blogs.akamai.com

# Enterprise Security - SSL/TLS Primer Part 1 - Data Encryption

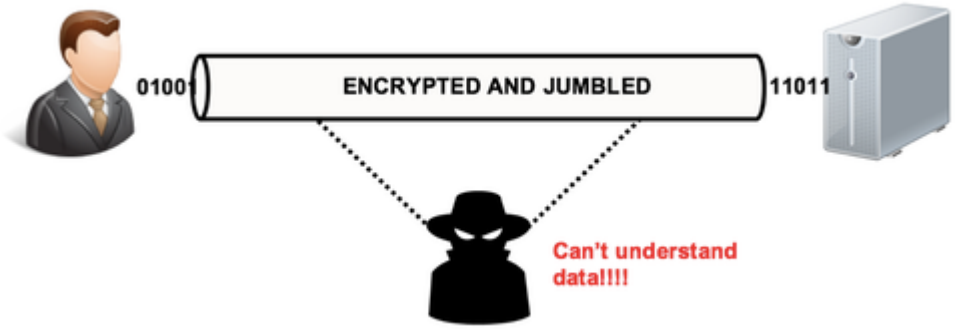
5-6 minutes

Join me over the next few posts as I talk about how to provide fast, reliable, and secure applications in the branch while protecting end-users and promoting a transparent and open Internet. Let's start with the basics.

**So what is SSL/TLS & how does it work?**

TLS is used to secure communication between two parties. Originally called Secure Sockets Layer (SSL) and later changed to Transport Layer Security (TLS), it utilizes both asymmetric cryptography as well as symmetric cryptography to provide data privacy, integrity, and authentication.

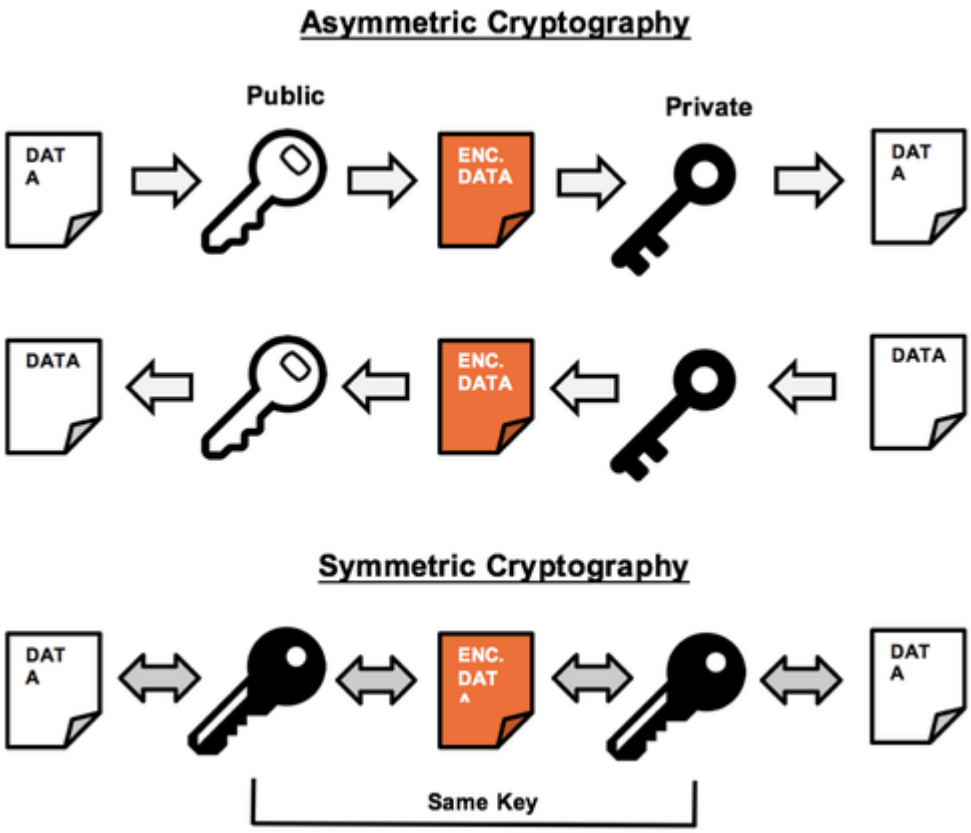
This means that two peers can exchange information with a reasonable expectation that a man in the middle is unable to read the contents of their messages. In addition, the peers are able to authenticate to ensure they really are talking to whom they think.



## What is the difference between asymmetric and symmetric cryptography?

In asymmetric cryptography, there are two keys: a public and a private. Any data encrypted with one key is theoretically only decryptable with the other key. A party will distribute their public key to everyone in the world. Whenever someone wants to send encrypted data to that party, they need only encrypt it with the public key, knowing that it will only be decryptable with the private key. Since the private key is kept private with the original party and never distributed, privacy is ensured.

In symmetric cryptography, the same key is used to both encrypt and decrypt data. This means that both parties need only the single key to both encrypt and decrypt data. It also means that distributing that single key is extremely dangerous, as anyone that gets the key has access to everything, where as with asymmetric cryptography, the private key never leaves the owner and safety is ensured.




**Why use symmetric if asymmetric cryptography is safer to distribute keys for?**

This is a commonly asked question, and the answer comes down to compute resources. It turns out that asymmetric cryptography is extremely taxing on a CPU. Symmetric algorithms, however, are very lightweight and often implemented directly in the silicon of chips today.

**How does TLS work?**

TLS has two phases: The asymmetric phase and the bulk data encryption (symmetric phase). In this series of blog posts, I will discuss TLS in terms of how it is commonly deployed for HTTPS, since it is easiest to think about it in terms of this real and practical use case.

The purpose of the asymmetric phase is to authenticate the web

server (and optionally the end user if client certificate mutual authentication is requested) and to negotiate a set of keys for the symmetric phase.

This makes sense if you think about it. Asymmetric as we said earlier, allows for easy key distribution, but is harsh on CPU resources. So TLS uses the heavyweight asymmetric phase to establish a safe authenticated channel and then uses that channel to send a set of dynamically generated symmetric keys safely to the peer that needs it.

At that point, both sides use the much faster and more efficient symmetric keys to communicate, knowing that they are the only parties in the world with the newly generated keys.

**How does this differ from HTTPS?**

HTTPS is simply HTTP inside of a TLS session. A TLS session is established between two peers and then normal HTTP interactions occur exactly the same inside of it. This means if you were able to peek into the inside of a TLS session, you would see an exchange that is indistinguishable from normal HTTP (with some minor edge caveats that are not important for this document).

**What other protocols does TLS encrypt?**

TLS is a TCP based session layer protocol. It can be used to encrypt any protocol as long as both sides agree to use it as the session layer. Today, many protocols utilize TLS as their encryption layer. As a matter of fact, many protocols masquerade as HTTPS in order to break out of firewalls.

Imagine this common scenario in an enterprise: A network

administrator sets up the company firewall to only allow desktops to communicate over port 80 (HTTP) and port 443 (HTTPS). This network administrator feels confident that this will stop protocols from being used that he or she does not wish to exist on the network. What many applications do today to get around this is design their communication protocol to use TLS as the encryption layer over port 443. This lets the data egress through the firewall due to it allowing port 443 traffic, and any deep packet analysis will be thwarted because the data stream is encrypted and the firewall has to assume it's normal HTTPS.

If that sounds sneaky, it is. And everyone uses it. A popular web conferencing platform for example will try this very trick to escape a network. So the lesson is that there are many protocols that utilize TLS because it is an industry standard encryption session layer protocol, and that many of those very protocols also use it over port 443 to look like HTTPS purposefully.

Watch for the next post in this series where we will discuss certificates, man in the middle, and the impact of BYOD.