CS 240: Asymmetric Key Encryption Transcript

[00:00:00] Now let's talk about asymmetric encryption algorithms.

[00:00:03] And these are oftentimes called public key algorithms for reasons that will be clear shortly.

[00:00:09] But these algorithms are asymmetric because the encryption and decryption keys are different.

[00:00:14] And it turns out that being able to use different keys for encryption and decryption, lets you do all manner of interesting things that uh we value. And so, let's talk about how these algorithms work a little bit more.

[00:00:28] So first of all, we have our encryption algorithm and the encryption key and the decryption keys are different that in and of itself is kind of uh interesting because how in the world do you create a pair of keys that somehow um work in tandem like that so that they reverse each other? So, I use one key to encrypt, use the other key to decrypt. Well, it turns out that these two keys must have a certain mathematical relationship so that they um are basically inverses of each other.

Start visual description. The professor demonstrates the concept of asymmetric encryption by explaining how the encryption and decryption keys are different and must have a certain mathematical relationship to work together. End visual description.

[00:01:05] And so these two keys have to be generated together. So, when we generate keys for a public key algorithm, we run a key generator algorithm that knows how to produce these key pairs that um can work together properly.

[00:01:21] And I guess that's the, the core idea to these algorithms.

[00:01:24] That was the great invention I suppose is that figuring out the mathematics that underlies these, these keys and how they can um invert each other and so on.

[00:01:34] But we don't have to worry about the uh the math that underlies all this.

[00:01:38] Um So these keys have to have a certain relationship. The other thing is, is that um it's important that if I give you one of these keys, but not the other one that based on the key that I gave you, you're not really able to, to figure out what the other key is.

[00:01:53] So it's infeasible to derive one of the keys from the other key.

[00:01:57] So that's another part, not only do they have to have a particular mathematical relationship, but it has to be infeasible to derive one from the other.

[00:02:06] And, and so you can imagine that was uh maybe a, a great research accomplishment to figure out how to do that.

[00:02:12] But that, that's kind of how it works. So, other than that um yeah, encryption algorithm where the, the encryption and decryption keys are different.

[00:02:24] Now, it's actually true that um we generate a pair of keys that work together.

[00:02:29] You can actually use either one of the keys to encrypt the data and then you would use the other one to decrypt the data and it doesn't really matter which, which key you pick for encryption or decryption as long as you use the other one for the opposite operation.

[00:02:42] And as you can see here in this, this diagram, Bob is still trying to send uh some data to Alice. I guess he's trying to get a date with her or something.

Start visual description. The professor shows a diagram where Bob is sending encrypted data to Alice using one key for encryption and the other for decryption, illustrating the practical application of asymmetric encryption. End visual description.

[00:02:51] And um so he takes his um his data that he wants to send his message, and he runs it through the encryption algorithm.

[00:02:58] He passes in one of the keys and then out comes the encrypted data and then sends the encrypted data to Alice, she would decrypt it with the, the companion key and then recover the original message.

[00:03:11] So that, that's, that's how it works.

[00:03:14] The most popular public key algorithms that are in use today are RS A.

[00:03:22] So the original inventor or creator of, of um asymmetric key encryption was a guy named Ron Raves actually uh Rast and some of his colleagues, Shamir and Edelman, they are credited with being kind of the uh the original inventors of, of this idea.

Start visual description. The professor introduces the RSA algorithm, mentioning its inventors and the key sizes (2048 bit and 4096 bit) used in this popular public key encryption method. End visual description.

[00:03:41] And so RSA is the name of their algorithm, and their RSA algorithm can use either 2048-bit keys or 4096-bit keys.

[00:03:54] And um obviously, you have to use their algorithm to generate the keys as well.

[00:04:01] So uh that, that's the most popular kind of public key encryption that that's out there.

[00:04:08] And in more recent years, we have elliptic curve cryptography. This is another um algorithm that has a different kind of mathematical foundation behind it.

[00:04:18] But it has the same properties that the keys uh are the inverses of each other.

[00:04:23] And for elliptical curve cryptography, there's different key sizes you can choose depending on uh how secure you want your encryption to be.

[00:04:33] And it's interesting to note that elliptic curve cryptography is used by the Bitcoin algorithms to do the encryptions that are required in the Bitcoin system.

[00:04:41] So that's just a kind of an interesting tidbit.

[00:04:48] So why is it called public key encryption? I've been using two words, asymmetric key and also public key.

[00:04:55] Um asymmetric key is perhaps more descriptive, but public key is what people usually call it.

[00:05:00] And the reason is that, well, the reason is that if I want people to send me data securely, what I would do is I would generate a, a key pair.

Start visual description. The professor explains the concept of public key encryption, describing how individuals generate a key pair, publish the public key, and keep the private key secret to enable secure data transmission. End visual description.

[00:05:12] So just imagine a world where every individual has their own key pair.

[00:05:17] So I'm going to generate a key pair and in this case, maybe it's an RS a key pair.

[00:05:23] And what I'm going to do is I'm going to take one of the keys and I'm going to publish it, make it public.

[00:05:28] So it's kind of like publishing your phone number.

[00:05:29] If people can't find your phone number, they can't call you.

[00:05:32] So it's common to publish your phone number so that people can call you well, in the same way you would take your public key and publish it.

[00:05:40] So anybody out there who wants to send you data securely can uh find out what your public key is.

[00:05:47] Now, the other key in that case would be your private key.

[00:05:50] And it’s very important that you keep the private key secret because if people get their hands on your private key, then they can decrypt your data.

[00:05:59] So the first thing I’m going to do as a, as an individual, I’m going to generate a key pair.

[00:06:05] I’m going to keep one of the key secrets and that’s my private key. I’m going to give the other key to everyone or at least anyone I want to be able to send me data securely and that’s the public key and it’s no longer a secret, it is public.

[00:06:19] And then if anybody ever wants to send me encrypted data, all they have to do is encrypt the data with my public key.

[00:06:28] And then when I receive the data, it’s going to be encrypted.

[00:06:30] But I’m the only one in the world who can decrypt it because I’m the only one that has my, the corresponding private key.

[00:06:36] So then I can go ahead and decrypt the data using the private key.

[00:06:41] And so you can kind of see how the, the public and private key work together, but it’s essential we keep one of those keys secret so that we’re the only one that can decrypt the data.

[00:06:54] So let’s look at um a code example here.

[00:07:07] So let’s go to the public key encryption demo.

[00:07:16] Now, the first thing we need to think about is uh generating a key pair.

Start visual description. The professor demonstrates generating an RSA key pair using Java's key pair generator class, specifying the key size and showing the process of creating and storing the key pair. End visual description.

[00:07:20] So you can't use this algorithm without key pair. And so, um there's a method in here called create RS a key pair and it shows you how to use the java libraries to generate a, a key pair.

[00:07:32] So in this case, I'm going to use Java's key pair generator class and ask it for an instance of the RS a key generator.

[00:07:40] So I want to generate an RS a key pair.

[00:07:43] And once I get back the RS a key generator, then I'm going to tell it what key size I want.

[00:07:47] I want a 2048 bit key.

[00:07:50] And then I just ask it to generate uh a key pair for me.

[00:07:53] And it will do all the, the mathematical algorithm necessary to generate an appropriate key pair and return it to me.

[00:08:01] And of course, the key pair class contains both the public and the private key.

[00:08:06] So that's how you generate a key pair.

[00:08:10] And then assuming you've, you've done that similar to AES, you could write a little method here that knows how to run Rs A on a body of data.

[00:08:20] And so in this case, it looks a lot like the, the code for running AES.

[00:08:25] We use Java cipher class to create an instance of the RS A cipher.

[00:08:30] The input parameters are similar as well cipher mode, encrypt or decrypt, input data, output data.

[00:08:37] And then we have the key that we're using to encrypt or decrypt.

[00:08:43] So we get an instance of the RS a cipher, we initialize it with the mode and the key we were given through the parameters.

[00:08:51] And then we just go through the process again of reading the data from the input, passing the data through the cipher update method.

[00:08:58] And every time we pass the cipher some data, it passes us back the encrypted data.

[00:09:04] And then we just write all that data to the output stream.

[00:09:07] And then at the end, we give the algorithm a chance to perform its final steps and then it gives us any remaining encrypted bytes that, that it has for us. And then we write those to the output.

[00:09:17] So it's, it's really just looks uh like the uh the method for running AES for the most part.

[00:09:25] So it's not hard to generate keys. It's not hard to run the algorithms.

[00:09:31] And um now one thing that is important is that when you generate a key pair like we did here, you're typically going to store your, your keys in a file somewhere.

[00:09:43] I mean, that's the only way you're going to remember what your private key is and what your public key is by storing them in, in a file.

[00:09:51] And so you're going to have to keep that file secure and make sure nobody ever gets it.

[00:09:56] And of course, the public key file would be published to the world potentially.

[00:10:00] But the private key need to be stored securely.

[00:10:06] So you can see in, in uh this program we create a key pair, then we get the public key and the private key out of the key pair.

Start visual description. The professor discusses the importance of securely storing the private key file and the potential risks if it is compromised, emphasizing the need for secure key management. End visual description.

[00:10:14] And then we save the two keys to files, name, public key and private key.

[00:10:20] So if we look over here, we can see the files.

[00:10:24] Well, I guess they don’t have it. Yeah, those are binary files, I guess.

[00:10:26] So they’re not really viewable easily.

[00:10:29] So we have the public key and private key files.

[00:10:32] And then you can see here before I uh take my data.

[00:10:35] So here’s my data that I want to encrypt four score. And seven years ago, our fathers bought fourth et cetera.

[00:10:42] I mean, you can see here that, that I’m going to encrypt the data and then decrypt it.

[00:10:49] But you’ll see up here that I can randomly select which key I use for encryption and decryption.

[00:10:55] So in this case, I’m going to use public key for encryption and private for decryption, but I could have easily made the opposite choice, and it would still work.

[00:11:03] So here’s the data that I’m going to encrypt. Here’s the plain text, convert the plain text to bytes and then pass the data into the RS A encrypt algorithm.

[00:11:16] Get back to the cipher texts and then I can decrypt it using the same process, take the encrypted data, pass it to RSA decrypt and uh get back the decrypted bytes.

[00:11:29] And so in this case, um the encrypted or the, the plain text is here, and we can see here that the decrypted file.

[00:11:41] Um Oh no, that’s the wrong one. That’s for a different example.

[00:11:44] So we actually get the same string back when we decrypt it so we can run it and improve that.

[00:11:59] So here’s the encrypted data and then here’s the data that we got back after we decrypted it. So, we did indeed get the original data back, which is important.

[00:12:11] OK. So, it just gives you a sense of how you would uh use uh the RS A algorithm in A, in an application.

[00:12:26] Now, now that we’ve looked at public key encryption a little bit, it does have some big disadvantages that you need to be aware of.

[00:12:36] First is that typically uh a public key encryption algorithm can only be used to encrypt a, a small amount of data.

[00:12:44] For example, RS A, the RS A algorithm can only be used to encrypt data up to the size of the key.

[00:12:53] So if I have a 2048 bit key, that means I can only encrypt up to 2048 bits of data or with a 4096-bit key, I could encrypt up to 4096 bits of data.

[00:13:05] And that’s a, that’s a very small amount of data.

[00:13:08] So you can’t actually encrypt very much with um these algorithms. Typically.

[00:13:15] The other disadvantage of public key algorithms is that they’re much slower than symmetric key algorithms.

[00:13:21] For example, AES is much faster than RS A or elliptical curve cryptography. And so symmetric key is just faster.

[00:13:33] And um the other disadvantage of public key encryption is you do have this private key file that you do need to store securely and make sure it never gets shared with others.

[00:13:42] Because if somebody gets your private key, then um your data is pretty compromised at that point.

[00:13:49] So that begs the question if, if public key encryption has all these disadvantages, why would you ever use it? And it turns out that public key encryption has some very important um applications. So, two of them are symmetric or secure symmetric key exchange, which we’ll talk about soon and also digital signatures. So, both of these applications use public key algorithms at their core.

[00:14:18] And as you’ll, you’ll understand when we talk about them that these are two very important things.

[00:14:22] So it does turn out that uh public key algorithms, even though they can’t encrypt much data and they’re slow.

[00:14:29] It turns out that they are a critically important discovery in computing that really has changed the world in pretty significant ways.

[00:14:38] A lot of the security that we depend on every day is at its heart dependent on public key encryption.

[00:14:44] And so public key cryptography is probably one of the greatest inventions, most important inventions in the history of computing.

[00:14:50] So it is important, but we need to talk about um symmetric key exchange and digital signatures before you'll, you'll see why it's so important.