GAZİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BM402 BİLGİSAYAR AĞLARI

Gamze Aksu

171180005

ÖDEV-4

MART 2022

İÇİNDEKİLER

Sayfa

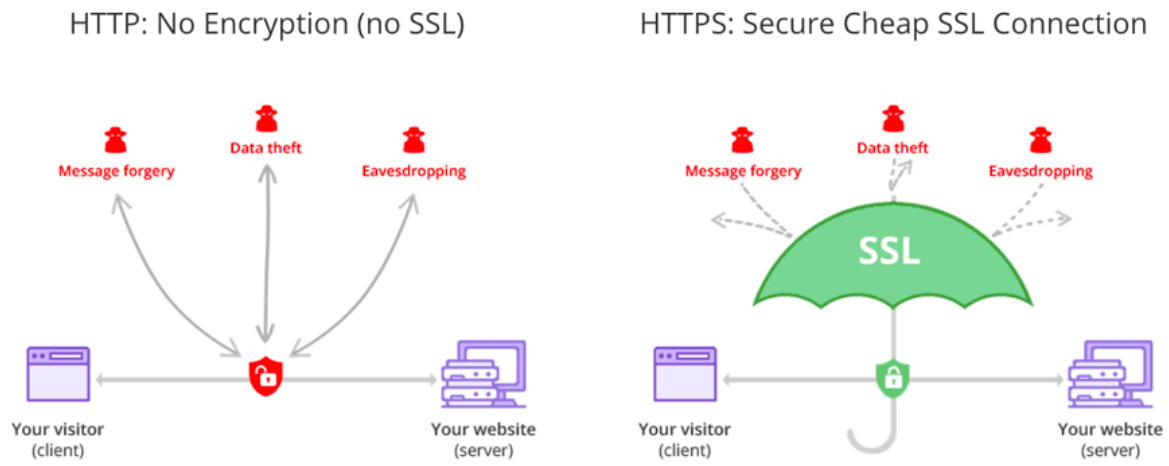
1. SSL/TLS

1.1. SSL/TLS Nedir?

SSL açılımı Güvenli Yuva Katmanıdır (Secure Socket Layer). SSL bir ağ üzerinden geçen iletişimin şifreli bir hale getirilmesini ifade eder. Bir şifreleme protokolüdür. SSL 90'lı yıllarda Netscape tarafından tanımlanmıştır. İnternet üzerinde web sunucuları ile web istemcileri arasındaki iletişimde veri bütünlüğünü, gizliliğini ve kimlik doğrulama sağlamak amacıyla gerçekleştirilmiştir. İnternet bağlantılarının güvenliği dışında bir de ağ katmanı üzerindeki diğer uygulamalar için de kimlik doğrulama ve şifreleme için SSL protokolü kullanılmıştır. SSL protokolü sayesinde bir web tarayıcı (istemci) ile web sitesi (sunucu) arasındaki iletişim güvenli hale getirilmiştir. Bir tarayıcıdan bir web sitesine gönderilen veriler ya da bir web sitesinden bir tarayıcıya gönderilen veriler dış tehdit aktörleri tarafından ele geçirilebilir. Bu gerçekleştiğinde tehdit aktörünün eline veriler şifreli bir şekilde geçer. [1]

TLS protokolün açılımı Aktarım Katmanı Güvenliğidir (Transport Layer Security). TLS protokolü uluslararası bir standartlar kuruluşu olan İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force - IETF) mühendisleri tarafından ilk versiyonu 1999 yılında yayınlanmıştır. SSL protokolünün gelişmiş bir versiyonudur. [2]

SSL/TLS protokolü kullanan web sitelerinde HTTP (Hypertext Transfer Protocol) yerine HTTPS (Hypertext Transfer Protocol Secure) protokolü kullanılır. Bir kullanıcının bir web sitesinde SSL/TLS protokolünün kullanılıp kullanılmadığını anlayabilmesi için adres çubuğuna bakması yeterlidir. Burada SSL/TLS protokolü kullanılan web sitesi adresi HTTP yerine HTTPS ile başlar ve adres çubuğunun yanında bir asma kilit ikonu bulunur.



Şekil 1: HTTP ve HTTPS karşılaştırması

1.2. SSL/TLS Tarihçesi

1.2.1 SSL Tarihi

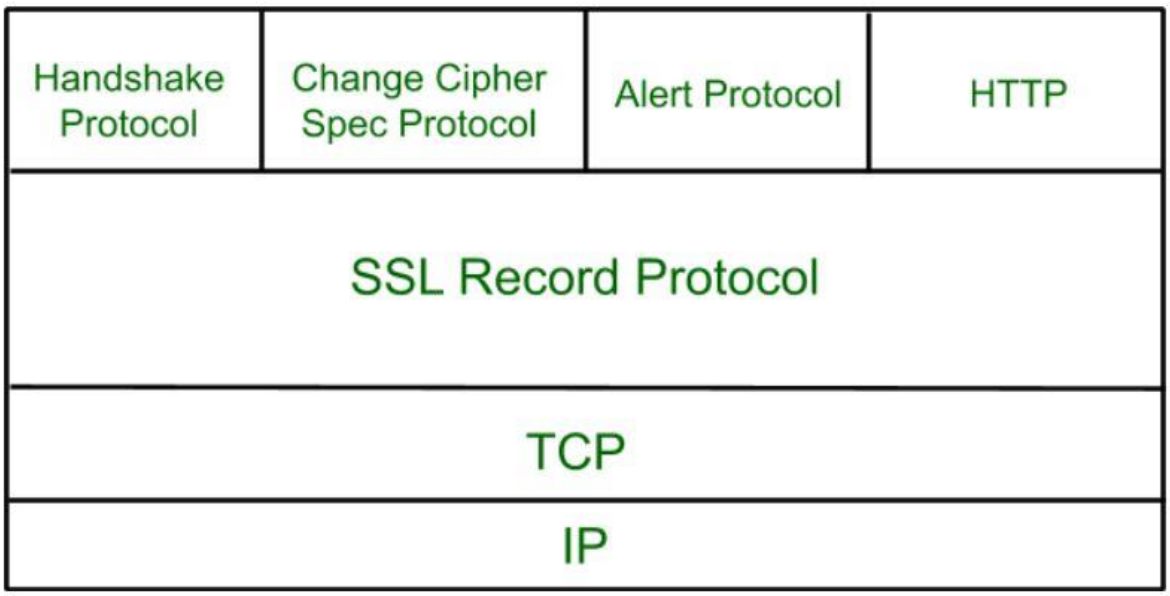
SSL ilk olarak Netscape tarafından geliştirilmiştir. İnternette şifreli mesajlaşma amaçlanmıştır. Ancak güvenlik nedeniyle SSL versiyon 1.0 hiçbir zaman piyasaya sürülmemiştir. Daha sonra SSL versiyon 2.0 1995 yılında yayınlandı. Yayınlandıktan sonra birçok güvenlik ve kullanılabilirlik kusur ortaya çıktı. SSL versiyon 2.0'da aynı kriptografik anahtar ile hem şifreleme hem de kimlik doğrulama işlemleri gerçekleştiriliyordu. Ayrıca ortadaki adam saldırılarına bir zafiyet olarak hem açılışta bir el sıkışma hem de mesaj kapanışı koruması yoktu. SSL versiyon 3.0 1996'da piyasaya sürüldü. Daha önceki versiyondan kalan sorunlar çözüldü ve internetin çalışma şekli değiştirildi. 2011 yılında SSL versiyon 2.0 kullanımdan kaldırıldı, 2015 yılında ise SSL versiyon 3.0 kullanımdan kaldırıldı. Bunun sebebi 2014 yılında SSL blok şifrelerini etkilen bir saldırının (POODLE) ortaya çıkmasıydı. [3] [4]

1.2.2. TLS Tarihi

TLS, SSL üzerinde gerçekleştirilen iyileştirmeler sonucu geliştirilmiştir. TLS versiyon 1.0 1999 yılında piyasaya sürülmüştür. SSL versiyon 3.0 gelişmiş bir versiyonu olmasına rağmen beraber çalışamayacak kadar önemli değişiklikler gerçekleştirilmiştir. Daha sonra TLS versiyon 1.1 2006 yılında ortaya çıkmıştır. Ancak iki sene sonra 2008 yılında piyasaya sürülen TLS versiyon 1.2 sayesinde çok fazla benimsenmemiştir. TLS versiyon 1.2'de şifreleme MD5-SHA1 kombinasyonundan SHA-256 hash fonksiyonuna geçilmiştir. 2018 yılında TLS versiyon 1.3 piyasaya sürüldü. Şu an kullanılabilir halde olan protokoller TLS versiyon 1.2 ve TLS versiyon 1.3 protokolleridir. [3][4]

1.3. SSL/TLS Protokolü Yapısı

Şekil 2'de görüldüğü gibi SSL/TLS protokolü TCP protokolünün üstünde olacak şekilde tasarlanmıştır. Uygulama katmanı ile Taşıma katmanının arasında yer alır. İki farklı katmandan oluşacak bir şekilde tasarlanmıştır. SSL/TLS protokolü güvenliği sağlamak için TCP protokolünü kullanır. Bu yaklaşımın bir avantajı olarak hizmet verilen uygulamanın özel ihtiyaçlarına göre yeniden uyarlanabilir olması söylenebilir. [5]

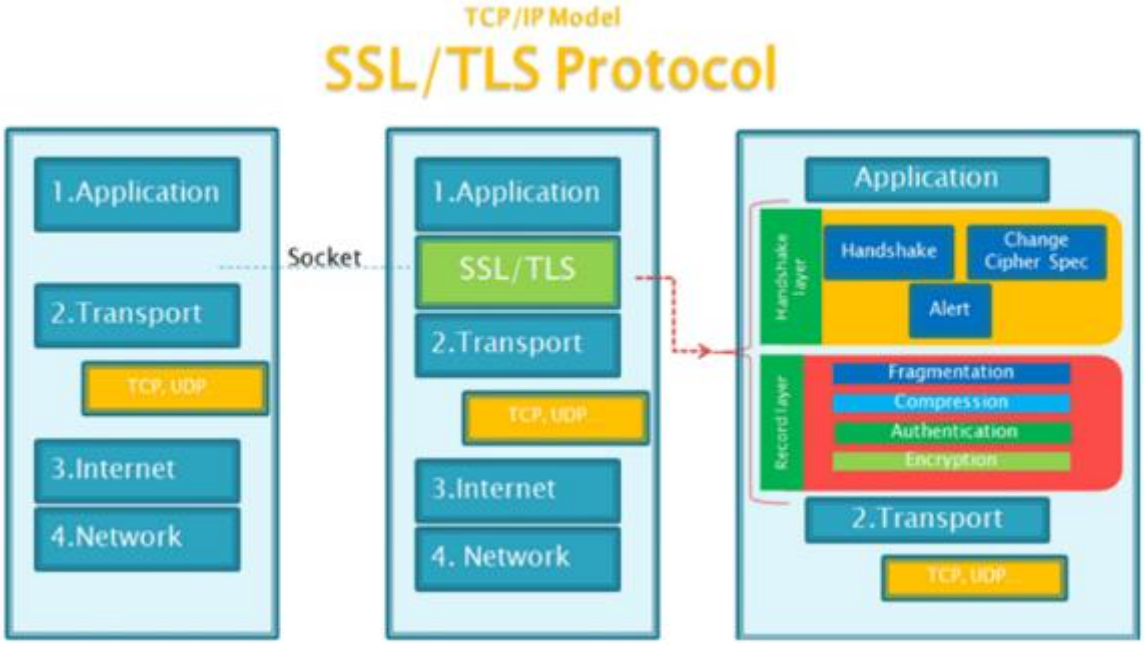


Şekil 2: SSL/TLS Protokol Yığını

SSL/TLS protokolü dört farklı alt protokolden oluşur. Şekil 3'te SSL/TLS protokollerinin ayrıntılı bir şeması görülmektedir. Bunlar aşağıda listelenmiştir:

1. **SSL Kayıt Protokolü (SSL Record Protocol):** SSL kayıt protokolü SSL bağlantısı için gizlilik ve veri bütünlüğü sağlar. SSL kayıt protokolünde ilk olarak veriler parçalara ayrılır. Bu aşamaya fragments denir. Daha sonra bu parçalar sıkıştırılır bu sıkıştırma işlemi isteğe bağlıdır. Sonra SHA (Secure Hash Protocol) and MD5 (Message Digest) gibi şifreleme algoritmaları ile oluşturulan MAC (Message Authentication Code) sıkıştırılmış olan parçaya eklenir. Daha sonra şifreleme adımı gerçekleştirilir. En sonunda da SSL başlığı eklenir. Kısaca adımlar listelenecek olursa Parçalama (Fragments), Sıkıştırma (Compression), MAC ekleme (Authentication), Şifreleme (Encryption).
2. **SSL El sıkışma Protokolü (SSL Handshake Protocol):** SSL handshake protokolü oturum oluşturmak için kullanılır. Bir nevi kimlik doğrulama da denilebilir. İstemci ve sunucu arasında mesajlar gönderilerek kimlik doğrulama işlemi gerçekleştirilir.
3. **Change – Cipher Protokolü:** El sıkışma tamamlanmadığı sürece SSL kayıt çıktısı bekleme durumunda kalacaktır. El sıkışma tamamlandıktan sonra bekleme durumu mevcut duruma değiştirilir. Bu protokol bu durumda devreye girer. Bu protokolün amacı bekleme durumunda olan kayıt çıktısını mevcut duruma kopyalanmasını sağlamaktır. Change-Cipher protokolü 1 bayt uzunluğunda ve yalnızca bir değeri olabilen tek bir mesajdan oluşur.

4. **Uyarı Protokolü (Alert Protocol):** Uyarı protokolünde her mesaj 2 bayttan oluşur. SSL protokolü ile ilgili oluşan uyarıların iletilmesi için kullanılır. Uyarı protokolü Level (1 bayt) ve Uyarı (1 bayt) olmak üzere iki farklı bölümden oluşmuştur. Level bölümü mesajları iki farklı şekilde sınıflandırılmıştır. İlki uyarı (warning) mesajıdır. Bu uyarı gönderici ile alıcı arasındaki iletişimde etkisi yoktur. Diğeri ise önemli hata (fatal error) mesajıdır. Bu durumda gönderen ve alıcı arasındaki bağlantı kesilir.



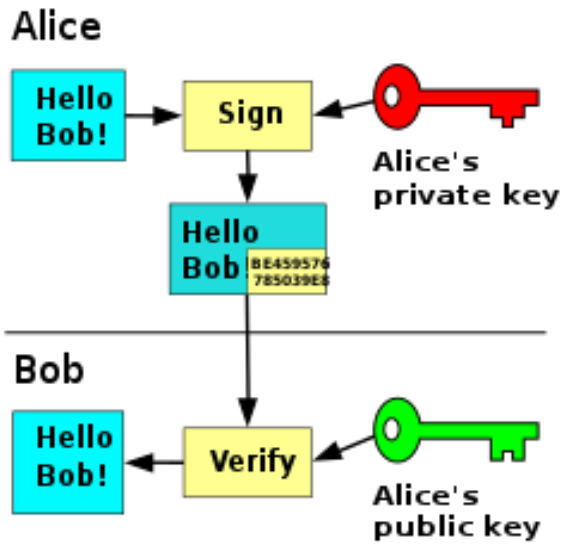
Şekil 3: SSL/TLS Protokolü

1.4. SSL/TLS Sertifikası

Bir SSL/TLS sertifikası bir dijital belgedir. SSL/TLS sertifikası, bir X.509 sertifikası türüdür. Bir web sitesinin kimliğinin bir genel anahtar ve bir özel anahtardan oluşan bir anahtar çiftine bağlanmasını sağlar. Sertifikada bulunan genel anahtar sayesinde bir web sunucusuyla şifreli bir oturum başlatılabilir. Özel anahtar sunucuda saklı tutulur ve belgeleri dijital olarak imzalamak için kullanılır. Ek olarak bir SSL/TLS sertifikası bir web sitesi hakkında alan adı da dahil olmak üzere tanımlayıcı bilgileri ve isteğe bağlı olarak sitenin sahibi hakkında tanımlayıcı bilgileri içerir. Bir web sunucusunun SSL/TLS sertifikası genel olarak güvenilir bir sertifika otoritesi (Certificate Authority - CA) tarafından imzalanmış ise son kullanıcılar web sitesine güvenebilirler. [6]

1.5. SSL/TLS Çalışma Prensibi

SSL/TLS protokolü açık anahtarlı şifrelemeye dayanan bir çalışma prensibine sahiptir. SSL/TLS protokolünde ilk olarak kullanıcı internet sunucusundan güvenli bir bağlantı isteğinde bulunur. Daha sonra sunucu kullanıcıya sertifikasıyla birlikte açık anahtarını (public key) gönderir. Sonra kullanıcının tarayıcısı sununun gönderdiği sertifikanın güvenilir bir sertifika olup olmadığını kontrol eder. Burada sertifikanın güvenilir bir sertifika otoritesi (Certificate Authority - CA) tarafından imzalanıp imzalanmadığını ve geçerliliği kontrol edilir. Sonra kullanıcının tarayıcısı rastgele bir simetrik şifreleme anahtarı üretir ve sunucunun açık anahtarını (public key) kullanarak bu simetrik şifreleme anahtarını şifreler ve bağlanmaya çalıştığı sunucuya gönderir. Sunucu kendi açık anahtarı (public key) ile şifrelenmiş olan simetrik şifreleme anahtarını kendi özel anahtarıyla (private key) çözerek simetrik anahtarı elde eder. İnternet sunucusu bundan sonra kullanıcıya göndereceği verileri elde etmiş olduğu bu simetrik anahtarı kullanarak gönderir. Kullanıcı aynı simetrik anahtarla internet sunucusundan gelen verileri çözerek internet sayfasını güvenli bir şekilde görüntüler. [7] [3]



Şekil 4: Açık Anahtarlı Şifreleme

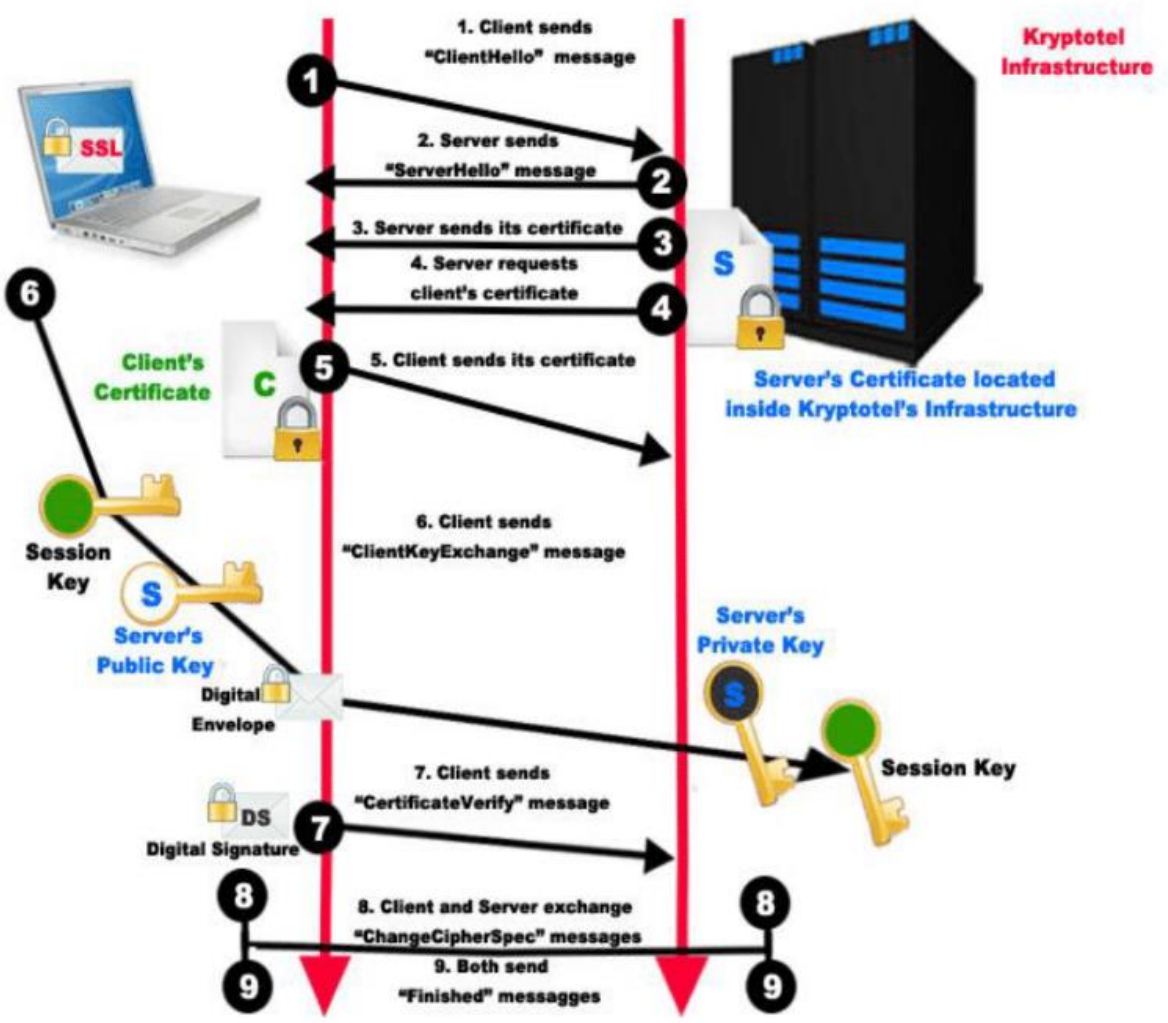
Bir istemci sunucu ile bağlantı kurarken şifreli bir bağlantı istediği zaman şifreli bağlantı istediğini belirtmesi gerekmektedir. Çünkü protokoller SSL/TLS ile ve SSL/TLS olmadan olarak iki farklı şekilde çalışır. İstemcinin şifreli bağlantı istediğini belirtmenin iki farklı yolu vardır. Bunlardan ilki farklı bir port numarası kullanmaktır. Örnek olarak HTTP protokolü 80. porttan hizmet verirken HTTPS protokolü 443. porttan hizmet vermektedir. Bir diğer yöntem ise normal bir port numarası kullanılırken SSL/TLS protokolü için özelleştirilmiş

mekanizmaların kullanılmasıdır. İstemci özelleştirilmiş mekanizma kullanarak sunucunun bağlantıyı SSL/TLS protokolüne yönlendirmesi için istekte bulunur. Örnek olarak mail ya da haber protokolleri için STARTTLS mekanizmasının kullanılması verilebilir.

1.6. SSL/TLS Handshake Süreci

İstemci şifreli bir bağlantı istediğini belirttikten sonra el sıkışma süreci başlar. El sıkışma süreci kaynak [3] ve [8] den yararlanılarak yazılmıştır. SSL/TLS protokolünün el sıkışma aşaması üçlü DES, RC4, RC2, DES gibi şifreleme algoritmalarının kullanıldığı varsayılarak anlatılacaktır. El sıkışma sürecinde ilk olarak istemci, kendi SSL sürüm numarasını, şifre ayarlarını, oturuma özgü verileri ve sunucunun istemciyle SSL kullanarak iletişim kurması için ihtiyaç duyduğu diğer bilgileri sunucuya gönderir. Daha sonra sunucu kendi SSL sürüm numarasını, şifre ayarlarını, oturuma özel verileri ve istemcinin sunucuyla SSL üzerinden iletişim kurmak için ihtiyaç duyduğu diğer bilgileri istemciye gönderir. Bununla birlikte sunucu sertifikasını da gönderir. Eğer istemci şifreli bir kaynağa ulaşmak istiyorsa o zaman sunucu istemciden sertifika talebi de gönderir. İstemci, sunucunun kimliğini doğrulamak için sunucu tarafından gönderilen bilgileri kullanır. Doğrulama gerçekleşmezse istemci bilgilendirilip bağlantı kesilir. Daha sonra istemci oturum için sunucu işbirliğiyle ve şifreleme algoritmasına göre değişen bir ikincil paylaşılan gizli veri oluşturur, bunu sunucunun açık anahtarını kullanarak şifreler ve bu şifrelenmiş ikincil paylaşılan gizli veriyi sunucuya gönderir. El sıkışma sürecinde isteğe bağlı olarak sunucu istemciden kimlik doğrulaması talep ettiyse istemci ayrıca bu el sıkışmaya özel ve hem istemci hem de sunucu tarafından bilinen başka bir veri parçasını imzalar. Sunucu istemcinin kimliğini doğrulayamazsa oturum sona erer. İstemci kimliği doğrulanabilirse sunucu ana gizli anahtarın şifresini çözmek için kendi özel anahtarını kullanır daha sonra bir dizi adımlar takip ederek esas paylaşılan gizli veriyi oluşturur. Ardından istemci de gizli veriden başlayarak bu adımları tekrarlar. Daha sonra hem istemci hem sunucu elde ettikleri esas paylaşılan gizli veriyi kullanarak simetrik oturum anahtarları oluşturur. Bu anahtarlar ile oturum boyunca şifreleme ve şifre çözme işlemleri gerçekleştirilir. Daha sonra istemci sunucuya, istemciden gelecek olan mesajların oturum anahtarı ile şifreleneceği bilgisini içeren bir mesaj gönderir. Ardından, el sıkışmanın istemci kısmının bittiğini belirten ayrı (şifreli) bir mesaj gönderir. Daha sonra aynı şekilde sunucu istemciye bundan sonra gelecek mesajların oturum anahtarı ile şifreleneceği bilgisini içeren bir mesaj gönderir. Ardından, el sıkışmanın sunucu kısmının bittiğini belirten ayrı (şifreli) bir mesaj gönderir. Bu şekilde SSL/TLS protokolünde el sıkışma süreci bitmiş ve oturum açma işlemi gerçekleşmiş olur. Bu

oturum içerisinde istemci ve sunucu verileri şifrelemek ve şifreli veriyi çözmek için oturum anahtarlarını kullanırlar. Bu, güvenli kanalın normal çalışma koşuludur. Herhangi bir zamanda, dâhili veya harici uyaran nedeniyle her iki taraf da bağlantının yeniden kurulmasını talep ederek yeniden bu süreci tekrar edebilirler.



Şekil 5: SSL/TLS Handshake Süreci

1.7. SSL/TLS Amaçları

1. **Gizlilik:** SSL/TLS protokolünde veriler şifrelenerek gönderildiği için verilerin gizliliği sağlanır. Kredi kartı numaraları, şifreler ve kişisel bilgilerin başkaları tarafından ele geçirilmemesi gerekmektedir.
2. **Bütünlük:** Verilerin bütünlüğünün korunmasını ifade eder. Veri ağ üzerinden gönderilirken arada bir tehdit aktörünün verinin içeriğini değiştirmemesi çok önemlidir. Örneğin kredi kartından çekilecek para miktarı gönderilirken arada birinin bu miktarı değiştirememesi gerekmektedir.
3. **Kimlik Doğrulama:** Kullanıcının bir sisteme erişirken kim olduğunu doğrulamasına denir.
4. **Kriptografik Güvenlik:** SSL/TLS protokolü istemci ve sunucu arasında güvenli bir bağlantı oluşturmayı amaçlar.
5. **Birlikte Çalışabilirlik:** Birbirinden bağımsız programcıların SSL/TLS protokolü kullanarak şifreleme parametrelerini başarılı bir şekilde değiş tokuş yapabilecek uygulamalar geliştirebilmeleri hedeflenmektedir.
6. **Genişletilebilirlik:** SSL/TLS protokolü gerektiğinde yeni ortak anahtar ve toplu şifreleme yöntemlerinin protokole dâhil edilebilecek esnek bir yapıda olmayı hedeflemektedir.
7. **İlişkisel verimlilik:** Kriptografik işlemler genel olarak çok fazla işlemci gücü tüketmektedir. Bu yüzden SSL/TLS protokolü sıfırdan kurulacak olan bağlantı sayısını azaltmayı amaçlar. Bunun için isteğe bağlı olarak oturumların önbelleğe alınması sağlanır.

KAYNAKÇA

1. Cobb, M. Loshin, P. (2021). SSL (secure sockets layer) https://www.techtarget.com/searchsecurity/definition/Secure-Sockets-Layer-SSL
2. https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/
3. Wikipedia katılımcıları (2022). Transport Layer Security. *Vikipedi, Özgür Ansiklopedi*. https://tr.wikipedia.org/wiki/Transport_Layer_Security
4. Yackel, R. (2020). What is SSL? Understanding the History of SSL and How it Works https://www.keyfactor.com/blog/what-is-ssl/
5. Bhakhra, S. (2021). Secure Socket Layer (SSL). https://www.geeksforgeeks.org/secure-socket-layer-ssl/
6. What is SSL?. (2021). https://www.ssl.com/faqs/faq-what-is-ssl/
7. Wikipedia katılımcıları (2021). Açık anahtarlı şifreleme. *Vikipedi, Özgür Ansiklopedi*. https://tr.wikipedia.org/wiki/A%C3%A7%C4%B1k_anahtarl%C4%B1_%C5%9Fifreleme
8. Elnaggar, A. (2015). Secure Socket Layer. 10.13140/RG.2.1.2671.3044. https://www.researchgate.net/publication/283297122_Secure_Socket_Layer