**GÜVENLİ İLETİŞİM PROTOKOLLERİ**

**SSL/TLS**

**HAZIRLAYAN: ESMANUR HURMA**

**Artificial Intelligence Security Lab (AISECLAB) aiseclab.org**

**Haziran / 2023**

**Tarihçe**

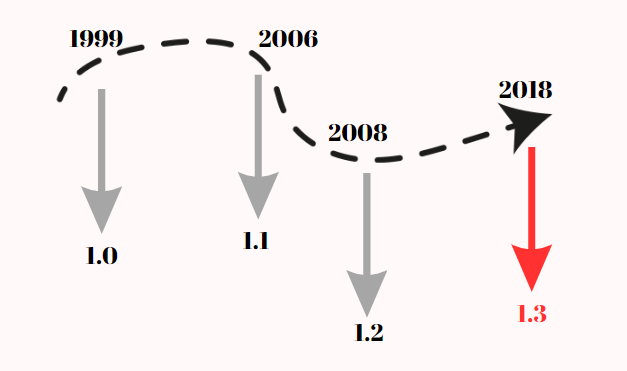
Web’in ilk zamanlarında güvenlik pek önemsenmiyordu. HTTP (Hypertext Transfer Protocol) güvensiz bir protokoldür. Ancak bilgisayar bilimcileri web'i finansal veriler gibi hassas iletişimler için kullanmada güvenliğin önemli olduğunu fark ettiler. Netscape, SSL (Secure Sockets Layer) protokolünü sürüm 1.0'dan itibaren geliştirdi ama önemli güvenlik açıkları nedeniyle hiçbir zaman yayınlanmadı. 1995 yılında SSL 2.0 sürümü piyasaya sürüldü ve yaygın olarak kullanılmaya başlandı. Ancak, güvenlik açıklarıyla birlikte geldi ve daha sonra 1996 yılında SSL 3.0 ile değiştirildi. SSL 3.0, önceki sürümlere göre sadece küçük bir gelişim değil, tam bir revizyondu. Bu sürüm, RFC 6101 olarak yayınlanmıştır.

TLS 1.0, SSL 3.0'a bir yükseltme olarak 1999 yılında piyasaya sürüldü. Ancak, SSL 3.0 ile uyumlu değildi. TLS 1.0, ileti kimlik doğrulaması ve bütünlüğü için GOST karma algoritması desteği gibi yeni özellikler eklendi. Önceki sürümlerde sadece MD5 ve SHA-1 desteklenen karma ileti kimlik doğrulama kodlarına alternatif olarak kullanıldı.

TLS 1.0, Nisan 2006'da TLS 1.1 ile değiştirildi. TLS 1.1 geliştirilmiş başlatma vektörleri ve AES için şifre blok zinciri desteği gibi bir dizi şifreleme iyileştirmesi getirdi.

Ağustos 2008'de ise TLS 1.2, RFC 5246 olarak piyasaya sürüldü. Bu sürüm, önceki sürümlere göre birçok iyileştirme içeriyordu.

TLS 1.3 Ağustos 2018'de RFC 8446'da tanıtıldı ve bir dizi güvenlik iyileştirmesi getirildi.



Şekil 3: TLS sürümleri ve tarihleri

**DİJİTAL İMZALAR**

Dijital imzalar temel olarak bir asimetrik şifreleme algoritmasını kullanarak veri parçalarını şifreler ve bu süreci tersine çevirir. Gönderenin ortak anahtarıyla şifrelenen belirli veri parçaları, herkesin bu şifreyi çözebilmesini ve doğrulayabilmesini sağlar. Dijital imzaların iki türü vardır: doğrudan ve tahkim edilebilir. Her birinin kendine özgü güçlü ve zayıf yönleri bulunur ve bu nedenle farklı durumlarda kullanılırlar. Önemli olan, hangi imza türünün daha iyi olduğu değil, belirli bir durum için hangi imzanın daha uygun olduğudur.

* Doğrudan Dijital İmza

Doğrudan imza, yalnızca iki tarafı (gönderen/imzalayan ve alıcı) içeren bir imza türüdür. Bu imza genellikle iki yöntemle oluşturulur. İlk yöntem, gönderenin özel anahtarıyla tüm mesajı şifrelemektir. İkinci yöntem ise gönderenin özel anahtarıyla şifrelenen bir şifreleme karması, ileti kimlik doğrulama kodu veya HMAC oluşturmaktır. Her iki durumda da alıcı, iletiyi şifresini çözmek/doğrulamak için gönderenin ortak anahtarını kullanır. Doğrudan imza, gönderenin kimliğinin üçüncü taraf doğrulamasını gerektirmez. Bu imza türü genellikle e-posta iletişimlerinde kullanılır. Doğrulama, alıcının, iletiyi doğrulamak için gönderenin e-posta adresine ve iddia edilen gönderenin ortak anahtarını kullandığı gerçeğine dayandırılır. Eğer e-posta taklit edilmişse (yani gerçekten iddia edilen gönderen değil, e-posta adresini taklit eden başka biri tarafından gönderilmişse), ileti imzası aslında iddia edilen gönderenden farklı bir anahtarla yapılmış olacaktır ve bu nedenle doğrulama başarısız olacaktır. Temel süreç Şekil 1'de gösterilmiştir.

Bu bölümde daha önce açıklanan tipik bir işlem aşağıdaki adımları içerir:

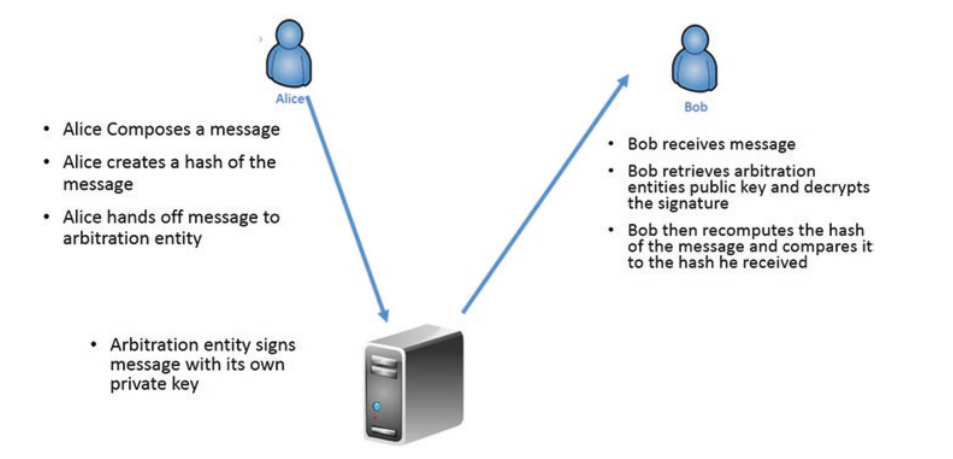
1. Mesajı yazın: İmzalanacak ve şifrelenecek olan mesajı hazırlayın.
2. Mesajın bir karma değerini oluşturun: Mesajın bütünlüğünü sağlamak için bir karma (hash) değeri hesaplayın. Bu, mesajın benzersiz bir temsilidir ve imzalama işleminde kullanılır.
3. Karmayı gönderenin özel anahtarıyla imzalayın: Gönderen, mesajın karma değerini özel anahtarıyla imzalar. İmzalama işlemi, mesajın bütünlüğünü ve gönderenin kimliğini doğrulamaya yardımcı olur.
4. Mesajı alıcının ortak anahtarıyla şifreleyin: Bu adım, mesajın sadece alıcının özel anahtarıyla çözülebileceği anlamına gelir.
5. Mesajı gönderin: İmzalı ve şifrelenmiş mesajı alıcıya gönderin.



Şekil 1: Doğrudan Dijital İmza

* Tahkimli Dijital İmza

Tahkimli dijital imza, doğrudan imzaya benzer şekilde çalışır, ancak bir farkla; iletiyi imzalayan varlık gönderen değil hem gönderen hem de alıcı tarafından güvenilen bir üçüncü taraftır. Bu üçüncü taraf, imza işlemine katılır ve gönderenin kimliğini doğrulamak için gerekli adımları gerçekleştirir. Böylece, alıcıya gönderilen ileti gerçekten iddia edilen gönderenden geldiği konusunda daha fazla güvence sağlanır.



Şekil 2: Tahkimli Dijital İmza

**DİJİTAL SERTİFİKALAR**

Asimetrik kriptografinin gücü, anahtar değişimi ile ilgili bir sorun olmamasıdır. Hem gönderenin hem de alıcının açık anahtarlarına erişebilmesi tamamen kabul edilebilir. Ancak, bu tür açık anahtarların nasıl yayıldığı sorusuna cevap vermez.

Bu süreçteki en önemli adım dijital bir sertifikadır. İki ana dijital sertifika türü vardır: X.509 ve PGP.

Her iki sertifika türü de dijital imzalar ve güvenli iletişim için asimetrik kriptografiyi kullanır. Dijital sertifikalar, güvenilir üçüncü taraflar (sertifika otoriteleri) tarafından imzalanır ve doğrulanır, böylece alıcılar sertifika sahibinin kimliğine güvenebilir ve iletişimi güvenli bir şekilde gerçekleştirebilir.

**X.509**

X.509 dijital sertifikalarda yer alan biçim ve bilgiler için uluslararası bir standarttır. X.509 dünyadaki en yaygın dijital sertifika türüdür. Sertifika Yetkilisi (CA) olarak bilinen güvenilir üçüncü taraf tarafından imzalanmış bir ortak anahtar içeren dijital bir belgedir. X.509 standardı 1988 yılında piyasaya sürülmüştür. O zamandan beri revize edilmiştir ve en son sürümü X.509 v3, RFC 5280'de belirtilmiştir. Bu sistem, sertifika sahibi hakkında bilgi sağlamanın yanı sıra bu bilgilerin güvenilir bir üçüncü tarafça doğrulanmasını da destekler. Bu SSL ve TLS gibi protokollerin güvenliğini sağlamanın temelidir.

X.509 sertifikasının gerektirdiği içerik aşağıdaki gibidir:

* Sürüm: X.509'un hangi sürümünün kullanıldığını belirtir. Bugün genellikle sürüm 3 kullanılmaktadır.
* Sertifika sahibinin ortak anahtarı: Sertifika sahibinin açık anahtarı, sertifikada bulunur ve açık anahtarlar bu şekilde dağıtılır.
* Seri numarası: Sertifikayı benzersiz bir şekilde tanımlayan bir numaradır.
* Sertifika sahibinin ayırt edici adı: Sertifika sahibinin benzersiz veya tanımlayıcı bir adıdır. Genellikle bir web sitesinin URL'si veya e-posta adresi gibi kullanılır.
* Sertifikanın geçerlilik süresi: Çoğu sertifika 1 yıl geçerlidir, ancak kesin süre bu alanda belirtilir.
* Sertifikayı verenin benzersiz adı: Sertifikayı veren güvenilir üçüncü tarafın adını tanımlar.
* Düzenleyenin dijital imzası: Sertifikayı verenin gerçekten o olduğunu doğrulamak için dijital imza kullanılır. Bu imzaya sahip olmak, sertifikanın geçerli olduğunu gösterir.
* İmza algoritması tanımlayıcısı: İmzayı doğrulamak için kullanılan imza algoritmasını tanımlar. İmzayı doğrulamak için imzalayanın açık anahtarı ve kullandığı algoritma bilgisine ihtiyaç duyulur.

**PGP**

PGP (Pretty Good Privacy) kriptografik karma, asimetrik kriptografi ve simetrik kriptografiyi birleştiren bir yöntemdir. Her ortak anahtar bir kullanıcı adı ve/veya e-posta adresiyle ilişkilendirilir bu nedenle PGP genellikle e-posta şifrelemesi için kullanılır. PGP kendinden imzalı dijital sertifikalar kullanır. Bu sertifika yetkilisinin olmadığı ve dolayısıyla gönderenin kimliğinin üçüncü taraf tarafından doğrulanmadığı anlamına gelir. Bu nedenle PGP sertifikaları genellikle web sayfaları veya e-ticaret için kullanılmaz.

PGP sertifikası aşağıdaki bilgileri içerir:

* PGP Sürüm Numarası: PGP'nin kullanılan sürümünü belirtir.
* Sertifika Sahibinin Ortak Anahtarı: Sertifika sahibinin ortak anahtarı, ortak anahtarların dağıtılmasını sağlar.
* Sertifika Sahibinin Bilgileri: Sertifikanın sahibinin kim olduğunu belirlemek için kullanılan bilgilerdir.
* Sertifika Sahibinin Dijital İmzası: PGP sertifikaları, sertifika sahibi tarafından imzalandığı için kendinden imzalıdır.
* Sertifikanın Geçerlilik Süresi: Sertifikanın ne kadar süreyle geçerli olduğunu belirtir.
* Tercih Edilen Simetrik Şifreleme Algoritması: Asimetrik kriptografi kullanıldıktan sonra, sertifika sahibinin simetrik şifreleme için tercih ettiği algoritmayı belirtir.

**AÇIK ANAHTAR ALTYAPISI**

Açık Anahtar Altyapısı (PKI), dijital sertifikaların oluşturulması ve dağıtılması için gereken altyapıdır. Dijital sertifikalar, asimetrik algoritmalar için açık anahtarların dağıtımında kullanılan araçlardır ve bu nedenle PKI, asimetrik şifrelemenin önemli bir bileşenidir. PKI'nın rolü, açık anahtarları bir sertifika yetkilisi (Certificate Authority- CA) aracılığıyla belirli bir kullanıcının kimliğiyle ilişkilendirmektir. Yani, ortak anahtarların yaygın olarak bulunması yeterli değildir, belirli bir ortak anahtarın belirli bir kullanıcıyla ilişkilendirildiğini doğrulayan mekanizmalara ihtiyaç vardır. PKI, bu doğrulamayı sağlayan bir CA tarafından gerçekleştirilir.

PKI'nın birkaç bileşeni vardır. Sertifikaların birbirinin yerine geçebilmesi için her sertifika veren, diğer sertifika verenler tarafından güvenilir olmalıdır. Örneğin, bir çevrimiçi banka sitesini ziyaret ederken, site tarafından kullanılan dijital sertifikaya bazı sertifika otoriteleri tarafından güvenilmesi gerekmektedir. Daha sonra, farklı bir e-ticaret web sitesini ziyaret edebilirsiniz ve bu site tamamen farklı bir sertifika otoritesi tarafından sağlanan bir sertifikaya sahip olabilir, ancak yine de güvendiğiniz bir otorite tarafından sağlanmalıdır.

Sertifika yetkilisi, sertifikaların verilmesi ve yönetilmesinden sorumludur. Bu süreç sertifikaların iptal edilmesini de içerebilir. Sertifikaların iptal edilmesi için iki yöntem bulunur:

• CRL (Certificate Revocation List- Sertifika İptal Listesi): Bu, iptal edilen sertifikaların bir listesidir. Bir sertifika çeşitli nedenlerle iptal edilebilir.

• Durum Kontrolü: CRL'lerin gerçek zamanlı olmadığı düşünüldüğünde, OCSP (Online Certificate Status Protocol- Çevrimiçi Sertifika Durum Denetimi Protokolü), bir sertifikanın hala geçerli olup olmadığını doğrulamak için kullanılabilen gerçek zamanlı bir protokoldür. OCSP, RFC 6960 standardında tanımlanmıştır.

**TCP Üçlü El Sıkışma**

Üç yönlü el sıkışma, istemci ve sunucu arasında güvenilir bir bağlantı kurmak için bilgisayar ağında kullanılan temel bir işlemdir. İnternet üzerinden iletişim için yaygın olarak kullanılan İletim Denetimi Protokolü'nün (TCP) önemli bir bileşenidir.

Adım 1- İlk Bağlantı İsteği SYN:

İstemci sunucuya SYN bayrağı ayarlanmış bir TCP segmenti gönderir. Bu segment, istemcinin iletişimi başlatabilmesi için bir sıra numarası (SYN sıra numarası) içerir. Ayrıca maksimum segment boyutu (MSS) gibi diğer TCP seçeneklerini de içerir.

Adım 2- Sunucu Yanıtı SYN-ACK:

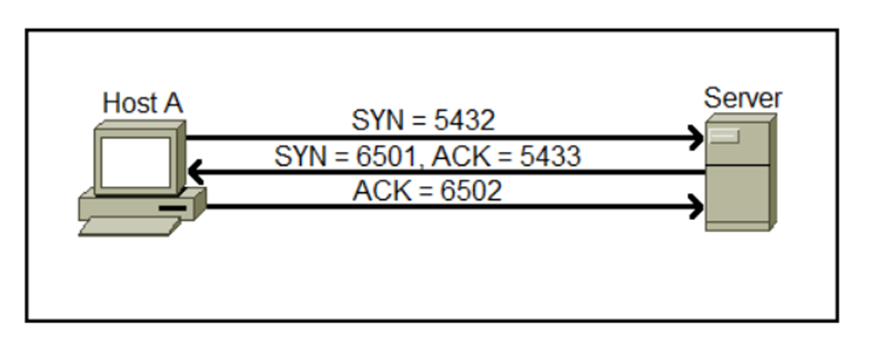
Sunucu SYN segmentini aldıktan sonra bağlantı için gerekli kaynakları ayırır ve kendi SYN sıra numarasını oluşturur. İstemciden gelen sıra numarasına bakar ve istemcinin gönderdiği sıra numarasını 1 arttırarak göndereceği ACK numarasını ayarlar.

Sunucu, istemciye hem SYN hem de ACK bayrakları ayarlanmış bir TCP segmenti göndererek yanıt verir.

Adım 3- İstemci Onayı ACK:

İstemci SYN-ACK segmentini aldıktan sonra sunucunun yanıtını kabul eder. İstemci ACK bayrağı ayarlanmış bir ACK segmenti göndererek sunucunun SYN sıra numarasını onaylar. Sunucudan gelen sıra numarasına bakar ve sunucunun gönderdiği sıra numarasını 1 arttırarak göndereceği ACK numarasını ayarlar.

Bu adımdan sonra istemci ve sunucu senkronize edilmiş sıra numaralarını kullanarak veri göndermeye başlayabilirler.



Şekil 3: TCP Üçlü El Sıkışma

TCP bağlantısı başlatıldıktan sonra veri iletmeye başlayabiliriz. Ancak HTTPS'li web siteleri için yapmamız gereken bir adım daha var, o da verilerin şifrelenerek güvenli bir şekilde iletilmesini sağlamak. Bu nedenle veri iletmeden önce TLS devreye girer.

**SSL/TLS**

SSL (Secure Sockets Layer) istemci ve sunucu arasında güvenli iletişim sağlamak için kullanılan protokoldür. Ancak zaman içinde SSL, güncellenerek TLS (Transport Layer Security) olarak adlandırılan yeni bir protokole dönüştü. TLS, aynı amaca hizmet eden ve SSL'nin yerini alan bir protokoldür.

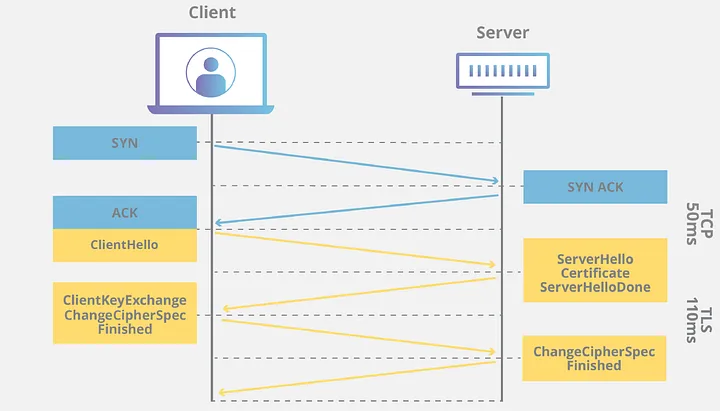
Her iki protokol de X.509 sertifikalarını kullanır ve kimlik doğrulaması için ortak anahtarlarla değişim yapar. SSL ve TLS, güvenli iletişimi sağlamak ve veri bütünlüğünü korumak için kullanılan şifreleme yöntemlerini belirler.

Bugün genellikle SSL/TLS terimi kullanılır çünkü TLS, SSL'nin geliştirilmiş ve güncellenmiş bir sürümüdür. SSL, özellikle eski sürümleri (SSL 2.0 ve SSL 3.0) güvenlik zafiyetleri nedeniyle yaygın olarak kullanılmamaktadır. TLS ise güvenlik açısından daha güncel ve güvenilirdir.

**Handshake (El sıkışma)**

SSL / TLS bağlantısı kurma işlemi oldukça karmaşıktır. Spesifik adımlar burada açıklanmıştır:

1. İletişim, istemcinin bir "merhaba" mesajı göndermesiyle başlar. Bu mesaj, istemcinin SSL sürüm numarasını, şifre ayarlarını (kullanılabilecek algoritmaları), oturuma özgü verileri ve diğer bilgileri içerir. İstemci, sunucuyla SSL üzerinden iletişim kurmak istediğini belirtir.
2. Sunucu istemciye bir "sunucu merhaba" iletisi ile yanıt verir. Bu iletişimde sunucunun SSL sürüm numarası, şifre ayarları (kullanılabilecek algoritmalar), oturuma özgü veriler ve istemcinin ihtiyaç duyduğu diğer bilgiler yer alır. Sunucu ayrıca X.509 sertifikasını istemciye gönderir. İstemci sunucunun kimliğini doğrulamak ve ortak anahtarını kullanmak için bu sertifikayı kullanır. İsteğe bağlı olarak istemci kimlik doğrulaması da gerekebilir.
3. İstemci sunucunun kimliğini doğrulamak için sunucunun X.509 sertifikasını kullanır. Bunun için sertifika yetkilisinden alınan ortak anahtarı kullanarak sertifikayı doğrular. Kimlik doğrulama başarılı olduğunda istemci sunucunun iddia ettiği kişi olduğuna güvenebilir ve iletişimi güvenli bir şekilde devam ettirebilir.
4. İstemci şimdiye kadar elde edilen tüm verileri kullanarak oturum için özel bir anahtar oluşturur. Bu anahtar sunucunun X.509 sertifikasından alınan ortak anahtarla şifrelenir ve sunucuya gönderilir. Bu adımda ortak bir anahtar belirlenir ve iletişim şifrelenir.
5. Sunucu istemci kimlik doğrulaması gerektiriyorsa, istemcinin sunucuya göndermesini gerektirir. İstemci X.509 sertifikasını sunucuya gönderir. Sunucu istemcinin kimliğini doğrulamak için bu sertifikayı kullanır.
6. İstemci kimlik doğrulaması gerekiyorsa ve istemcinin kimliği doğrulanamıyorsa oturum sona erer. İstemcinin kimliği başarıyla doğrulanırsa sunucu istemcinin gönderdiği pre-master gizli anahtarı parolasını çözmek için özel anahtarını kullanır.
7. İstemci ve sunucu oturum anahtarlarını oluşturmak için birbirlerine mesajlar gönderir. Oturum anahtarları simetrik anahtarlar olup el sıkışma adımında hem istemci hem de sunucunun anlaştığı algoritmayı kullanır.
8. İstemci pre-master gizli anahtardan simetrik anahtar oluşturmayı tamamladıktan sonra sunucuya bir mesaj gönderir. Bu mesaj gelecekteki mesajların oturum anahtarıyla şifreleneceğini belirtir. Daha sonra istemci el sıkışmanın tamamlandığını bildiren şifreli bir mesaj gönderir.
9. Sunucu pre-master gizli anahtardan simetrik anahtar oluşturmayı tamamladıktan sonra istemciye bir ileti gönderir. Bu ileti gelecekteki mesajların oturum anahtarıyla şifreleneceğini belirtir. Sunucudan gelen iletiler oturum anahtarıyla şifrelenir. Sunucu daha sonra el sıkışmanın tamamlandığını bildiren şifreli bir mesaj gönderir.



Şekil 4: TCP üç yollu anlaşma ve TLS anlaşması

TLS Handshake anlamayı kolaylaştırmak için şu kavramlara bakalım:

**Şifre paketi nedir?**

Güvenli bir iletişim bağlantısı kurmak için kullanılan bir dizi algoritmadır. TLS anlaşmasının önemli bir parçası istemci ve sunucunun hangi şifre paketini kullanacakları konusunda anlaşmaya varmalarıdır. Bu anlaşma genellikle desteklenen şifre takımlarının listesinden birinin seçilmesiyle gerçekleşir.

Şifre paketleri şifreleme, kimlik doğrulama, anahtar değişimi ve bütünlük kontrolü gibi işlemleri gerçekleştirmek için kullanılan algoritmaları içerir. Bu paketler güvenli bir iletişim sağlamak için protokol tarafından belirlenen kuralları takip eder. Her şifre paketi belirli bir güvenlik düzeyi ve performans özellikleri sunar.

**Generates master nedir?**

Anahtar paylaşımı ve anahtar türetme işlemi sırasında kullanılan bir terimdir. TLS (Transport Layer Security) protokolünde, istemci ve sunucu arasında güvenli bir bağlantı kurulması için bir oturum anahtarı üretilir.

Oturum anahtarı, iletişim sırasında kullanılan şifreleme ve kimlik doğrulama işlemlerinde kullanılır. Bu anahtar güvenli veri iletimini sağlamak için kullanılan şifreleme anahtarlarının üretilmesinde temel bir rol oynar. "Generates master" terimi bu oturum anahtarının oluşturulması ve kullanılması anlamına gelir.

**TLS Handshake**

**TLS 1.3 anlaşması:**

1. Sunucu, 443 numaralı bağlantı noktasındaki yeni bağlantıları dinler. (443 portu HTTPS- TLS/SSL üzerinden HTTP Protokolü)
2. İstemci, 443 numaralı bağlantı noktasına bağlanır ve sunucuya gönderilen bir ClientHello mesajıyla el sıkışma işlemini başlatır. TLS 1.3'te şifre paketlerinin listesi büyük ölçüde azaltılmıştır (37'den 5'e). İstemci, şifre paketlerinin her biri için bir anahtar çifti hesaplar ve bunu protokol sürümüyle birlikte sunucuya gönderir.
3. Sunucu anahtar oturumunu hesaplar. İstemcinin anahtar paylaşımı ile sunucu oturum anahtarını üretebilir.
4. Sunucu, "merhaba" anlamına gelen bir ServerHello mesajıyla yanıt verir. Sunucu kendi anahtar paylaşımını oluşturur ve bunu istemciye gönderir, böylece sunucunun şifreli SSL sertifikasıyla birlikte oturum anahtarını da oluşturabilir (3. Adımda oluşturulan oturum anahtarını kullanarak).
5. İstemci generates master oluşturur ve güvenli bir bağlantı kurulur. İstemci sunucunun anahtar paylaşımını alır ve oturum anahtarını hesaplar. Sunucu sertifikalarının şifresini çözer ve doğrular ve her şey başarılı gerçekleştiyse el sıkışma tamamlanır.

**TLS 1.2 ile ilgili sorunlar**

TLS 1.2 kullanıcılara TLS yapılandırma parametrelerinin birçoğunu bırakarak esneklik sağladı. Özellikle anahtar değişim yöntemi olarak Diffie-Hellman (DH) kullanmak mümkündü ancak DH parametrelerini ayarlamak kullanıcıların sorumluluğundaydı.

DH parametrelerinin doğru bir şekilde seçilmesi çok önemlidir çünkü bazı parametreler kolayca kırılabilir. Bu durum kullanıcıların doğru TLS 1.2 yapılandırmalarını sağlamak için teknik bilgi ve deneyime sahip olmalarını gerektiriyordu. Aksi takdirde zayıf parametreler veya hatalı yapılandırmalar kullanılabilir ve güvenlik riskleri ortaya çıkabilir.

Geriye dönük uyumluluk adına TLS 1.2 eski, güvensiz şifrelerin kullanılmasına da izin verdi. Zaman geçtikçe TLS'nin eski şifrelerinde veya sürümlerinde güvenlik açıkları keşfedildiğinden saldırılara daha yatkın hale geldi.

Tüm bunlara ek olarak RSA anahtar değişiminin birkaç zayıf yönü vardır:

* Aynı genel-özel anahtar çifti hem sunucunun kimliğini doğrulamak hem de pre-master sırrı şifrelemek için kullanılır. Bir saldırgan özel anahtarı ele geçirirse oturum anahtarlarını yeniden oluşturabilecek ve tüm oturumun şifresini çözebilecektir.
* Perfect Forward Secrecy (PFS)- Kusursuz iletme gizliliği yoktur. Sunucunun özel anahtarı değişmediğinden bir saldırgan, sunucunun özel anahtarı sızdırılırsa veya güvenliği ihlal edilirse yıllar sonra şifresini çözmeye çalışarak şifrelenmiş verileri yıllarca kaydedip depolayabilir.
* Oturum anahtarı kurulana kadar istemci ve sunucu arasında gidiş-dönüş yapmak gereklidir. Bu da verimsizliğe sebep olur.

**TLS 1.3'te neler değişti?**

* TLS 1.3, istemci ve sunucu arasında daha basit bir el sıkışma işlemi sağlar. Önceden kullanılan şifre paketlerinin sayısı büyük ölçüde azaltıldı ve aynı zamanda güvensiz seçeneklerin seçilmesi de engelledi.
* Yalnızca kısa ömürlü Diffie-Hellman anahtar değişim algoritmalarını destekler. Bu sayede her oturum için yeni bir anahtar üretildiği için Perfect Forward Secrecy (PFS) sağlanır. Bu sızan bir anahtarın yalnızca o belirli oturumun şifresini çözebileceği anlamına gelir. Gelecekte şifreleme anahtarlarına yönelik bir güvenlik ihlali meydana gelirse daha önceki oturumlar hala güvenli kalır. Bu güvenliği daha da artıran önemli bir özelliktir.
* 0-RTT (Sıfır Gidiş-Dönüş Süresi) devam ettirme desteği

TLS 1.3 "0-RTT" olarak adlandırılan bir modu destekler. Bu mod, istemci ile sunucu arasında iletişim için herhangi bir gidiş-dönüş veya ileri geri mesajlaşma gerektirmeyen hızlı bir TLS el sıkışması sağlar. Bu özellik, istemci ve sunucu arasındaki iletişimi hızlandırır. Örneğin; kullanıcının daha önceki oturumda kimlik

KAYNAKÇA

[1]William Easttom, Modern Cryptography Applied Mathematics for Encryption and Information Security, Springer International Publishing, October 29 2022

[2]<https://www.linkedin.com/pulse/understanding-tcp-3-way-handshake-computer-networking-haque/>

[3]<https://www.siberportal.org/yellow-team/constructing-network-environment/tcp-three-way-handshake/>

[4]<https://medium.com/@alysachan830/tcp-and-tls-handshake-what-happens-from-typing-in-a-url-to-displaying-a-website-part-2-243862438cd9>

[5]<https://auth0.com/blog/the-tls-handshake-explained/>

[6]<https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/#:~:text=A%20TLS%20handshake%20is%20the,and%20agree%20on%20session%20keys>.