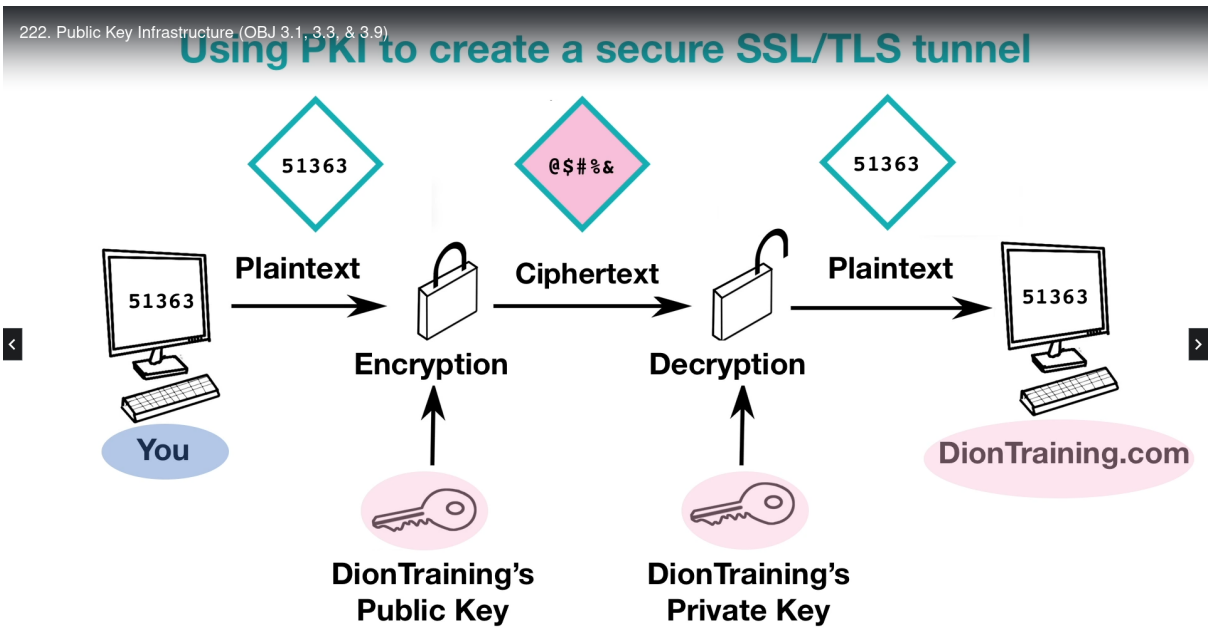
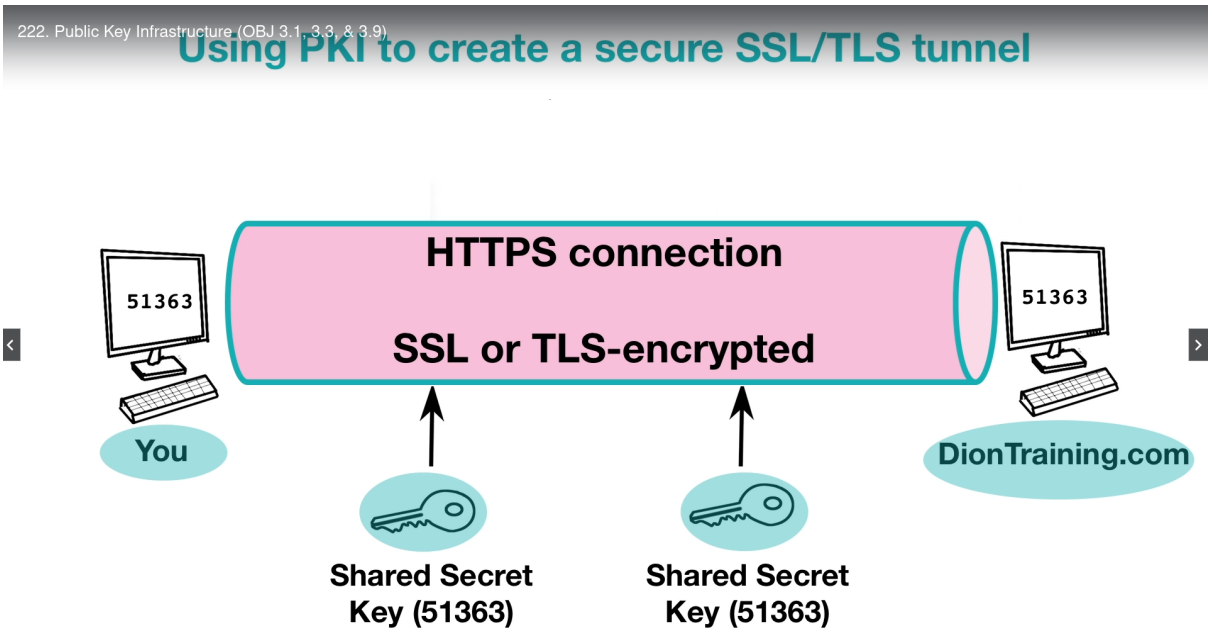
PKI

Now, Public Key Infrastructure is an entire system of hardware, software, policies, procedures, and people that is based on asymmetric encryption. If you have ever connected to a website using an https connection, you've been part of PKI. Now, if you want to establish a secure connection to diontraining.com, you would enter https://www diontraining.com into your web browser. Your web browser would go to a trusted third party called the certificate authority, and they're going to ask them for a copy of my web server's public key. Then, your web browser is going to pick a random long number to use as a shared secret key for use with the symmetric algorithm, something like AES that we're going to use for bulk encryption of the data between your browser and my web server. But, you have to get that random shared secret to my web server securely, and for that, we're going to use Public Key Encryption, known as asymmetrical encryption. Now, using my public key, your computer is going to encrypt that random shared secret key that you've created. In the example here, I'm using 51363 as our shared secret. Now, once you encrypt that using my server's public key, which anyone in the world has access to, it's then going to be sent over the Internet. Now, because it's encrypted with my public key, though, no one on the Internet can decrypt it unless they have my private key. And the only person who has that is me. So, as we go across the Internet, no one can see the fact that it's 51363 as that secret code. Now, once my web server receives the encrypted cipher text, it's going to use my server's private key to decrypt it and get it back to that shared secret key that you submitted. And now that I have it in plain text, I know what that number is, that 51363. So far, this is just using asymmetrical encryption, like we discussed in the last section.



Now, both you and my server know this shared secret key, though. So, we can create a symmetric tunnel. We can do that by using something like AES to create a secure TLS or SSL tunnel over the Internet and communicate safely and securely from anybody's prying eyes.



PKI & Public Key Cryptography

Well, PKI and Public Key Cryptography are closely related, but they are not the same thing. When we talk about PKI, this is the system that creates the asymmetrical key pairs that consist of those public and private keys that are used in the encryption and decryption process, as well as managing those key pairs to make sure they're valid and can be trusted. When we talk about Public Key Cryptography, on the other hand, we're just talking about the encryption and decryption process. So, it's a small part of the overall PKI architecture.

Remember, PKI uses Public Key Cryptography to do its function, but PKI is the entire system of things that are done to be able to create the secure connection from end to end. Now, when we talk about Public Key Encryption, on the other hand, it's just the asymmetric encryption and decryption piece.

Certificate Authority

For all of this to occur successfully, we need to have a trusted third party involved, though. This trusted third party is known as a certificate authority. These certificate authorities are going to issue digital certificates, and these certificate authorities are also going to keep the level of trust between all of the certificate authorities around the world. In this section of the course, we're going to focus on all of those other parts of the process that allow PKI to work, including those certificate authorities.

Digital Certificate

A certificate is a digitally-signed electronic document that binds a public key with a user's identity. Now, when I talk about a user here, the user can be a real live person like you and I or it can be a server, a work station, or another device for the purposes of a digital certificate. These certificates commonly use the X.509 standard for digital certificates. This is the common standard used inside of PKI and the certificates contain the owner's or user's information like their name, their organization, or even their public key and it also is going to contain the certificate authority's information. The certificate authority is the trusted third party who is going to issue these digital certificates, and therefore, the certificate is also going to contain their name, their digital signature, their serial number for that certificate, the issue date and the expiration dates, and the version of the certificate.

SAN

For example, I own diontraining.com but I also own jasondion.com. Now, if I wanted to use one certificate to cover both of those domains because they don't have the same root domain, I would have to modify the Subject Alternate Name or the SAN field. Now, the SAN field in a certificate specifies what additional domains and IP addresses are going to be supported by that certificate. Two other types of certificates that we have to think about are single-sided and dual-sided certificates.

Single-sided - Dual-sided Certificates

Two other types of certificates that we have to think about are single-sided and dual-sided certificates. Now, for example, when you connect on my website there's a secure session that's established and my server's going to identify itself to your web browser using my server's digital certificate. Now, you aren't required to have your own digital certificate to be authenticated back to me, though. This is known as a single-sided certificate because only one side of this authentication is happening with the certificate. Now, some organizations require both the server and the user to validate each other using certificates. When this occurs, this is called a dual-sided certificate. Now, using dual-sided certificates, it's better for security but it does require twice the processing power on the server, so, it's usually only used in high security environments. Now, with digital certificates, each certificate is validated using the concept of a chain of trust, moving from the bottom upward.

Certificate Encodings

As I said before, digital certificates are usually based on the X.509 standard but the certificate itself must be encoded before it can be used. Now, there are three different encoding methods that are classified under the X.690 standard. They're known as BER, CER, and DER.
BER is the Basic Encoding Rules and it's the original ruleset governing the encoding of data structures for certificates. But there are several different encoding types that can be used as part of BER. Now, for the Security+ exam, you don't need to know the specific encoding types underneath BER. So, we're not even going to cover them here but just realize that BER has the ability to have multiple encoding types. And that makes it different than CER.
CER is the Canonical Encoding Rules, which is a restricted version of BER that only allows the use of one encoding type.
DER is the Distinguished Encoding Rules. And this is another restricted version of BER, and it only allows one encoding type, as well, but it has more restrictive rules

for length, character strings, and how a particular element of a digital certificate is stored. In fact, DER is what is used commonly for X.509 encoding of certificates.

Certificate Formats

When dealing with digital certificates you may come across a few different file types on your machine, including the PEM, CER, CRT, KEY, P12, PFX, and P7B.

The .pem format is used for Privacy-enhanced Electronic Mail and it uses the DER encoding method. Sometimes, it also stores itself as a .cer, .crt, or .key file.

The .p12 file is going to be used to store a server certificate, an intermediate certificate, and a private key in one encrypted file. It's called the .p12 because it's a binary format of the Public Key Cryptographic System #12 or PKCS#12 certificate.

Now, the .pfx file is called the Personal Information Exchange and it's used by Microsoft for release signing. This file is going to contain both the private and public keys in it.

The .p7b file is used as the basis for S/MIME, the secure email protocol. And this is also going to be used for single sign-on. It's called the .p7b because it's based on the PKCS#7.

Certificate Authority

For a digital certificate to be issued, a user first has to request a digital certificate from a Registration Authority known as an RA. The Registration Authority, then, requests the identifying information from the user and forwards that certificate request up to the CA known as the Certificate Authority. This Certificate Authority then creates the digital certificate, including the user's public key and their identity information, and passes that back to the user. There are many root certificate authorities out there including companies like Verisign, Digisign, and numerous others. They act as a trusted third party to validate the certificates are being issued to the correct people. The certificate authority also maintains a publicly-accessible copy of that user's public key and this allows them to have that for use by other users who wish to send them confidential information.

CRL

Now, they also maintain what’s known as a CRL which is a Certificate Revocation List. The Certificate Revocation List is an online list of digital certificates that the certificate authority has already revoked. Usually, this is because those certificates have become comprised. The Certificate Revocation List is a full list of every certificate that has ever, ever been revoked by that particular certificate authority. Whenever your computer tries to connect to a new server, it requests the current public key digital certificate from the certificate authority. The certificate authority first checks the Certificate Revocation List before they send you that public key or digital certificate to ensure it hasn't already been revoked.

OCSP

Now, if you want to determine if a certificate was revoked, we're going to use a protocol known as the OCSP or Online Certificate Status Protocol. This protocol is going to allow you to determine the revocation status of any digital certificate using its serial number. This is an alternative to the Certificate Revocation List and operates much more quickly and much more efficiently because it doesn't use encryption, but that makes it less secure. Just as OCSP was an alternative to Certificate Revocation List, OCSP Stapling is an alternative to OCSP. This process used to be known as the TLS Certificate Status Request Extension. This OCSP Stapling allows the certificate holder to get the OCSP record from the server at a regular interval and include it as part of the SSL or TLS handshake. By doing so, it eliminates an additional connection being required at the time of the user's request and this speeds up the secure tunnel creation process. Now, one concern with digital certificates is if an attacker can impersonate a server.

Public Key Pinning

Now, one concern with digital certificates is if an attacker can impersonate a server. To prevent this from occurring, public key pinning was created. Public key pinning allows an HTTPS website to resist impersonation attacks from those who are trying to present fraudulent certificates by presenting a set of trusted public keys to the user's web browser as part of its HTTP header. Now, if the web browser doesn't get the matching public key from the certificate authority, then, it knows that website was compromised and it's going to alert the user.

Key escrow

Key escrow occurs when a secure copy of a user's private key is held, just in case that user accidentally loses their key. If your organization simply can't accept any data loss, then you need to ensure key escrow has been setup. Now, remember, whenever you use key escrow, you have to protect that key store from anybody who's trying to steal those keys. It's recommended that key escrow services require two different administrators be present anytime a key is being taken out of escrow.

Key Recovery Agent

A key recovery agent is a specialized type of software that allows the restoration of a lost or corrupted key to be performed. Think of it as a backup for all of the certificate authority's keys, just in case an incident or disaster occurred.

Web of Trust

The web of trust is a decentralized trust model that addresses issues associated with the public authentication of public keys within a CA-based PKI system. One of those issues is that you have to pay to get one of these digital certificates from a CA. Now, with a web of trust, we instead use a peer-to-peer model, where I trust you and you trust me, and because of that, we now can give that trust to other people as we go around. So, how do we know who we're going to be able to trust, when there's no third party? Well, one of the ways we can do it is by trusting somebody just because they said so. So, if I have a web server and I want you to trust it, I can install a self-signed certificate. That says hey, trust me because I said I'm Jason and you can trust me. Now, you have to decide if you're really going to trust me, though. If you see one of these self-signed certificates, your web browser's going to give you a error, like this one in Firefox. Now, you can choose to trust them by clicking on the I understand the risks, or you can say, you know, I don't trust that. I'm going to go to a different website and get my information there. For security purposes it's not a good idea to trust a self-signed certificate and so, this is kind of frowned upon. You should probably, if you're having a website, you should spend the money and get a real digital certificate from a trusted third party.

The second thing we can do is trust the collective intelligence of others. This is the system that's used by Pretty Good Privacy. It's basically a web of trust, where every person who trusts you starts helping to increase your rating and then, as more

people know you and trust you, other people are going to know you and trust you. The same thing kind of happens on Twitter and Facebook and other social media.