed with the first of this 128 bits that was derived from the key, just a simple XOR operation. Then it goes through a process of mixing the data, which is essentially three different steps. There's a thing called an S-box which is commonly used in crypto that takes a byte in and maps it to a byte out. That's all it is. It's a 256-entry lookup table called an S-box. And so the bytes of the cipher that come from the XOR, each byte goes through this standard lookup table. And there's only one of them, and it's always the same. So that doesn't change. It just maps 1 byte in to 1 byte out. Then the way to visualize the way Rijndael works is, if you take the 32 - I'm sorry. You take the 128-bit block, and you put it into a grid of bytes so that you've got 4 bytes down and 4 bytes across. So you've got a 4x4 grid of bytes. Then the next thing that happens is that the rows of this grid are shifted. And each row is shifted a different amount of bytes over.

So again, first of all we've got - we have a byte translation table. Then we're just shifting bytes around. Then the final thing is that the columns of this 4x4 grid of bytes, they're mixed within themselves, that is, so a column of 4 bytes is going to be 32 bits. That's mixed with a matrix multiplication which implements a polynomial which only has the factors of 0, 1, 2, and 3 because it's easy to multiply by 0, 1, 2, and 3. Well, obviously multiplying by 1 does nothing. Multiplying by 2 is just a shift operation in binary. And multiplying by 3 is just you shift and then you add the original in, so you've got 2 plus 1 is 3. So the designers of Rijndael carefully chose these things to be easy to do.

Okay, now, that's all there is in Rijndael. That is, you do the XOR, then you do the lookup table, you shift the rows over, and you mix the columns. And you do that 14 times, and you're done. And it turns out that there are attacks that have been found on so-called "reduced rounds" versions of Rijndael. That is, if you, for example, only ran through that process five or six times, it turns out that things are not yet mixed up enough that it's not possible to find, like by analyzing just from the outside, it's still possible to, like, determine some bits of the key because things haven't been obscured sufficiently. So and in fact that's exactly why they chose 10 rounds. They added four rounds to six, where six was the last point at which any weakness could be found because from a crypto standpoint they understood what each round did, and two rounds was enough to create bit dependencies that couldn't be tracked. So they added - they thought of it as that six rounds with two ahead and two afterwards to give them a total of 10. And they were able to demonstrate that that is extremely conservative and absolutely safe.

And then if you think about it, they've essentially - they've XORed a chunk of this internal key for each round. So as the data goes through, it then XORs the next 128 bits. Then it runs through this simple process of byte mapping, shifting, and mixing the rows of columns. Then it XORs the next chunk of the internal key and does it again. The next chunk, it does it again. Well, that process is reversible, that is, everything - and then of course you have to have reversibility because that's where you get decryption. There's no, like, magic other way to decrypt it. Literally decryption is just running Rijndael in reverse. And this is the way...

Leo: That's where people got the idea that you could kind of skip, if you did it doubly, you could just - because it's reversible.

Steve: Yes. And so...

Leo: Like an XOR.

Steve: So, well, exactly. But it's reversible. The reason you cannot simply say I'm going to use one key and encrypt and another key and encrypt and there's a third key somewhere that's equivalent is the key space. I mean, we were talking about that. The number of possible mappings that that simple process produces is that thing with 1,296 billion billion billion billion...

Leo: And you have no way of knowing which one it was.

Steve: Exactly. And remember, too, that the key - we have a much, much, much smaller key space. That is, the total number of keys, 2 to the 256, even though that's a big number, I mean, it's so big we can't, I mean, it's strong against brute force because brute force means trying them all. Well, the fact is, and unfortunately I'm not enough of a cryptographer to be able to state this definitively, that is, that there is no third key that is equivalent to the first two. But it is phenomenally unlikely because - and there may be a way of demonstrating and proving definitely that there is no third key that is equivalent of the first two. But just in terms of the number of possible mappings, you would - it would have to be that two keys selected two mappings such that there happened to be another one that a key could select. And that's the point I'm trying to make is that that hypothetical third mapping would have to be available by some Rijndael key.

Leo: And it's a pretty low chance that it would be, given the size of the...

Steve: Precisely, the size of the total number of possible mappings that 128-bit cipher can have, I mean, it's just so ridiculously small. And as you said, Leo, you'd still have to find it. And you can't find it because 2 to the 56 is a massively large number that's completely infeasible to brute force. So essentially this is how Rijndael works. Internally it's not complex. It's pretty simple to implement. They designed it to be easy enough that an 8-bit processor running on a smart card would be able to do it. And it would be very fast on 32-bit machines. And essentially it owes its strength to the fact that it's a bunch of simple operations where each cycle does a good job of mixing stuff up. And when you do it 14 times, it's so mixed up that no analysis from the outside can figure out where the bits went when they went inside. They just got all scrambled around and all interdependent in a way that nobody could figure out from the outside.

Leo: Very clever.

Steve: So the fact is, the original question that was asked back on Episode 120 was, if I encrypt it twice, with different keys, isn't that better than once? And it's absolutely the case that it is because, remember, somebody would be looking at the output from the second encryption, and the only attack is a brute force attack trying keys, you know, like a dictionary attack. And they would be looking for it to get plaintext out of the decryption. But the plaintext out of the second encryption is the encryption from the first, which means there is never going to be any plaintext. And as we've seen, the key spaces are such that there's just no chance another one of those keys, I mean, virtually no chance another one of those keys is going to magically perform the double encryption for you. That's just not - you have no access to the total number of mappings that are possible through a 128-bit block cipher.

Leo: It's a numbers game. It's just too vast a universe.

Steve: And really that's all crypto is. Crypto is just a numbers game. I mean, as we saw, the Romper Room cipher that uses 4-bit block size and 128-bit key, it's trivial because the numbers, there just aren't enough of them. It's a nice cipher. I like it. We were able to explain how it works. But it's just - it's not useful because there aren't enough bits to make it nontrivial. But as soon as you start increasing the width of the cipher, there are 2 to that many bits factorial possible mappings, it just goes out the window.

Leo: Wow. Well, you know what, I didn't think you could do it, but even my thick skull kind of gets it.

Steve: Yeah, I mean, I'm looking here at equation...

Leo: It's not that complicated, actually.

Steve: It's really not. And that's all that's going on in what has now been - it's the universal standard, this is what everyone's using, Europe apparently has adopted it, as we have here, as our new federal computing cipher. It was all done in an academic environment. It's been implemented all over the place in lots of different languages. The PPP system uses Rijndael, and it's been - on our software page, it's in virtually every language. So it's completely available because it's just not that hard to do.

Leo: Now, when you - okay, I'm going to ask a question about public key cryptography and how that's different from symmetric key in just a second. And I know that's another subject, but I just want to understand that. All right. So I gave you a little time because I know I gave you a touch one. Symmetric versus public key.

Steve: Yes. Okay. Or really the way to say it is symmetric versus asymmetric, that is, nonsymmetric. What it is that is symmetric, that is, the symmetricness is that the same key is used to encrypt as decrypt. And we just saw why because lord knows, who knows what other key could possibly provide a reverse mapping.

Leo: Good point, good point.

Steve: No one is going to come up with a key that's going to undo that nightmare of scrambling and shifting and mixing over and over 14 times. So you use exactly the same key. You expand that key to the internal key material. And basically you run Rijndael in reverse in order to step-by-step unscramble the data that was put in, in order to get back out what the encryption had put in. But the only way to decrypt it is to run it backwards.

Leo: But the negative of that is that both you and the person receiving the cipher have to have the key. So you have this difficulty of getting them the key.

Steve: Yes, exactly. And that's why it's regarded as a so-called, like a secret key approach, is that anybody who has the key that was used to encrypt it is able to decrypt it. So what's

different about an asymmetric cipher and the asymmetry is the keys. One key encrypts, and that key cannot be used to decrypt. So, I mean, literally it's a one-way function of encryption. And then only the matching, the key that was made, along with the encryption key comes a decryption key, although people who have really been paying attention will remember that there's no difference between them except that one is one and one is the other, meaning that you produce a key pair. One will undo what the other does. But that one can't undo what it does, nor can the other undo what it does.

Leo: And see, that's great because you can publish your public key, and anybody can send an encrypted message to you, but nobody can decrypt it except you. Is it less strong, then, as a result?

Steve: Well, yes. The algorithms fundamentally function differently. So, for example, public keys, where we were talking about 128 bits being all the strength you would ever need, public keys need to be 1024 bits in order to have the equivalent strength. And now there are 2048-bit public keys, again because the nature of the algorithm is such that there are different attacks on them. So to have, for example, a 128-bit symmetric key is about equivalent to a 1000-bit asymmetric key. And that's part of the reason that the process of using them is so slow, is there's just a much - much more work needing to be done for public key crypto. And it's just - it's absolutely infeasible to use it for encrypting, like, a whole file. And so people don't. What they do is, they will choose - they'll get a very good random number, and they use that as a symmetric key to encrypt the file. Then they use the public key cryptography, that is to say an asymmetric cipher, to encrypt that random number. So then all you have to do is give that random number which has been encrypted and this blob while is the file, you give that to someone. If they have the matching decryption key, they will take the random number which has been encrypted with the public key, decrypt it into the actual symmetric key which was used with a bulk encryption algorithm like Rijndael to encrypt the blob, and then they decrypt it.

Leo: Got it.

Steve: And there you go.

Leo: So that's why symmetric is used and is necessary for every kind of encryption.

Steve: Yes, exactly. You will always have that. And it's why I really thought it would be fun to kick off the new year by explaining exactly how Rijndael, which is now, you know, everyone's going to see it around. It's what Jungle Disk uses for its encryption. It's what, you know, we talked about that Omziff program last week, the cute little standalone. It's got a bunch of crypto in it. You could choose, I think, Twofish and Blowfish and some others, and Rijndael is also there.

Leo: So Rijndael has replaced Triple DES and all of these others? It's kind of the one everybody's using?

Steve: Well, yes. And in fact Triple DES, that's another example of where multiple encryption is good because that's all Triple DES is.

Leo: Yes, a 64-bit DES encryption done three times.

Steve: Exactly. Because 56 bits was not enough, you use a key which is three times 56 bits, and you use the first third and encrypt something, then you use the second third and encrypt it again with DES, and then you use the third third and encrypt it a third time. And by the time that 64 bits comes out, it is really confused. But that's a way of giving you the equivalent of a much larger key length is just by multiple symmetric encrypting with different keys. Which is really just what that guy back in Episode 120 asked us.

Leo: That's what he was doing. So that's interesting. So because Rijndael seems to me a little newer than the others that I've known about like Blowfish and DES, Triple DES.

Steve: Yeah. It is certainly - it is newer. It uses a state-of-the-art understanding of crypto and attacks on crypto. One of the other cool things, there's an attack known as a side-channel attack. And that is that some crypto algorithms give away information about what they're doing inside based on their power consumption, or the time it takes. So somebody who's really serious about attacking - this is all NSA sort of stuff - they can look at variations in the power consumed by the processor during encryption, and that will reveal bits of the key. Or because many ciphers do different things based on what the key is.

So again, the guys who did Rijndael said okay, we're aware of side-channel attacks. We're going to make what Rijndael does not key dependent. And so what's so cool is the only thing - and this is what's sort of mind-boggling is the only thing that the key is used for is producing that expanded set of blocks of key material which is XORed for every round, that all that internal stuff, that S-box, the shifting of the rows, the mixing of the columns, that never changes. That's not dependent upon the key. And so Rijndael was designed not to give away any information by someone looking at it, that is to say its timing never changes and its power doesn't change because the key is only used for XORing the data as it comes out of each cycle of round. It's never used to, like, go down different crypto pathways.

Leo: Very interesting.

Steve: It's all you need. And believe me, I'll be using it a lot in the future.

Leo: And obviously easy to implement because it's, I mean, it's just so simple.

Steve: Yes, exactly. It's very straightforward. And it's one of the things that makes it a modern cipher is there's nothing screwy where it's - remember the old cartoon with some guy, I guess it was Einstein, who was working on how E=MC squared. And he comes to the end, and he scratches his head, and he says, "And then a miracle happens."

Leo: And there you go. And now you have it.

Steve: Yeah. Rijndael has no miracles, it's just really simple, straightforward. And every step of the way these guys were able to say this is why we chose this polynomial. This is why we chose this S-box. This is why we're shifting the rows this way. And the result is a really, really strong scrambling of bits, a mapping between all these possible input bit combinations and output.

Leo: That's neat. Very cool. I'm glad, you know, I have to say I enjoy these shows where you really explain stuff so much. And we've kind of gotten away from it a little bit. So I'm glad we could do this, and I hope we'll do more of these where the basic technique - I guess what happened is early on we kind of covered all these basic technologies, and so we didn't need to go back and do more. But I love this, and I love learning how all this stuff works. And you're very good at explaining it.

Steve: I love explaining.

Leo: If you want more, including transcripts of this show or any of our 125 episodes, or 16KB versions, Steve keeps those on his site, GRC.com/securitynow. It's a great place to go because not only can you get all the information, the show notes and everything, you can also get Steve's free, and there are many of them, security utilities. You can use ShieldsUP to test your router, Unplug N' Pray to turn off Plug and Play, DCOMbobulator, Shoot The Messenger. That's where he keeps his Perfect Paper Password algorithms, all the different versions of that. And the forums, too, where you can contribute. And you can submit your questions at GRC.com/feedback. But don't forget, that's also where you'll find SpinRite, the world's best, everybody's favorite, my favorite, hard drive maintenance and recovery utility, GRC.com.

Steve: Now, I do want to remind everyone, please do not use the Romper Room cipher.

Leo: That would be bad.

Steve: That was presented for demonstration purposes only.

Leo: You know, it'd probably work, though, if your kid brother's not too swift. But other than that. Thank you Miss Nancy. Steve, we'll talk again next week with lots more stuff. And as we said, we're going to start covering some security news starting the next week, as well.

Steve: Absolutely.

Leo: Keep you up to date on what's going on in the security world. Thanks, Steve. Have a Happy New Year, and we'll see you next week.

Steve: Right-o.