SSL stands for secure socket layer, TLS stands for Transport Layer Security, both are protocols and both are used to provide *Secure Communication* between two parties (for example client and server). TLS is a successor of SSL and was introduced due to SSL’s security shortcomings, for example susceptibility to the POODLE attack which compromised SSL 3.0 version. At the time of writing this article (July 2019) SSL is still supported by Chrome, IE and Mozilla, however it is anticipated that it will be dropped completely in near future.

Key phrase from above paragraph is *Secure Communication,* this term goes beyond nuts and bolts (algorithms and hardware) of computers and enters the realm of higher level of abstraction that probably does not have a specific name hence it is probably quantum mechanics. In practice this means that the same principles, mechanisms and solutions which guarantee Secure Communication in computer networks will also work in communication between two medieval kings who use pigeons, letters, seals, codes etc instead of protocols, algorithms, certificates and so on.

Secure Communication is regarded as communication where Authentication, Confidentiality and Data Integrity are ensured. Authentication is met when two parties are sure that the person they are talking are indeed who they claim they are. Confidential communication is when nobody can eavesdrop on both of you, data integrity means that message leaving the sender does not change when it is received by the receiver.

It is clear that the principles of Secure Communication go beyond two computers talking to each other using bytes and algorithms and all the fancy stuff if we consider following real life example. When two humans speak to each other Authentication principle is met by both of them physically seeing and hearing each other. Data Integrity is ensured by hearing the other person well as well, having a language that enforces certain rules ( “Man eating chicken” vs “Man-eating chicken” or “Let’s eat Grandma! vs Let’s eat, Grandma”), asking for repetition if something was not clear, nodding to show that you understand. Confidentiality would be ensured if people talking were sure that nobody listens to them, or that they use previously agreed upon code words that only they know the true meaning of (like in spy movies).

What follows as well from Secure Communication principles is that there is no single way of ensuring it because different systems will require different approaches. There is a difference when you use web browser to check the weather vs when you are checking your bank account balance. Therefore terms like Crypto System or Security System, or Security Architecture, Cryptographic Suite, Cryptographic Scheme, Public Key Infrastructure arise. All of those are high level solutions for the problem of ensuring Secure Communication (or its components - Authentication, Data Integrity and Confidentiality) in specific systems, requiring specific approaches.

However, there are some building blocks that pop up in almost every Security Architecture and it is worth to understand the fundamental aspects of them, which are universal.

Symmetric Cryptography and Asymmetric Cryptography.

Symmetric (aka private key cryptography) is a solution which allows to encrypt and later decrypt given message using the same Key. What is important in our discussion is only two high level features of this solution, namely:

1. Symmetric Cryptography is faster than Asymmetric Cryptography - Asymmetric Cryptography (Public Key Cryptography) is just more computationally expensive ( <https://crypto.stackexchange.com/questions/5782/why-is-asymmetric-cryptography-bad-for-huge-data> )
2. Symmetric Cryptography has a huge drawback of making sure that the Key used stays private - that only the two parties communicating has the Key. Without ensuring that ( protecting the private key) it is not possible to state if Confidentiality is kept.

Despite the huge disadvantage of Protecting the Private Key, private key cryptography is widely used (for example in TLS communication) due to the huge advantage - speed. In case Private Key is compromised and leaks somewhere immedietaly Authentication aspect and Confidentiality Aspect are compromised - you can’t tell if person who has a private key is indeed the person who should have the private key, if somehow you are sure that you are talking to a person you want to talk to, you can’t be sure if someone is eavesdropping on you - looking at encrypted messages you send to each other and decrypting it with their stolen Private Key. Here is more comprehensive overview of the term [2]

Asymmetric Cryptography (aka Public Key Cryptography) addresses the problem of Protecting the Private Key. In this approach two Keys are present - each Key undoes what the other does. Therefore if Key A is used to encrypt the message, the complementary Key B can decrypt it. If Key B is used to encrypt the data then Key A can decrypt it. Application of this feature result in one Key being regarded as Private and the other as Public, Private Key is being used to decrypt the data and Public Key is used to encrypt it (easy way to remember is to think that everyone can encrypt something but only few people can decrypt something).

It is clear that Public Key encryption ensures (or gives good reasons to believe in) Authenticity of the party who posses the Private Key since the hazardous operation of exchanging was not performed. Being sure that only the receiver can decrypt the message you send ensures that no one is eavesdropping on your messages (Confidentiality).

It is worth noting that both Private and Public Key cryptography schemes rely on mathematical properties of specific functions. For example at the heart of RSA Public Key encryption algorithm lies a mathematical problem of solving (finding d for given e,n,m) following equation:



here public key consists of modulus n and the number e, private key is d, our message is m, note that e,n,d cannot be chosen arbitrarily. Question that may come up is “why don’t I just arbitrarily choose a number or keyword that I will use for encryption? If I choose it arbitrarily and no one knows it then it should be secure.” The answer is not - it would not be secure, if you choose arbitrarily a number 100 as your Private Key and multiply each message by 100 then this code will be very easy to break.

Utilizing public key cryptography gives us two out of three features necessary for Secure Communication - Authentication (we trust a person who has the Private Key) and Confidentiality (no one is decrypting the message we sent). The next thing we have to ensure is Data Integrity. We need to make sure that once we send a message it is not the case that someone took it, changed it and passed further to the Receiver, or that it took the Receiver’s response, changed it and sent it to us. This problem is solved by (SSL) Certificates, Private Keys and Digest Values of exchanged messages - this is how SSL works.

Certificates were introduced to facilitate trusting of unknown websites - it is good to know if website you are trying to enter, or host you are trying to connect to is regarded as trusted. Browsers trust websites who will present them a Certificate that browser respects. Certificate contains a Digital Signature of some Certificate Authority, browser looks at the Digital Signature, checks if the Certificate truly corresponds to the Digital Signature (digital signatures allow for that) and if it does and if Digital Signature’s author (Certificate Authority) is respected by the browser, the host is regarded as trusted.

Certificates are small files containing information about an entity that requested the certificate (like date of issue, name of company, name of Certificate Authority issuing the certificate, company location, expiration date of the certificate etc.). To get a certificate a company has to issue a Certificate Request and send it to the Certificate Authority, CA will vet the company and issue the Certificate or not. Thanks to the vetting process the Certificate serves as an entity which binds a physical company, person or server to a digital file. Depending on the type and importance of the company requesting the key the vetting process may be very comprehensive and the whole procedure may involve real humans traveling around the globe to provide a company with a Certificate.

Digital Signatures that are part of the Certificate are piece of bytes that allow to check if the Certificate was changed or not and if it was issued by the Certificate Authority. First thing is covered by Digital Signature containing the Digest Value of the Certificate, second thing is covered by the fact that in order to get the Digest Value a person looking at the Certificate must decrypt it with Verifying Key. The process of creating Digest Value is similar to encrypting the message with Private-Public Key cryptography. in Digest Value we are dealing with signing vs verifying keys instead of private and public keys. Signing key is used to encrypt the Digest Value (create Digital Signature), verifying Key is used to decrypt the Digest Value.

Note that therefore a key pair is required to generate a Certificate (private/public or signing/verifying) therefore what can be and actually is done is that “Certificates are used to sign other Certificates”. If you create yourself a key pair use one of the keys to sign the certificate (provide the Digital Signature) then

The browsers trust the certificates that are stores in OS trust store - list of trusted certificates end up on your computer whenever you install an OS. It is possible to add/remove given certificate to/from a trust store.

In order to handle the problem of the list of certificates the browser trusts becoming too long what is actually happening is that browser checks if the trust-chain of the certificate presented ends up in an OS root store. Certificates can be signed using other certificates

Digest Value is a value that uniquely represents the message we sent, it is computed from our message and appended to it prior to sending to the receiver. It is very difficult to find two different messages which have the same Digest Value. Leveraging

Look here for more details of the algorithm [3]

---------------------------------------------------------------------------------------------------------------------------

[1] <https://en.wikipedia.org/wiki/Digital_signature>

[2] <https://en.wikipedia.org/wiki/Symmetric-key_algorithm>

[3] <https://en.wikipedia.org/wiki/RSA_(cryptosystem)>

is keep private and not shared.

Digital Signatures - A digital signature is generated by the hashing of a digital certificate. The hash is then encrypted using the CA`s private key. This encrypted md5 string is then appended to the public certificate as a digital signature.

Digital Signatures

Digital Signature is a mathematical scheme for verifying the authenticity of digital messages [1]. The key word in the definition above is *authenticity* - what Digital Signature does in practice is ensuring that indeed message M came from sender S. This is achieved utilizing *asymmetric cryptography* - in a scenario where sender S sends message M to receiver R and wants his message to contain Digital Signature D following approach applies:

1. Sender S creates public-private key pair - in Digital Signature model known as signing and verifying key respectively.
2. Sender S computes the hash (or Digest Value) of his message.
3. The Hash value of message and signing key are provided to the Digital Signature Algorithm.
4. DSA computes the signature and appends it to the message.
5. Receiver receives message (encrypted or not) with appended Digital Signature.
6. Receiver computes his own Hash of the message.
7. Receiver uses Signer’s Verifying key to retrieve the Hash of the message computed by the Signer.
8. Receiver compares the two Hash values - if they are equal it means that Signature that came with the message indeed corresponds to this message.

SSL Certificates contain Digital Signatures of entities that verified an entity that requested the certificate.