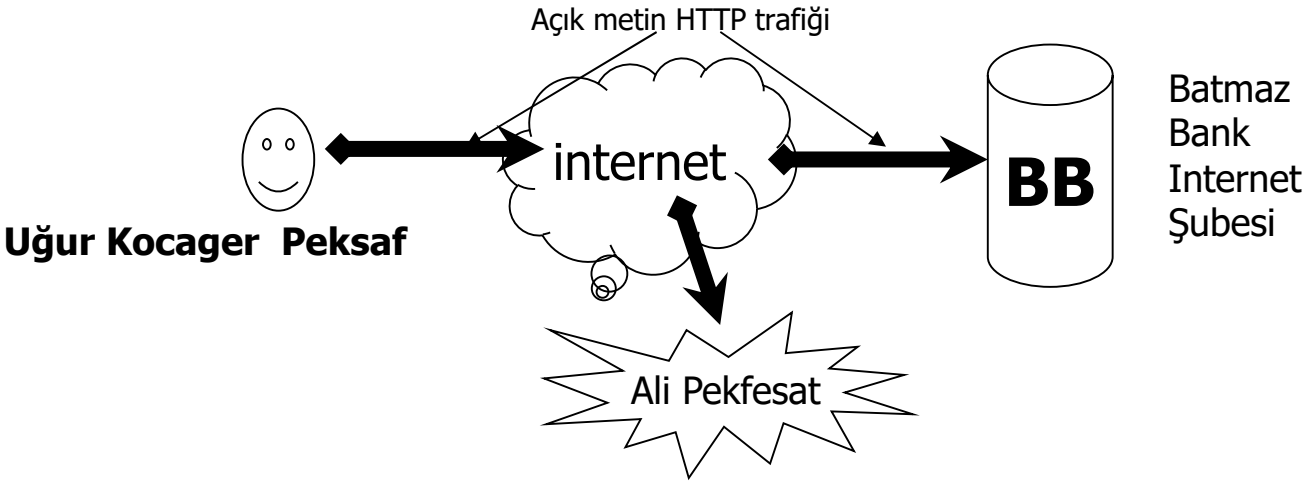
Transport katmanında güvenlik

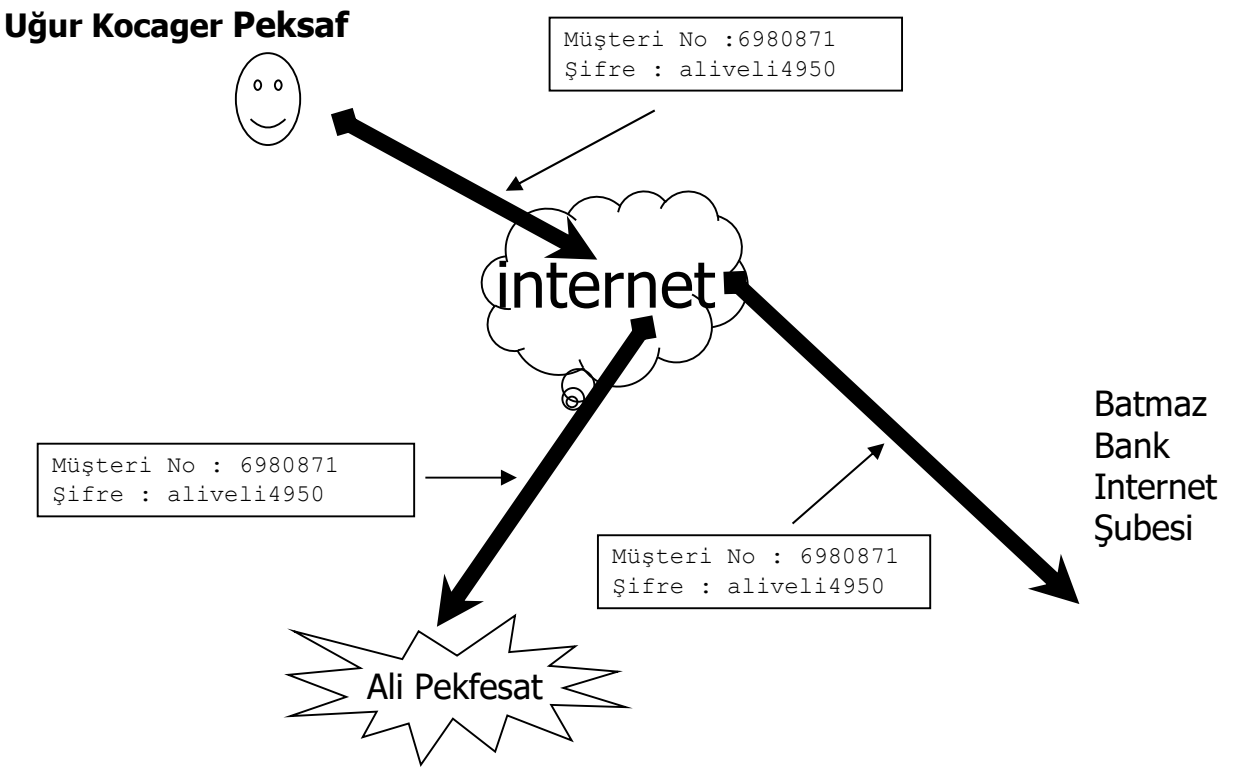
► **Bilgisayar ağlarının en fazla kullanım sahası İnternet'tir. WEB mekanizaması iki-yollu bir mekanizma olduğundan serverler herzaman için saldırılabilir hedeflerdir.**

► **Çoğu organizasyonlar Web'i mağazalarının vitrini olarak kullanırlar. Dolayısıyla web siteleri atak yediği takdirde önemli parasal kayıpları sözkonusu olabilir.**

► **Temeli oluşturan yazılım karmaşıktır, ve çok sayıda potansiyel güvenlik açıklarını gizlemek için kullanılabilirler.**

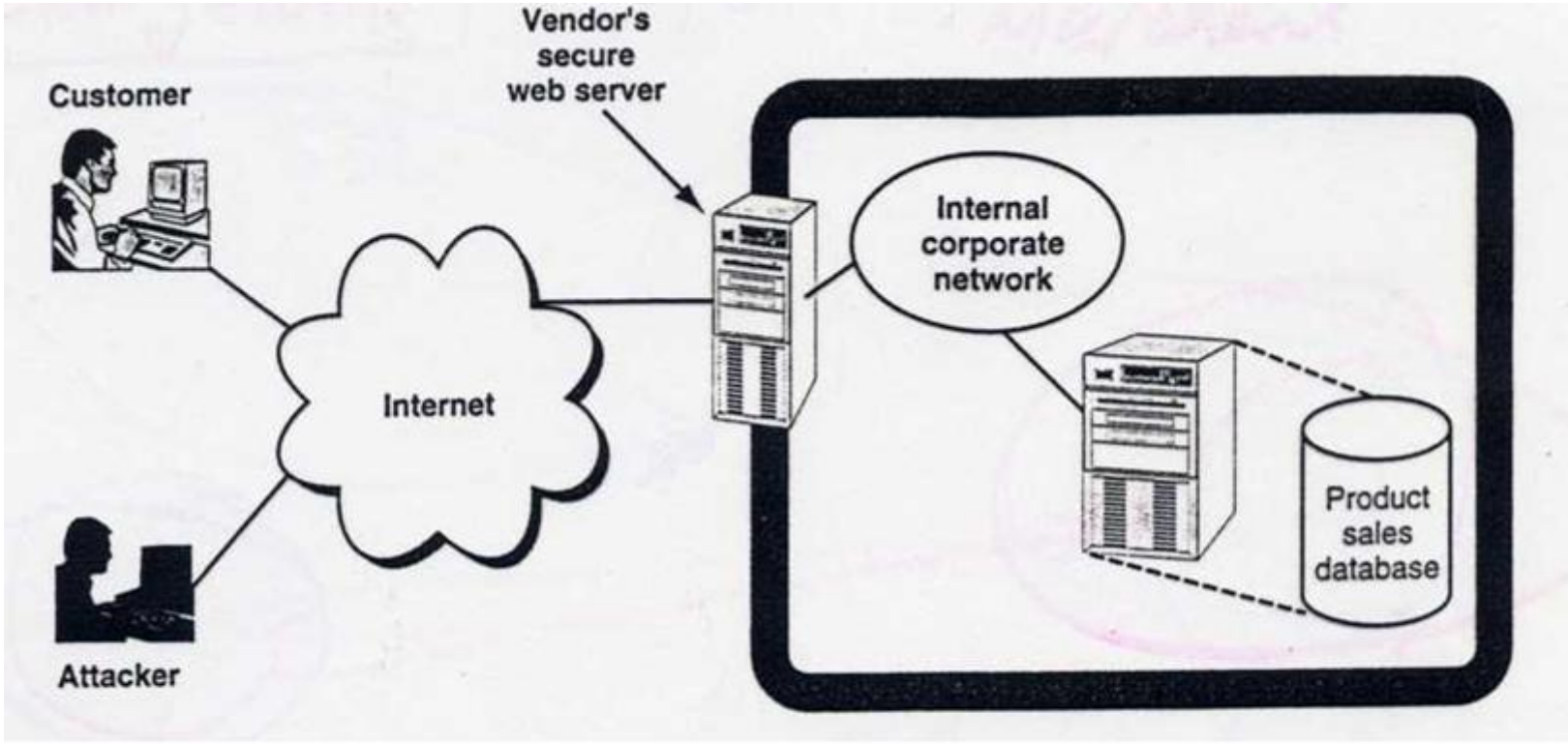
► **Bir web serverin yıkılması, onun bulunduğu intranet'e girmenin en önemli basamağıdır.**





► Internet'in önemi artıkça yapısından kaynaklanan. Güvenlik problemleri de önem kazanmakatdır. Korunması gereken kişisel bilgilerin Internet üzerinden aktarılacağı durumlarda, bankacılık işlemleri ve para transferleri v.b, yeterli güvenliğin sağlanması elzemdir.

WEB iletişimi için en önemli ve en alt seviye koruma (transport layer) *taşıma katmanı* koruma seviyesidir.



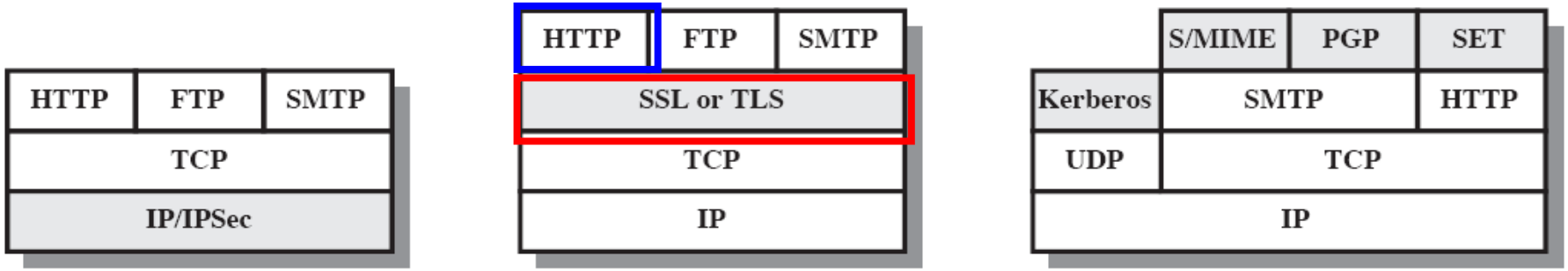
Bir e-ticaret sitesinin temel yapısı : satıcının Web server'i hem müşteriler hemde saldırganlar için erişilebilir bir yapıdadır. Kullanıcıların istekleri web serverde toplanır ve işletmenin veritabanına, işlenmek için gönderilir.

VERİ TABANI İşletme için hayati derecede önemlidir.!!!!!!!!!!!!!!!

"Üç-katmanlı sistem"

Web Güvenlik tehditleri – Bu tehditlerin IP ve TCP protokol açıklıklarından kaynaklandığını daha önceden bahsetmiştik. NELERDİ ??????

Web Trafiği güvenlik yaklaşımı: Varsayalım ki protokol açıklıklarını giderdiniz. Fakat veri güvenliği (***Veri bütünlüğün, gizliliğinin bozulması***, **Uçların *Kimlik doğrulaması*** v.b) için tedbir alınmamış ise durum ne olacaktır? FELAKET



(a) Network Level (b) Transport Level (c) Application Level

TCP/IP güvenlik süitinde, güvenlik birimlerinin lokasyonu.

Secure Socket Layer (SSL) ve Transport Layer Security (TLS)

Verinin güvenliksiz bir ortam olan internette bir yerden bir yere taşınması sorununa bir çözüm olması için SSL ve TLS protokolları ortaya atılmıştır.

SSL/TLS, internet üzerinde iki nokta arasında iletilen verinin bütünlüğü, gizliliği ve iki uç noktanın doğrulanması işlemlerini için bir çözüm yöntemidir.

Bu iki protokolda TCP katmanı ve uygulama katmanı arasında çalışır.

SSL 2.0 (Secure Socket Layer): 1994'de Netscape tarafından geliştirilmiştir. Daha sonra IETF (Internet Engineering Task Force) tarafından bir güvenlik standardı olarak kabul edilmiş ve Netscape'ten devralınmıştır.

TLS (Transport Layer Socket) : İnternete özel olmayan geliştirilmiş bir standarttır (RFC 2246, 1999).

TLS 1.0 : SSL 3.0'ın upgarde 'i olarak bilinir. Bazen TLS'ye SSL3.1 denebilir.

TLS 1.2, bazen SSL 3.3 versiyonu olarak ta isimlendirilir.

SSL (Secure Socket Layer)

SSL, güvenli olmayan bir iletişim ortamında verinin göndericiden alıcıya güvenli bir şekilde iletimini sağlamak amacı ile;

- *Sayısal imza* (digital signature) ve
- *Public key - Privite key* şifrelemesini , aynı anda kullanabilen bir yapıda tasarlanmıştır.

En yaygın kullanım şekli, web ortamında, Sunucu ile tarayıcı (Internet Explorer gibi..) arasındaki iletişimin doğrulanması ve şifrelenmesi şeklindedir. Veriyi sunan Web Server, Sertifika otoritesi tarafından imzalanmış özel ve herkese açık bir anahtara sahiptir. Bu anahtar, alıcı tarafından sunucudan güvenli bir şekilde *veri* alınmasını sağlar.

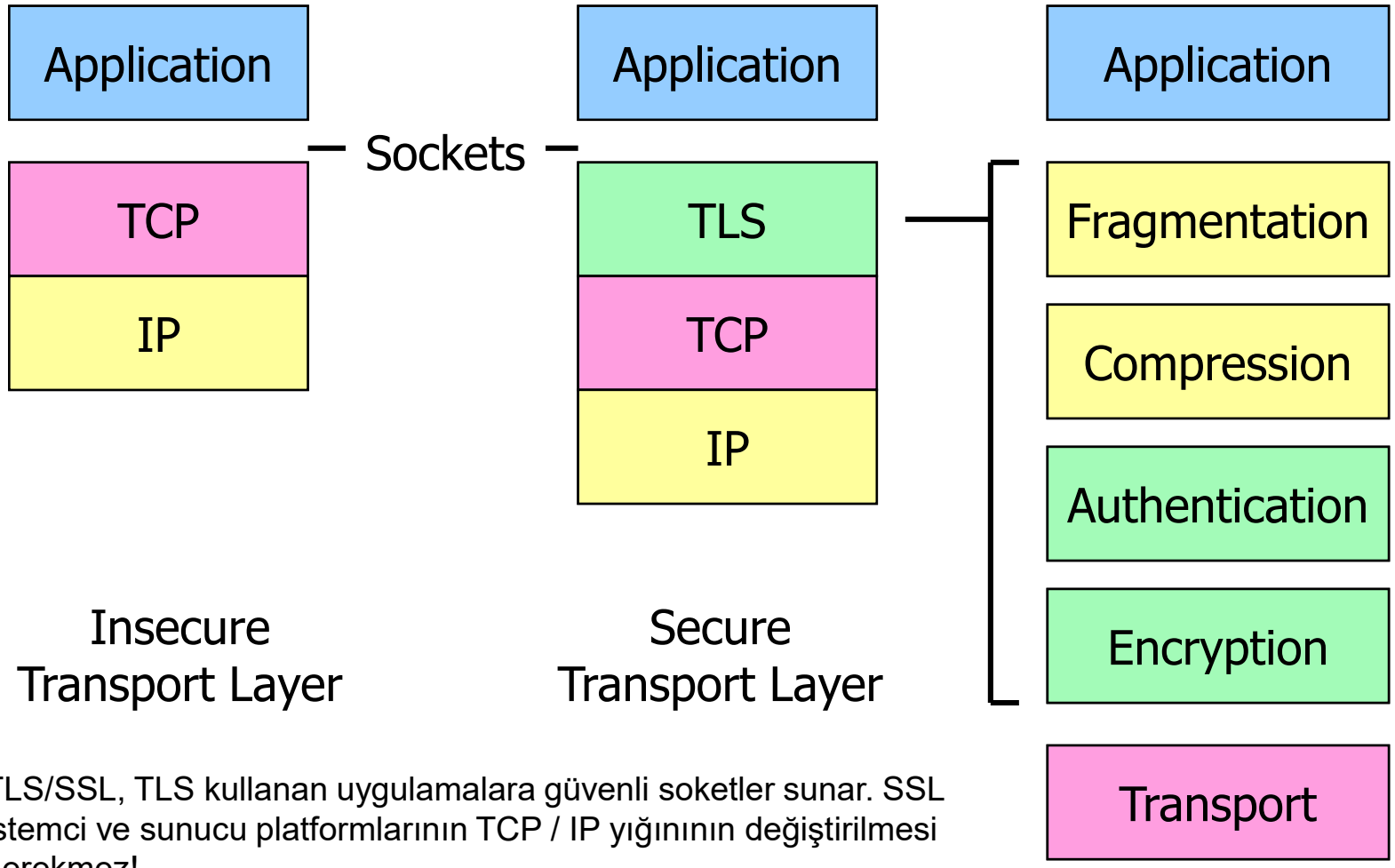
Önemli not: **Bu şifrelemeler; onay sağlayan birimler (Sertifika otoriteleri - CA) yardımıyla doğrulanır ve çalışır. Bu birimler, ulaşılan ilgili sunucunun, ait olduğu iddia edilen şahıs veya şirketlere ait olduğuna dair onay verir.**

SSL, e-posta göndermek, anlık ileti gönderip almak, Web sitelerine bağlanmak, İnternet üzerinde alışveriş yapmak gibi uygulamalarda güvenliği sağlamak için kullanılır.

SSL fonksiyonları;

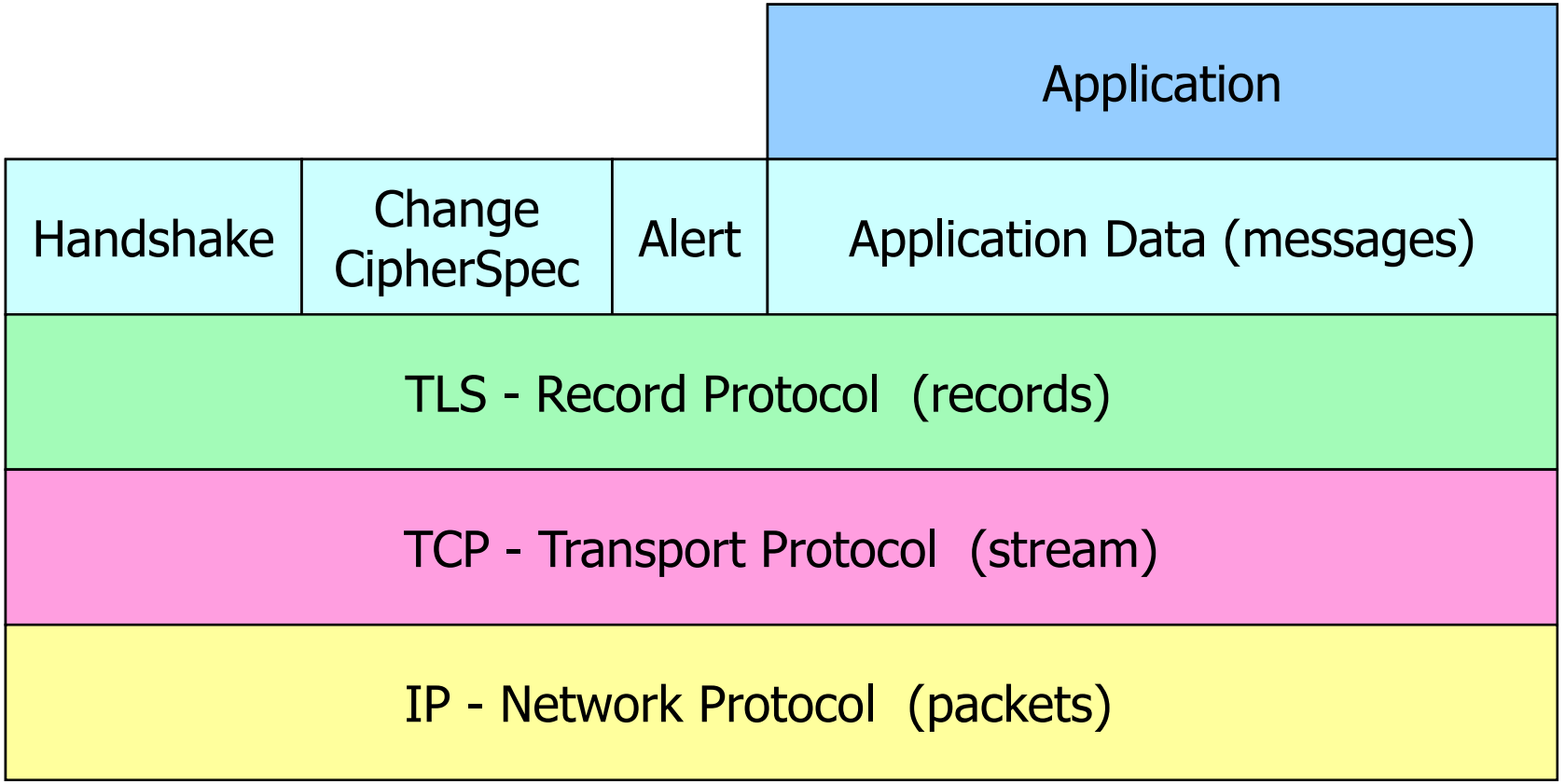
- Oturum için gerekli anahtarların oluşturulması suretiyle oturum sürecinde, mesajların şifrelenmesi ve deşifre edilmesindeki güvenlik ve gizliliği sağlar.
- Mesajı gönderenin ve mesajı alanın doğru yerler olduğunu garanti eder.
- İletilen dokümanların tarih ve zamanını doğrular.
- Doküman arşivi oluşturulmasını kolaylaştırır.
-Sunucuyla istemci arasında güvenli bir bağ oluşturur.

TLS/SSL Protocol Layers



TLS/SSL, TLS kullanan uygulamalara güvenli soketler sunar. SSL istemci ve sunucu platformlarının TCP / IP yığınının değiştirilmesi gerekmez!

TLS Record Protocol



- SSL teknolojisi, veri iletimi esnasında Hashing metodunu kullanır. Hashing metodolojisi ile sunucu, veriden tek bir hash değeri yollar.

- İstemci bu veri paketini aldığında, aynı hash fonksiyonunu kullanarak, gelen bu paketten bir hash değeri üretir.

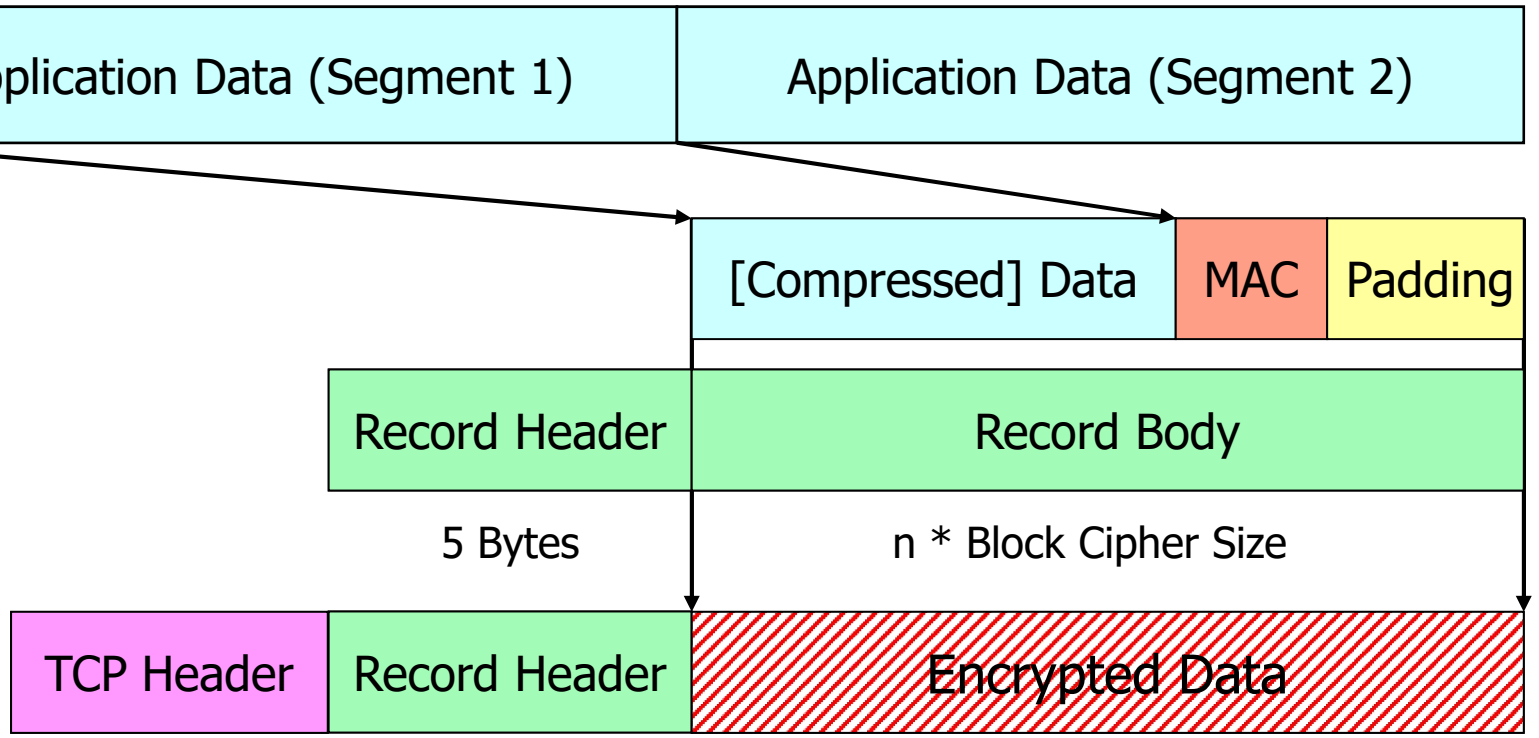
- İstemci ile sunucu, aralarında bağlantı sağlarken, aynı hash fonksiyonunu kullanmak için anlaşırlar.

- İstemcinin üretmiş olduğu bu hash değeri, sunucunun yolladığı hash değeri ile aynı olmak zorundadır. Eğer aynı değilse veri değişmiştir.

TLS Record Structure

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

```
⊞ Frame 9 (603 bytes on wire, 603 bytes captured)
⊞ Ethernet II, Src: Dell_92:44:9f (00:1a:a0:92:44:9f), Dst: SitecomE_5a:74:5e (00:0c:f6:5a:74:5e)
⊞ Internet Protocol, Src: 192.168.1.198 (192.168.1.198), Dst: hsr.ch (152.96.37.60)
⊞ Transmission Control Protocol, Src Port: 49825 (49825), Dst Port: https (443), Seq: 996, Ack: 1519, Len: 549
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Application Data Protocol: http
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 544
      Encrypted Application Data: C72EA7E35EE5501648FB77E216FCEA6CF62899BBD1ADDDAD...
```



HANDSHAKE
Sertifikalar

- Sayısal imzaları kullanarak bir verinin gerçekten beklenen kişi tarafından gönderildiği ve iletim esnasında değişikliğe uğramadığını anlayabiliriz.

- Peki, gönderilen verinin gerçekten göndermek istediğimiz kurum veya insana ulaştığından nasıl emin olabiliriz? Yani açık anahtarını kullanarak verileri şifrelediğimiz kişi gerçekte düşündüğümüz kişi midir? Nasıl emin olabiliriz.
- Burada sertifika tanımı ortaya çıkıyor. Sertifika, basitce kişinin açık anahtarının yetkili bir sertifika otoritesi tarafından imzalanmış halidir diyebiliriz. Yani karşı taraftaki kurum veya kişinin doğru kişi olduğunun doğruluğunu ispatlayan mekanizmadır.
- ***Sertifikasyon Kurumu Sayısal sertifikaların verilmesi ve yönetilmesini gerçekleştiren kurumdur. Sayısal sertifikalar bu kurumların gizli anahtarıyla imzalanır***.
- **Sertifika, kurumun public key'i ve bu public key'inin hash değerinin sertifika otoritesinin (CA) private key'i ile imzalanmasından oluşur.**

Sertifika = **(Private Key CA** (Hash (Public Key kurum))) + Public Key **= Sayisal Imza + Public Key**

Sertifikanın Edinilmesi

Sertifika otoritesi (CA), bahsi geçen sertifikayı firmaya verirken, öncesinde bir kimlik denetiminde bulunur.

Güvenilir bir firma olduğuna kanaat getirdiğinde, sunucu firmaya sertifika verir.

Sertifika Otoritesinin vermiş olduğu bu sertifika, sunucu firmanın güvenilir ve onaylanmış bir firma olduğunu belgeler.

Türkiyedeki Sertifika otoriteleri, TurkTrust v.b gibi firmalardır.

Bir SSL sertifikası hangi bilgileri içerir?

- Sertifika sahibi kurumun unvanı
- Sertifikanın seri numarası ve son kullanma tarihi
- Sertifika sahibinin açık anahtarı
- Sertifika veren kurumun ESHS (Elektronik Sertifika Hizmet Sağlayıcısı) elektronik imzası

SSL sertifikası, tek bir sunucu adı için verilen bir sertifikadır. Sertifikanın içinde adı yazılı olan sunucuya yüklenerek SSL güvenliği sağlar.

SAN SSL sertifikası, birden fazla sunucu veya alan adını (domain) içerebilen bir SSL sertifikasıdır. Ortalama olarak 10-15 farklı sunucu adını içerebilir. Buradaki ölçüt genellikle SAN alanındaki karakter uzunluğu sınırlamasıdır.

Wildcard SSL sertifikası, tek bir alan adı için alınabilen ve o alan adının tüm alt alan adlarını kapsayan bir sertifikadır. Bu tür sertifikalarda alan adının (domain) başında "*" karakteri tüm alt alan adlarını kapsayacak şekilde kullanılmaktadır (örneğin; "*.domain.com.tr").

SSL teknolojisi, 40 bit, 56 bit ve 128 bit, 256 bit veriyi destekler. Ama daha güvenli olması için tercih edilen 256 bitlik veri alışverişidir.

Bir SSL Sertifikasının görünümü:

Bir Netscape istemcisi ile, güvenli bölgenin detayları alınıyor.

LKD SSL sunucusunun sertifikası....

This Certificate belongs to:

```
www.linux.org.tr
webmaster@linux.org.tr
Bilgi Guvenligi Grubu
Linux Kullanicilari Dernegi
Ankara, TR, TR
```

This Certificate was issued by:

```
LKD Root Certificate Authority
bgg@linux.org.tr
Bilgi Guvenligi Grubu
Linux Kullanicilari Dernegi (LKD)
Ankara, TR
```

```
Serial Number: 01
This Certificate is valid from Tue Sep 12, 2000 to Wed Sep 12, 2001
Certificate Fingerprint:
  BF:F1:06:75:DE:56:F4:77:82:C1:FD:34:A5:40:2B:62
```

El Sıkışma Aşamaları

1. İstemci, sunucuya ilk ulaştığında el sıkışma başlar ve her iki taraf, güvenlik amacı ile kullanılacak olan şifreleme fonksiyonu üzerinde anlaşır.
2. İstemci, sunucunun kimliğini denetler.
3. İstemci, bir ortak anahtar oluşturur, sunucunun publicv anahtarıyla şifreler ve bunu sunucuya yollar.
4. Sunucu bu anahtarı alıp, kontrol eder.
5. Eğer istenirse, sunucu da (istemciden) tarayıcıdan bir kimlik denetimi isteyebilir.

- El sıkışma başladığında, İstemci sunucudan kimlik bilgilerini ister. Bunun üzerine sunucu, İstemciye sertifikasını yollar.
- İstemci aldığı bu bilgiler eşliğinde simetrik bir anahtar oluşturur. Daha sonra sunucunun herkese açık anahtarı ile, bu bulduğu anahtarı şifreler. Ve bu değeri sunucuya yollar.
- Kendi özel anahtarını kullanarak, sunucu, İstemcinin yolladığı bu şifrelenmiş anahtarı çözer. (Public key'in özelliği , Public key ile şifreleme yapılır fakat deşifreleme onun private (gizli) anahtarı ile elde edilir mi?)

SSL'de Kullanılan Şifreleme Sistemleri (Ciper suiti)

a. **Hash Tekniği:**Veri trafiği esnasında, verinin değişmediğini ve bütünlüğünün korunduğunu anlamak için kullanılan Hashing tekniğinde anlaşma.

b. **Anahtar Değişim Tekniği(RSA, Diffie-Helmann) ile Şifreleme:** İstemci ve sunucunun, el sıkışma sonrasında , veriyi şifrelemek amacı ile kullandıkları simetrik anahtarı birbirine nasıl ulaştıracağını belirlemek için.

c. **Simetrik Veri Şifreleme:** Veri şifrelenmesinde kullanılacak olan RC2, RC4 gibi bir veri şifreleme tekniğidir.

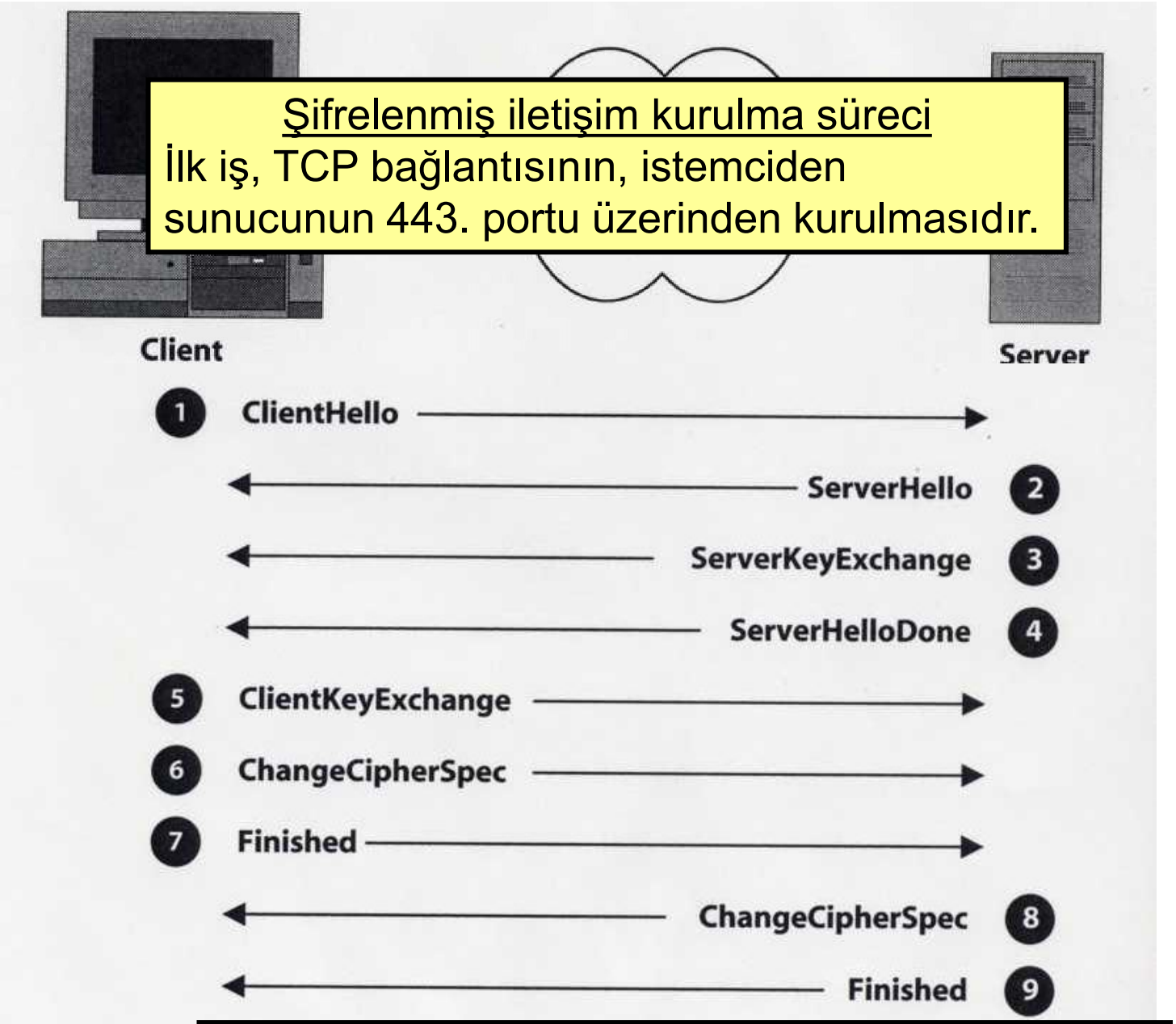
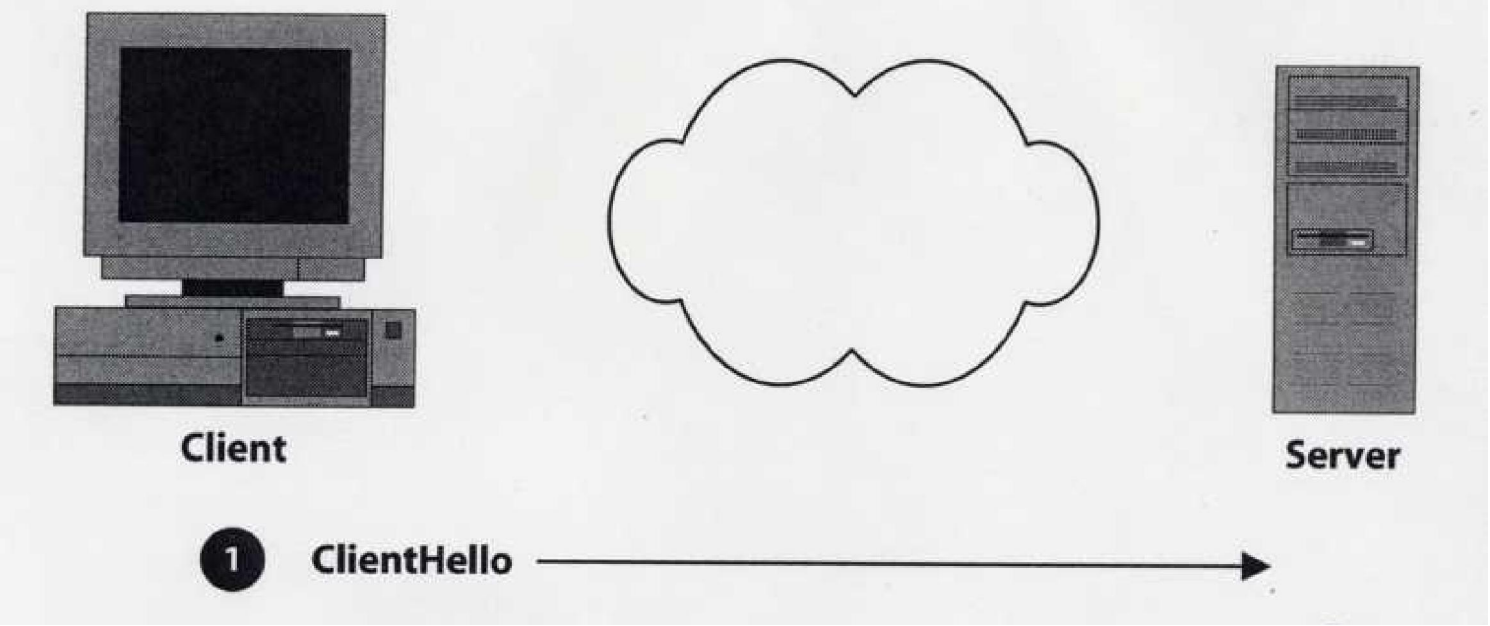


Figure 3-1

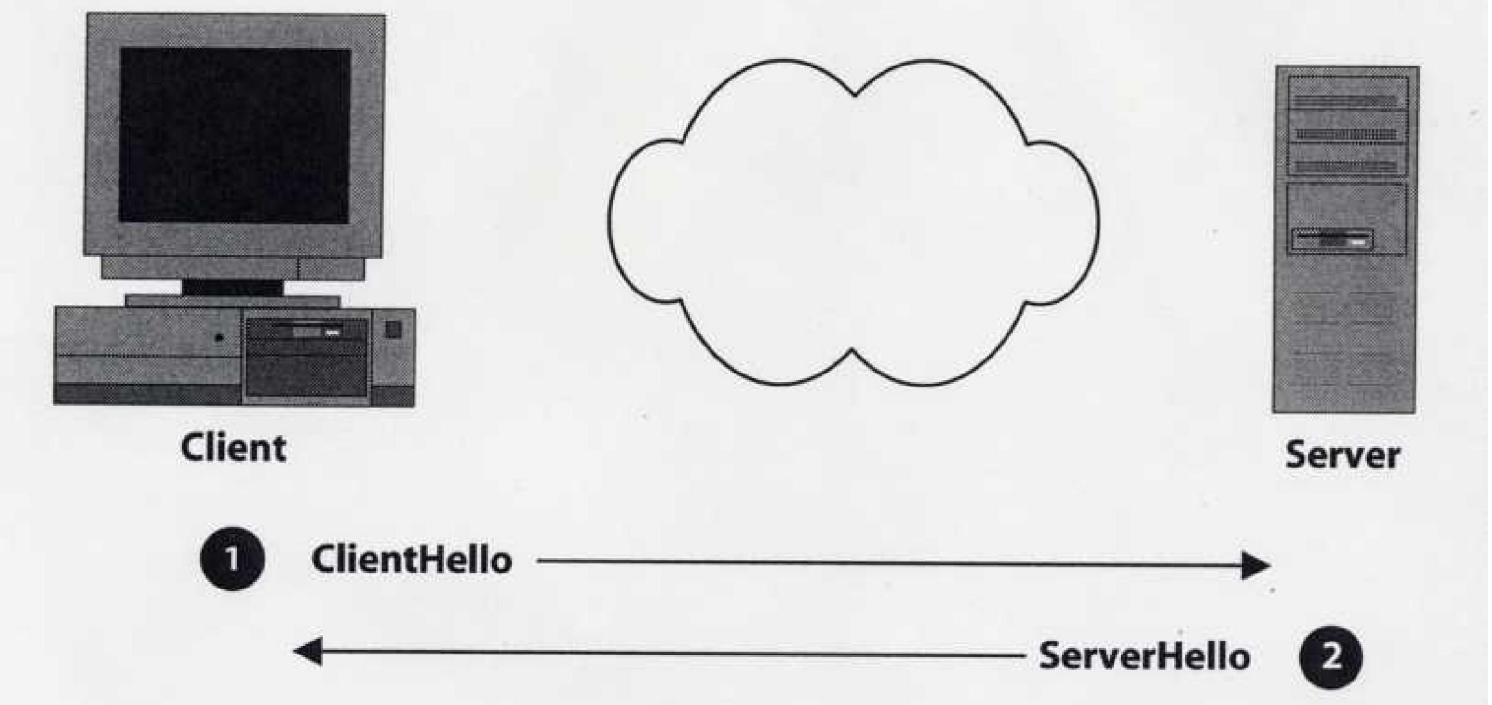
Güvenli kanal kuruldu – kullanmaya devam



Step	Action
1	Client sends ClientHello message proposing SSL options.

Client (İstemci- Web service requestor – Web browswer**), Server'a** (Servis sağlayıcıya- Web service provider***)*** ***ClientHello*** mesaj'ı ile bağlantı kurma isteğinde bulunur.

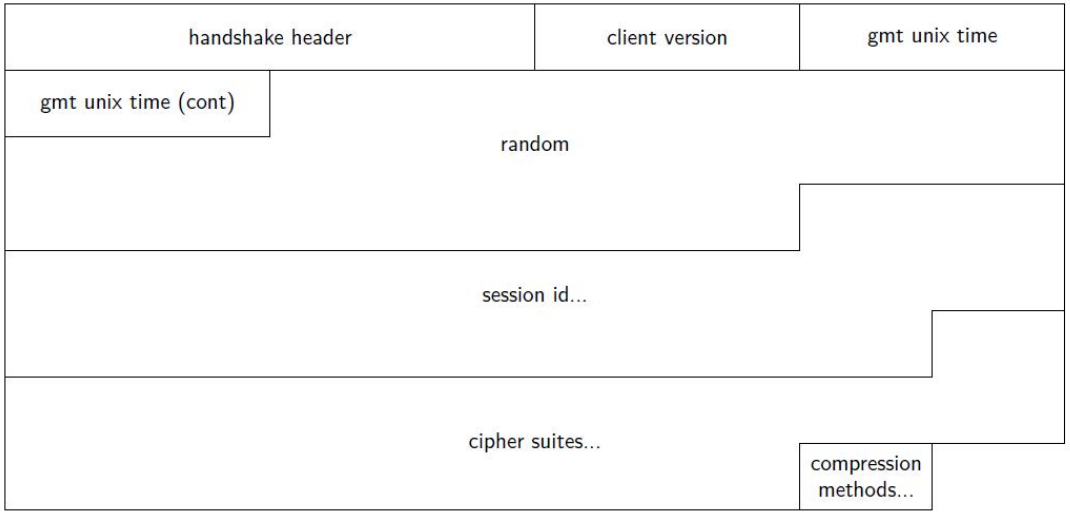
Bu mesajda, client'in kullandığı SSL sürümü no'su, desteklenen şifreleme algoritmaları (cipher paketleri) de dahil olmak üzere, hizmet isteyicinin desteklediği özellikler ve SSL oturum ID'si ve rastgele bir sayı (Şifrelemede kullanılan) vardır.



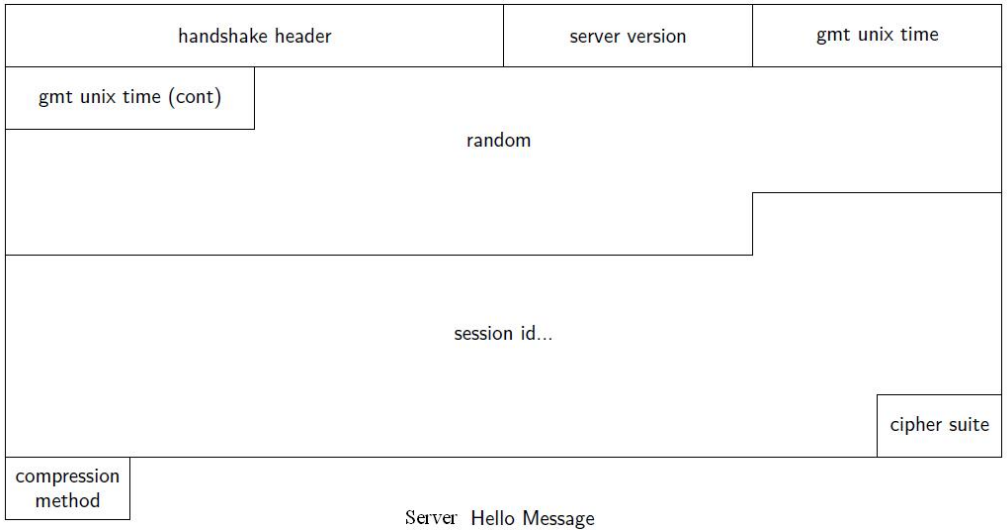
2	Server responds with ServerHello message selecting the SSL options.

Servis sağlayıcı buna bir ***ServerHello*** mesaj ile yanıt verir. Servis sağlayıcı, SSL için seçtiği şifre paketini (Şifreleme algoritma tipini) ve bu bağlantıyı tanımlayan bir oturum kimliğini bu mesaj ile client'a bildirir.

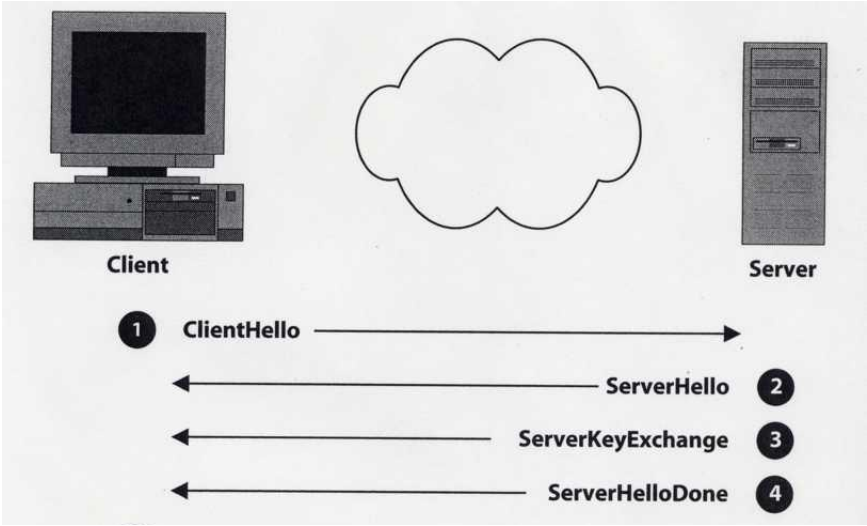
CLİENT ve SERVER'in HELLO Mesaj yapısı



Client Hello Message



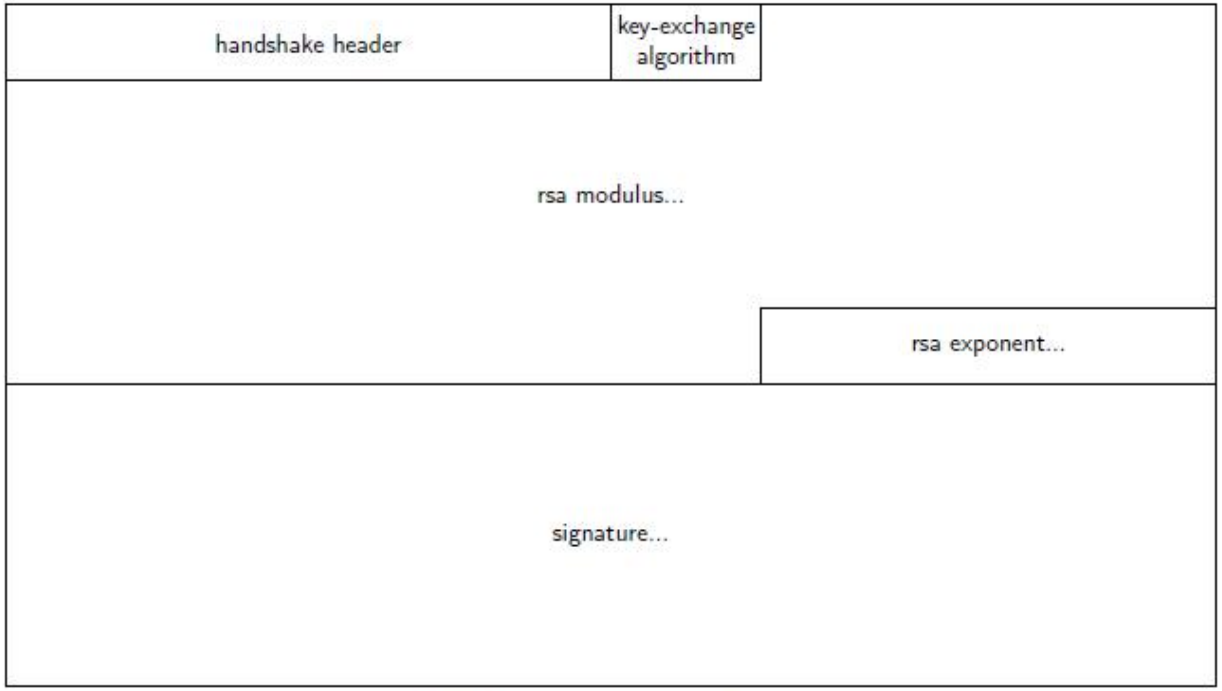
Server Hello Message



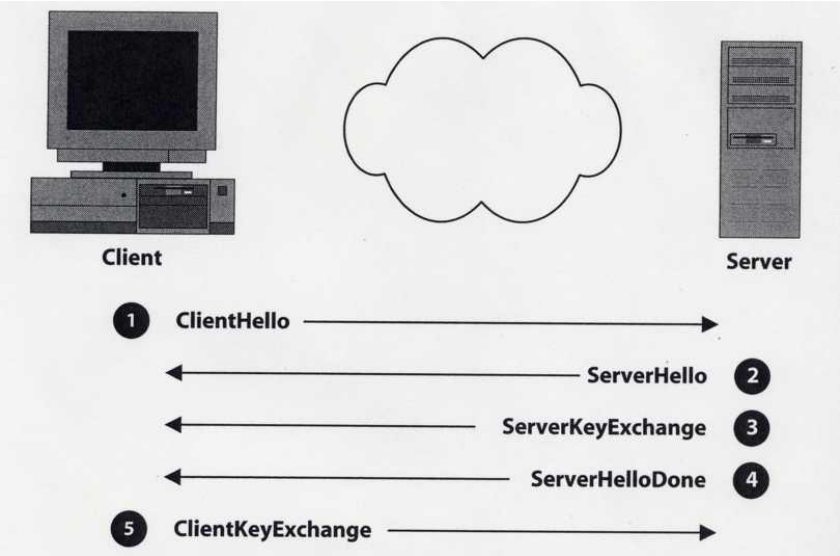
3 Server sends its public key information in ServerKeyExchange message.
4 Server concludes its part of the negotiation with ServerHello-Done message.

3- Sunucu istemciye kendi sertifikasını gönderir. Bu sertifika, bir sertifika yetkilisi (Certificated authuarity CA) tarafından imzalanmış X.509 sertifikası olmalıdır. (Öncelikle Web sunucu, bir Sertifika Otoritesinin kendisine tahsis etmiş olduğu sertifikayı yüklemiş olması gerekmektedir. **Böylelikle, Sertifika Otoritesi, sunucuyu onayladığını, istemcilere garanti etmiş olur**). Bu sertifika, sunucunun (Web provider) public key'ini (ortak anahtarını-açık anahtar) içerir.
Soru: CA'nın e-imzası ne işe yarayacak?..
4- Sunucu serverHello-Done mesajı ile oturum oluşturma için kendi üzerine düşeni yapar.

Server Certificate and Client Certificate Message

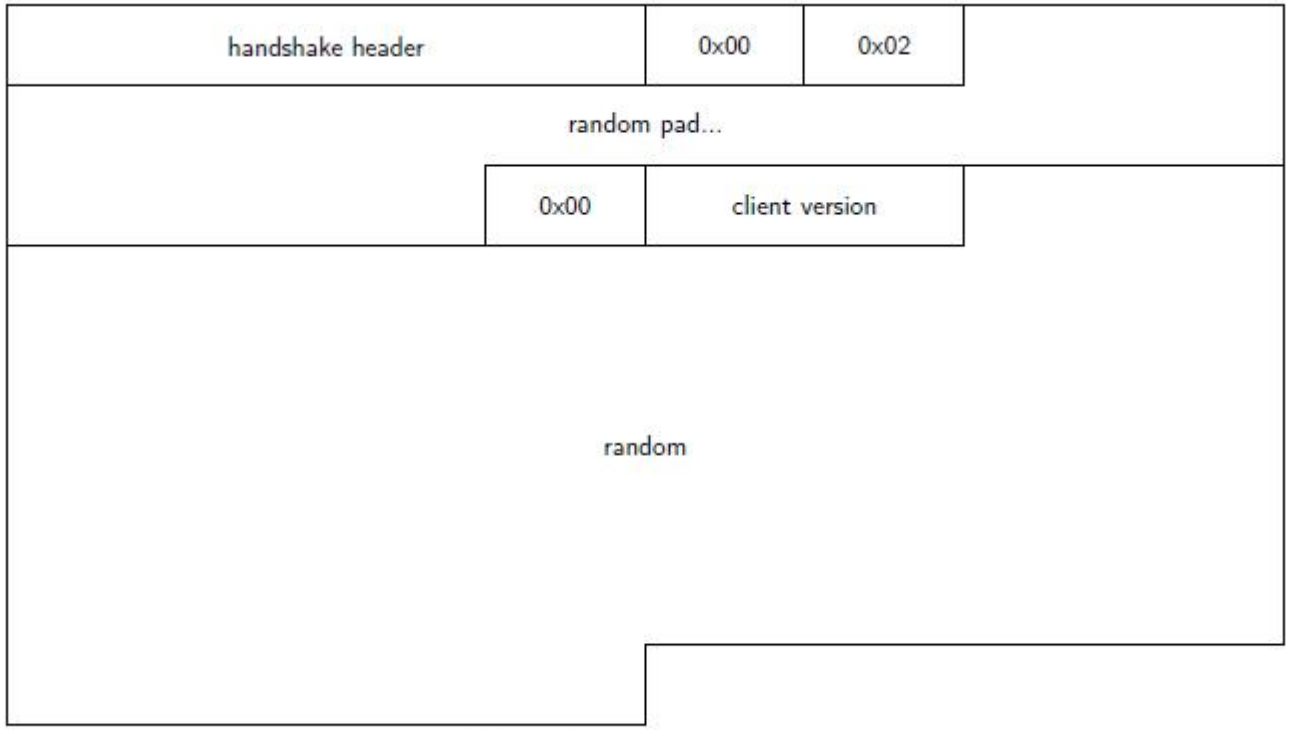


RSA Server Key Exchange Message

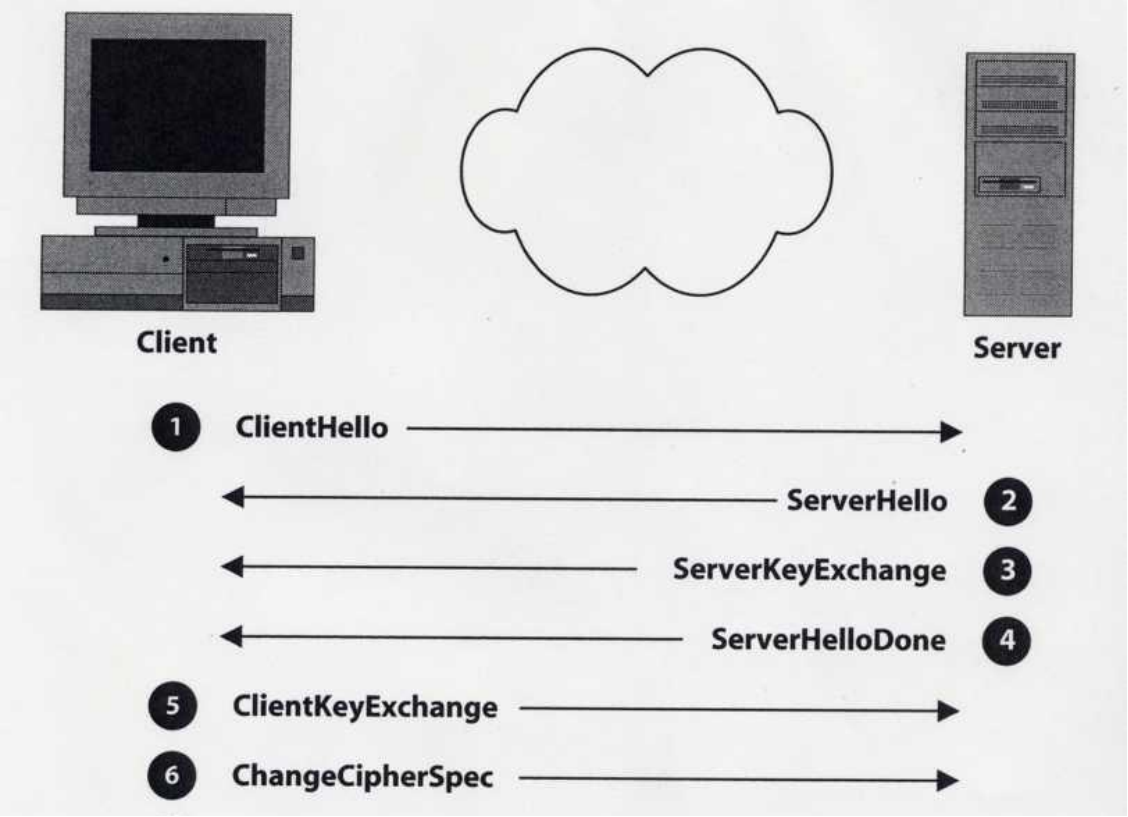


5	Client sends session key information (encrypted with server's public key) in ClientKeyExchange message.

5- İstemci, server'a ***"Client Key Exchange"*** mesajı gönderir. Bu mesajda, istemci tarafından, sertifika değerleri kullanılarak oluşturulmuş paylaşılan bir oturum anahtarı (shared key-simetrik bir şifreleme anahtarı) vardır. İstemci, servis sağlayıcıdan edindiği, bağlanacağı servis sağlayıcının ortak anahtarını (public key) kullanarak, oluşturduğu bu oturum (paylaşılan -Shared key) anahtarını şifreler ve servis sağlayıcıya geri gönderir. Bu şifreleme anahtarı, sunucu ve İstemci arasındaki verinin şifrelenmesi için kullanılacak olan ortak anahtar olmuştur.

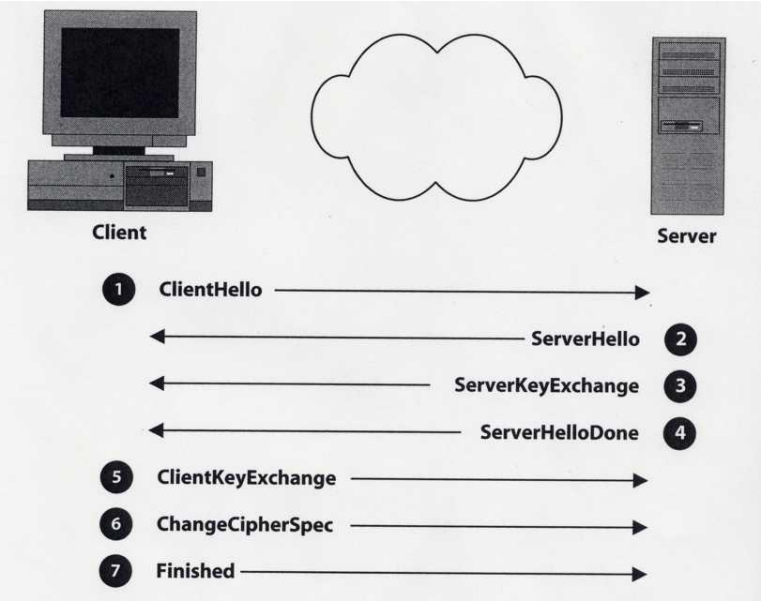


RSA Client Key Exchange Message



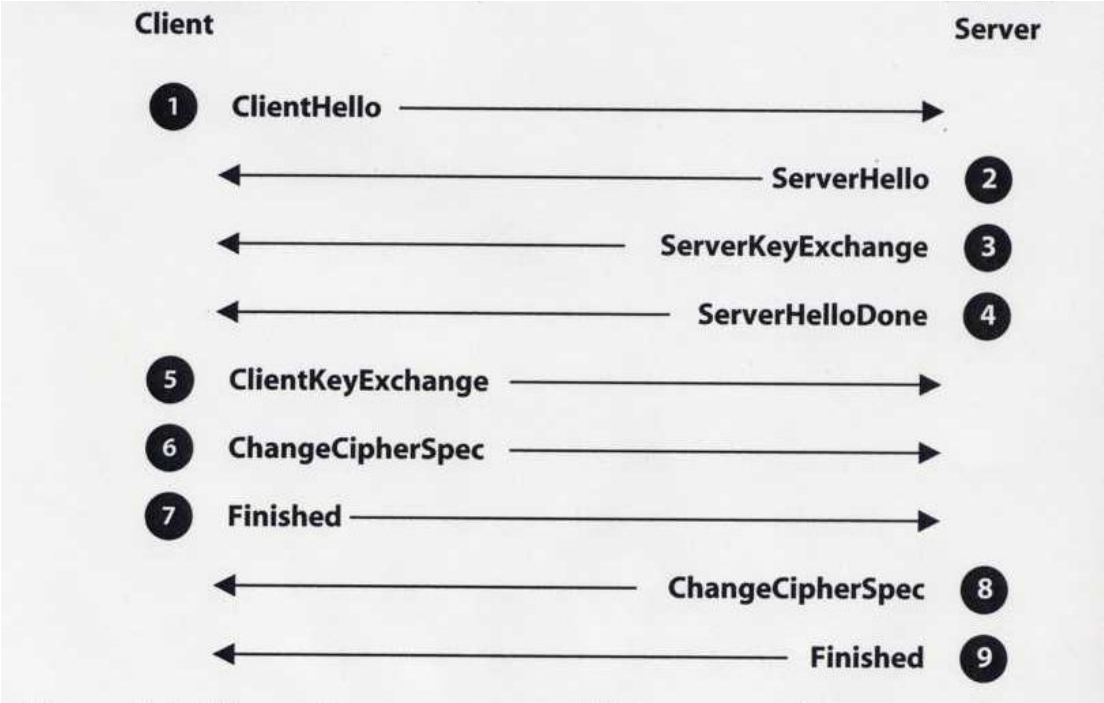
6	Client sends ChangeCipherSpec message to activate the negotiated options for all future messages it will send.

6- İstemci, gelecekteki tüm mesajları göndermek için aktif oturum seçenekleri için ***Changecipherspec*** mesajı gönderir.



7	Client sends Finished message to let the server check the newly activated options.

7-İstemci , kendi adına, el sıkışma (handshake) oturumun sonunu gösteren şifreli bir mesajı (MD5 veya SHA hasing v.b) sunucuya gönderir. Bu mesaj ***Finished*** mesajıdır. Mesaj aynı zamanda yeni aktiif edilmiş seçeneklerin kontrol edilmesini de sağlar. İstemci için El sıkışma aşaması biter.

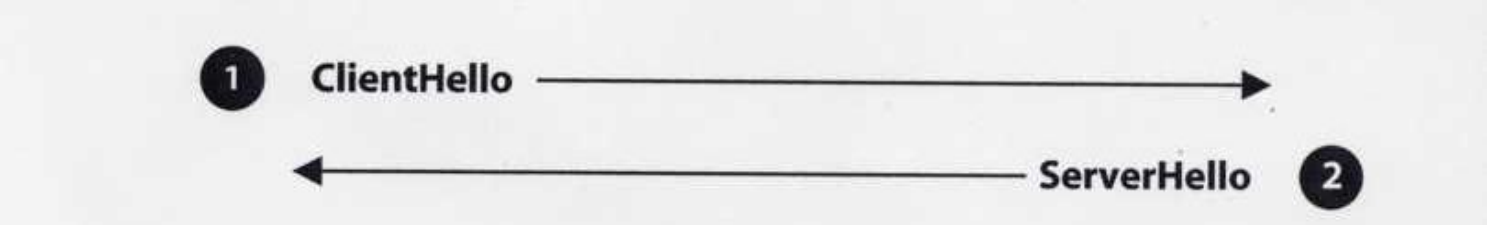


8	Server sends ChangeCipherSpec message to activate the negotiated options for all future messages it will send.
9	Server sends Finished message to let the client check the newly activated options.

6. Ve 7. adımlar server'dada aynı şekilde tekrarlanır (8.9. adım). Gerçek SSL oturumu başlar.

Istemci ve sunucu oturum anahtarını karşılıklı olarak kullanarak değiş-tokuş verilerinin bütünlüğünü doğrulamak, şifrelemek ve şifresini çözmek için kullanır.

Güvenli Kanal oluşturulmuştur.



1 ClientHello

Table 3-3 ClientHello Components

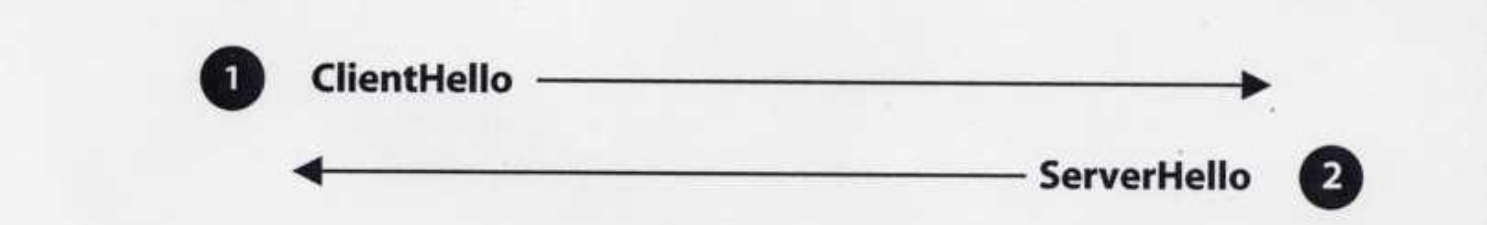
Field	Use
Version	Identifies the highest version of the SSL protocol that the client can support.
RandomNumber	A 32-byte random number used to seed the cryptographic calculations.
SessionID	Identifies a specific SSL session.
CipherSuites	A list of cryptographic parameters that the client can support.
CompressionMethods	Identifies data compression methods that the client can support.

Güncel versiyonlar:
SSL 3.3, TLS 1.2

1-Şifreleme işlemlerinin çekirdeği

2- reply ataklarını önleme

İlk 4 byt'ı oturum gününün tarihi



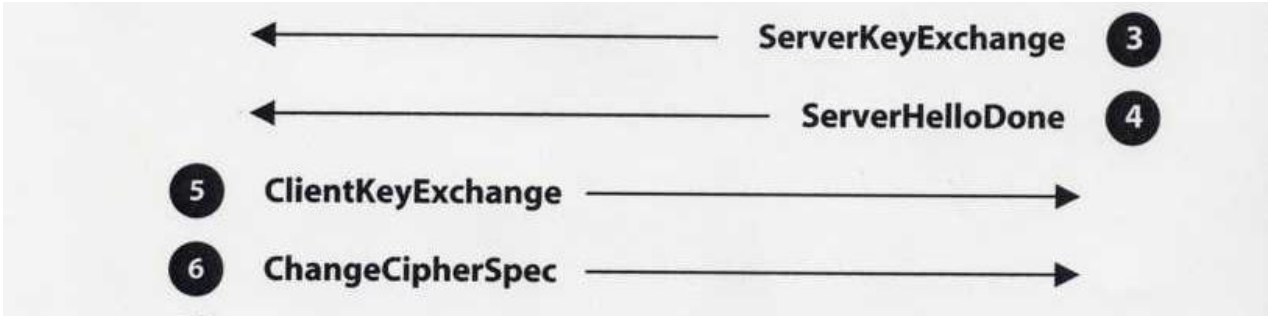
2 ServerHello

Table 3-4 ServerHello Components

Field	Use
Version	Identifies the version of the SSL protocol to be used for this communication.
RandomNumber	A 32-byte random number used to seed the cryptographic calculations.
SessionID	Identifies the specific SSL session.
CipherSuite	The cryptographic parameters to be used for this communication.
Compression-Method	The data compression method to be used for this communication.

Sunucunun karar verdiği

Sunucu, istemci tarafından sunulan menü seçer.



3. ServerKeyExchange

Server kendi public key'ini gönderir. Bu key certifika sağlayıcısından elde edilen certifakanın içindedir.

4.ServerHelloDone

Server tarafinda ilk uzlaşmanın tamamladığı bilgisidir.

5.ClientKeyExchange

İstemci, ürettiği özel anahtarı server'ın public key'i ile şifreleyerek sunucuya gönderir. Bu durumda sunucu, istemcinin bir özel anahtarı olduğunu teyit edecek ve artık biribirleriyle şifreli görüşmelerde bu anahtarlar kullanılacaktır.

6.Change Cipher Spec

Ön uzlaşmaların tamamlandığını istemci sunucu bildirir ve *"Ben kararlaştırılan bir şifre paketini kullanmaya başlamak istiyorum."* bilgisini gönderir.

6 ChangeCipherSpec

"Güvenli iletişim için bu süreç çok kritiktir. Her iki tarafında biribirini doğrulması çok önemlidir. SSL protokolunun en önemli süreçlerinden birisidir.

"SSL özellikleri bazı bilgilerin (özellikle, temel anahtar bilgilerinin) iletişimin her yönü için farklı olacağını tanımlar.

Diğer bir deyişle, bir dizi anahtarların, istemcinin sunucuya gönderdiği verileri garanti altına alacak ve farklı bir anahtar seti ise sunucunun istemciye gönderdiği verileri güvenlik altına alacağı olacaktır. “

“SSL , belirli bir sistemin , istemci veya sunucu olup olmadığını,, ***write state*** ve ***read state*** ile tanımlar. ***Write state*** sistemin gönderdiği veriler için güvenlik bilgisini tanımlar. ***read state*** sistemin aldığı verile için güvenlik bilgisini tanımlar.

Pending (pnd) : karara bağlanmamış olan, pek yakında "message authentication codes" (MAC).

6

ChangeCipherSpec

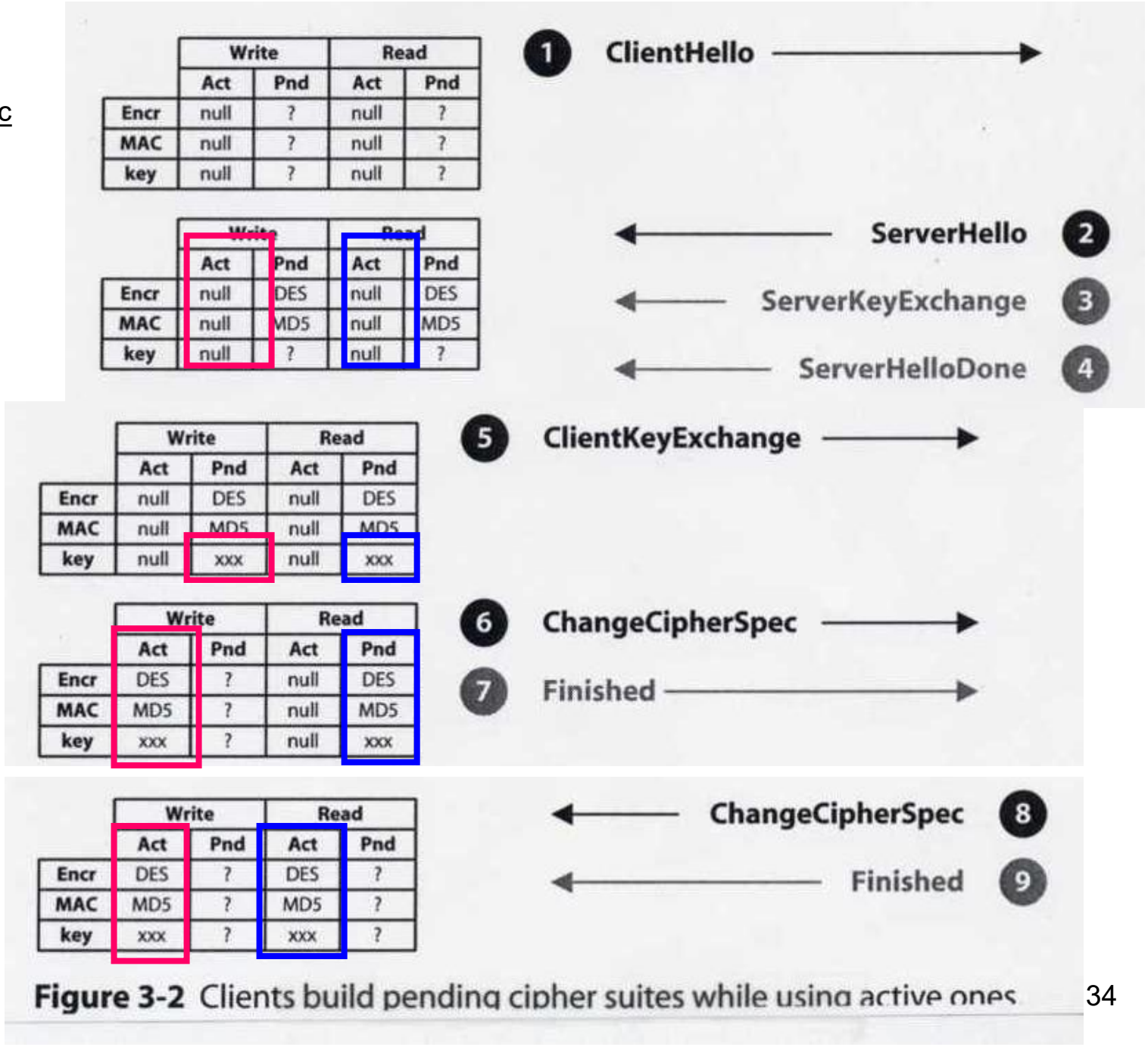


Figure 3-2 Clients build pending cipher suites while using active ones

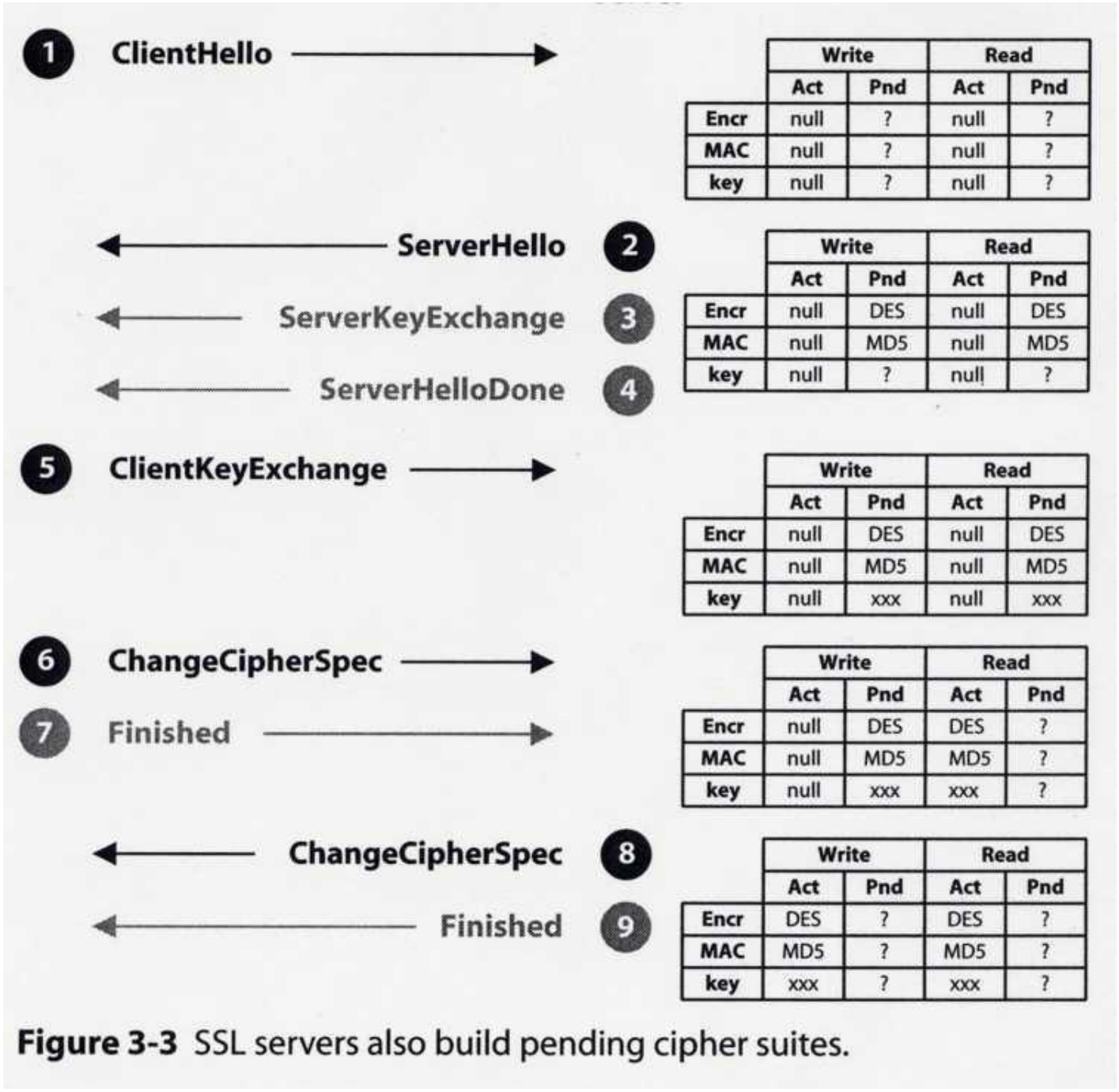


Figure 3-3 SSL servers also build pending cipher suites.

7 Finished

"***ChangeCipherSpec mesajları”*** gönderdikten hemen sonra, her bir sistem bir ***Finished*** mesaj gönderir. ***Finished*** mesajları her iki sistemin uzlaşmada başarılı olduğunu ve bu güvenlik tehlikesi olmadığını doğrulamak içindir. ***Finished*** mesajı iki yönü bu güvenliğe katkıda bulunur.

“Finished mesajının içeriği ile SSL uyuşma sürecindeki güvenliği korumak için hizmet vermektedir. Her Finished mesajı, sadece bitmiş uzlaşma ile ilgili önemli bilgiler içerir (act. Bilgileri) . Bu iletişim, gerçek mesajlara hayali mesajları eklemek ya da çıkarmak için kullanılan saldırganlara karşı da önemli bir korumadır.

SSL ve TLS kısaca güvenli bir oturum açarak iki nokta arasında güvenli bir veri transferi yapar

- **Bir oturum kurulur**

- Kullanılacak algoritmalarda mutabık kalınan.
- Gizliliğin paylaşıldığı.
- Kimlik doğrulamasının yapıldığı.

- **Veri transferi yapılır**

-Haberleşme gizliliği olan.

•Simetrik şifreleme uygulanarak

- Veri bütünlüğü olan.

Anahtarlanmış mesaj doğrulama kodu (HMAC) ile.

SSL Mimarisi

SSL, TCP üzerinde ***uçtan uca (end-to end) güvenli bir bağlantı*** hizmeti sunulması için tasarlanmıştır ve katmanlı bir yapıya sahiptir. SSL, TCP/443. port üzerinde çalışır.

Bir SSL bağlantısı yapılacağı zaman genellikle tarayıcı program bir uyarıda bulunur, bağlantı gerçekleştirildiğinde, örneğin IE'de bu *(Sürüm 5.x) sağ köşede kapalı bir asma kilit sembolü ile* belirtilir. Asma kilit sembolü üzerine kliklenerek detaylı bilgi alınabilir.

SSL bağlantısı kurulurken, kullanıcı ve sunucu arasında bir dizi uzlaşı işlemi gerçekleştirilir; uzlaşı başarısız olursa SSL oturumu kurulmaz, hiçbir bağlantı yapılmaz ve veri iletimi gerçekleşmez.

1- Bir oturumun kurulması
   -Algoritmalar üzerinde anlaşmak
   - Paylaşılmış güvenlik
   - Kimlik doğrulamasını başarmak
2- Uygulama verisini transfer etmek
   - Gizlilik ve bütünlüğü sağlayarak

Bağlantı ve oturum kavramları

Bağlantı (Connection) : Sunucu/Kullanıcı arasında uygun türde bir hizmeti sağlayan mantıksal bir bağlantıdır *(Bir web sayfasının indirilmesi gibi). SSL için bunlar uçtan* uca ilişkilerdir. Bağlantılar geçicidir. Her bağlantı bir oturuma ilişkilendirilmiştir.

- Transport hizmetlerini kullanır.
- TLS kriptolama ve bütünlüğü TLS sağlar.
- TLS Record Protocol'unu kullanır.

Oturum (Session): Sunucu ve kullanıcı arasında bir ilişkilendirmedir.
Oturumlar Uzlaşı Protokolü ile kurulurlar. Oturumlar, birden fazla bağlantı arasında kullanılabilen kriptografik güvenlik parametrelerini tanımlarlar. Oturumlar her bir yeni bağlantı için yeni güvenlik parametre uzlaşı işlemlerinin tekrarlanmasını engellerler.

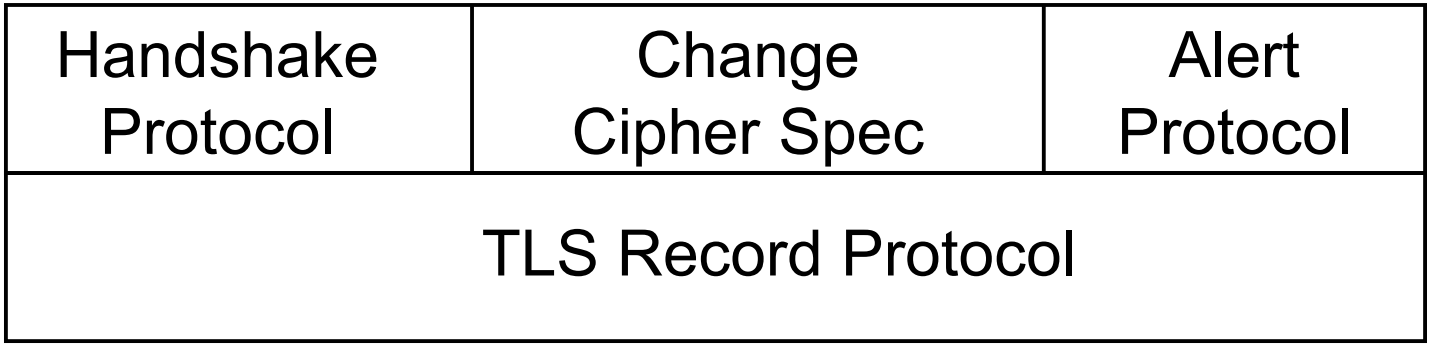
- İstemci ile sunucu arasındaki ilişkiyi düzenler.
Ortak kullanılan Doğrulama, güvenlik parametrelerini değiştirilmesi ile.
- Birden fazla bağlantıyı (coneection) içerebilr.
Ağır çalışma şartlarında: Bir sefer başlatıldıktan sonra.
-TLS Handshake Protocol'unu kullanır.
-Bir oturum aşağıdaki parametreler ile ile tanımlanır

Session id, peer certificate, compression method, cipher spec, server/client write key, initialization vectors, sequence numbers…

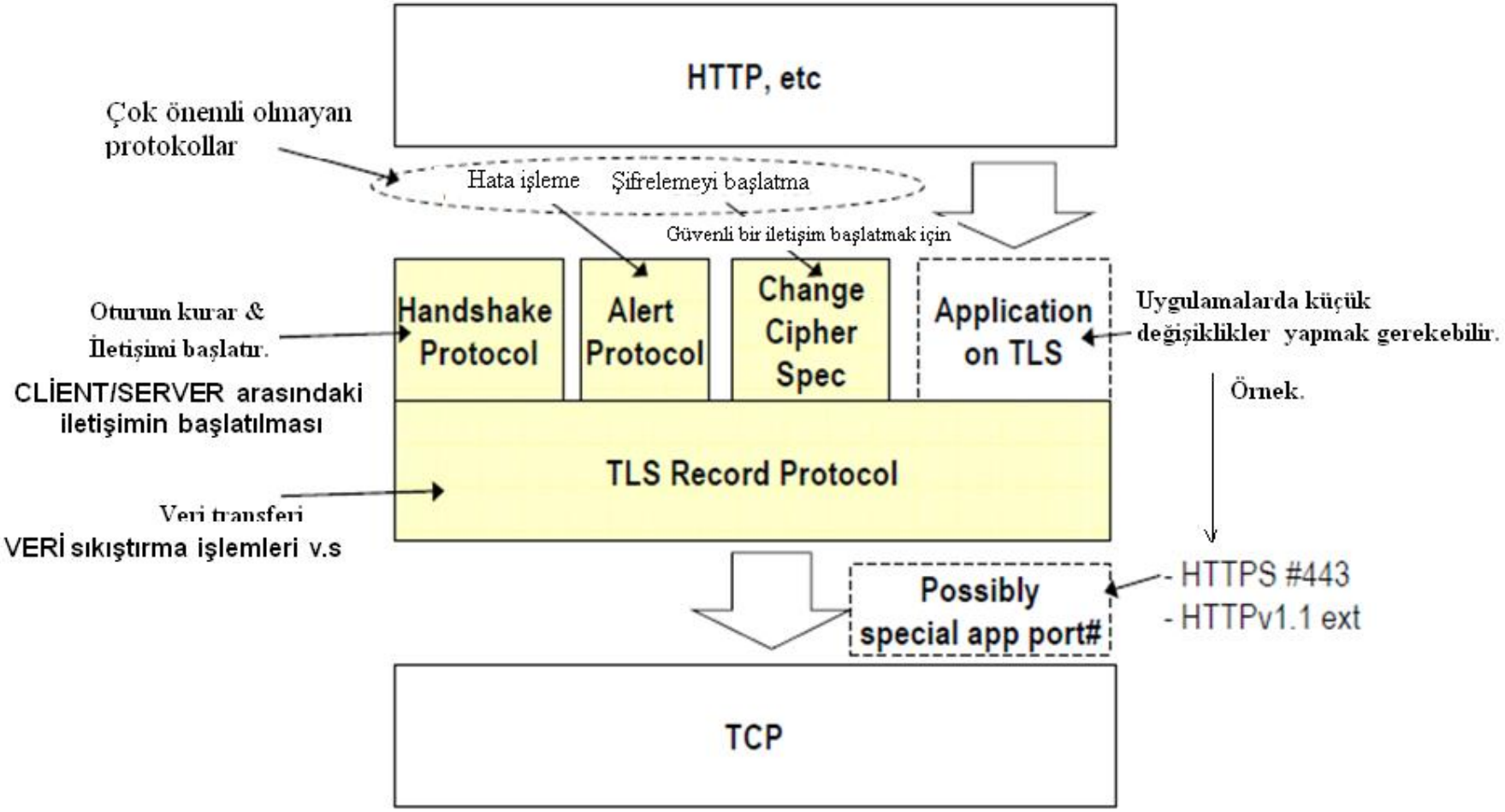
- Uygulama verisi iletildikten sonra TCP oturumu kapatılır. Ancak TCP ve SSL arasında doğrudan bir bağlantı olmadığından SSL durumu sürdürülebilir. Sunucu ve kullanıcı arasında yapılacak takip eden iletimler üzerinden uzlaşılmış parametreler ile gerçekleştirilebilir. Http için bu durum kullanıcının aynı sunucundan bir bağlantıya klikleyerek başka bir dökümanı talep etmesiyle yaşanır. Böyle bir durumda sunucu ve kullanıcı güvenlik parametreleri için başka bir uzlaşı gerçekleştirmeyecek ve daha önceki parametreler kullanılarak iletim gerçekleştirilecektir. SSL tanımı bu bilgilerin 24 saatten daha kısa bir süre tutulmasını önerir. Eğer bu zaman zarfında yeni bağlantı yapılmaz ise tutulmakta olan bilgiler silinir.

Mimari

- SSL mimarisi, üç adet üst seviye protokolundan oluşmaktadır. Uzlaşı (Handshake) Protokolü, CipherSpec Değişim (CipherSpec Exchange) Protokolü ve Uyarı- İkaz (Alert) Protokolü.
- Bir oturumun kurulması için Handshake Protokol'u kullanılır. (Kimlik doğrulama ve anahtar değişimi işlemi ile)
- Bekleyen şifreleme sırasını etkinleştirmek için Change Cipher Spec protokolu kullanılır.
- Record Protokolu TLS ve uygulama verilerini transfer etmek içindir.
- Şifrelemede veya diğer verilerde hata varsa bildirimi için Alert protokolu kullanılır.



SSL Mimarisi



SSL / TLS'nin desteklediği uygulamalar

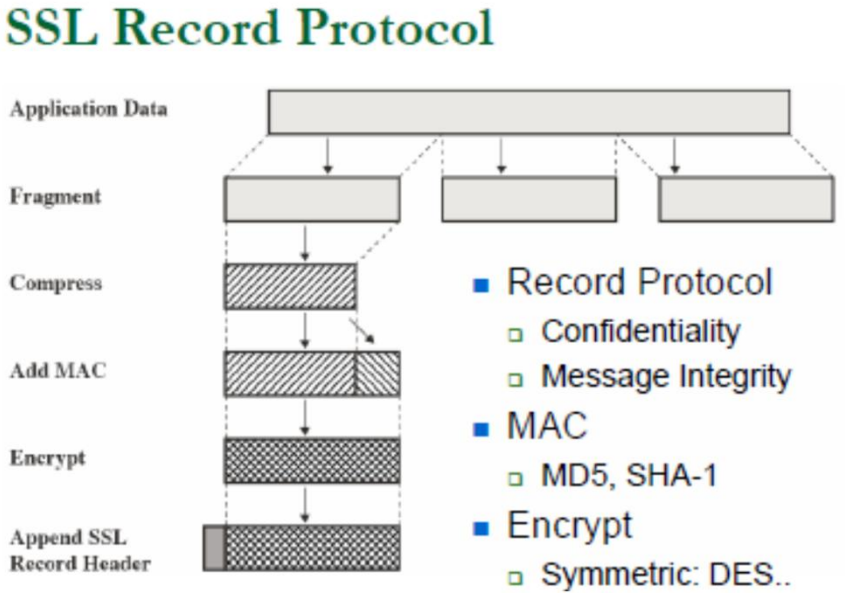
- **Typical approach: reserve a special port number for SSL/TLS mediated application**
  - Example:
    - →port 80 = HTTP over TCP
    - →Port 443 = HTTP over SSL/TLS (HTTPS)
- **SSL/TLS common application port numbers**
  - ssmtp 465
  - spop3 995
  - imaps 991
  - telnets 992
  - …
- **Alternative approach: slightly modify application to reuse traditional port number**
  - E.g. HTTPv1.1:
  - upgrade: TLS/1.0 new command (see RFC 2817)

SSL Kayıt Protokolü

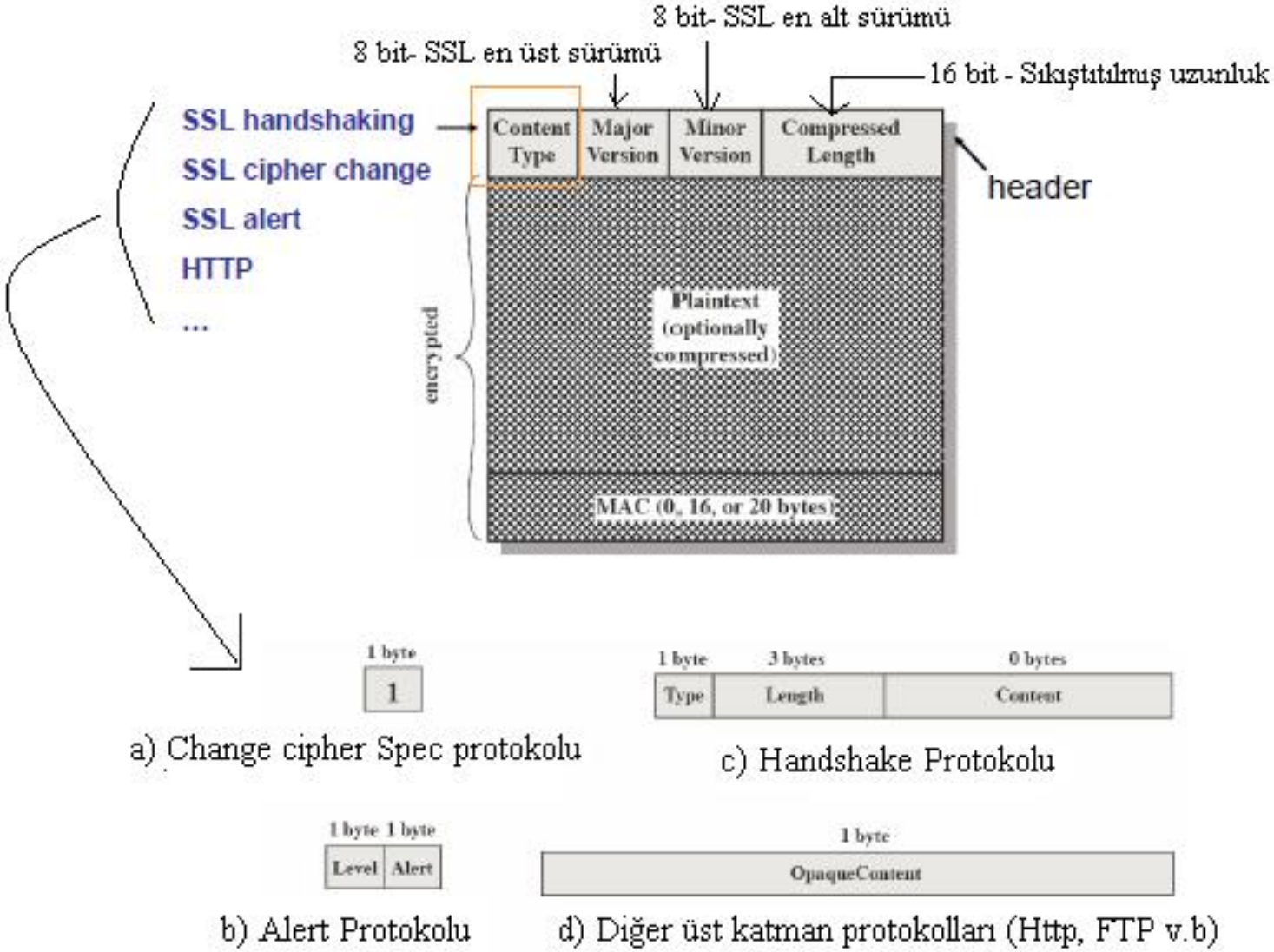
•SSL Kayıt Protokolü SSL bağlantıları için iki hizmet sunar.

-Uygulama katmanı verisinin kriptolanması ile güvenlik (Gizlilik), Bir mesaj doğrulama kodu *(Message Authentication Code, MAC) kullanılarak doğruluk sağlanır*.

***-Kayıt Protokolü, SSL'in bazı üst protokolleri* tarafından kullanılabilen temel bir protokoldür. Bunlardan biri, kriptolama ve doğrulama anahtarlarının alış verişi için kullanılan Uzlaşı (Handshake) Protokolüdür. Diğeri ise;**



Yandaki şekilde SSL Kayıt Protokolünün nasıl çalıştığı gösterilmektedir. Protokol, iletilecek uygulama mesajını alıp, yönetilebilir parçalara ayırdıktan sonra, seçime bağlı olarak sıkıştırır, bir MAC uygular, bir başlık ekler ve elde edilen paketi bir TCP yığını olarak gönderir. Alınan veri, çözülür, doğrulanır, sıkıştırılmışsa açılır ve tarayıcı gibi uygulama programına ulaştırılır.



SSL Kayıt FORMATI

CipherSpec Değişimi Protokolü

- Bu protokol değeri 1 olan tek Byte'lık bir mesajdan oluşur. Bu mesajın tek amacı bekleyen durumun (Pending) o anki durum üzerine kopyalanmasını sağlamaktır.
- Bu işaret bir koordinasyon işareti olarak kullanılır. Kullanıcı tarafından sunucuya ve sunucu tarafından kullanıcıya gönderilmelidir. Karşılıklı olarak tarafların bu işareti almasından sonra takip eden tüm mesajlar üzerinde anlaşılan şifreleme ve anahtarlar *(CipherSuite) ile alınıp verilirler.*

Alarm (Alert) Protokolü

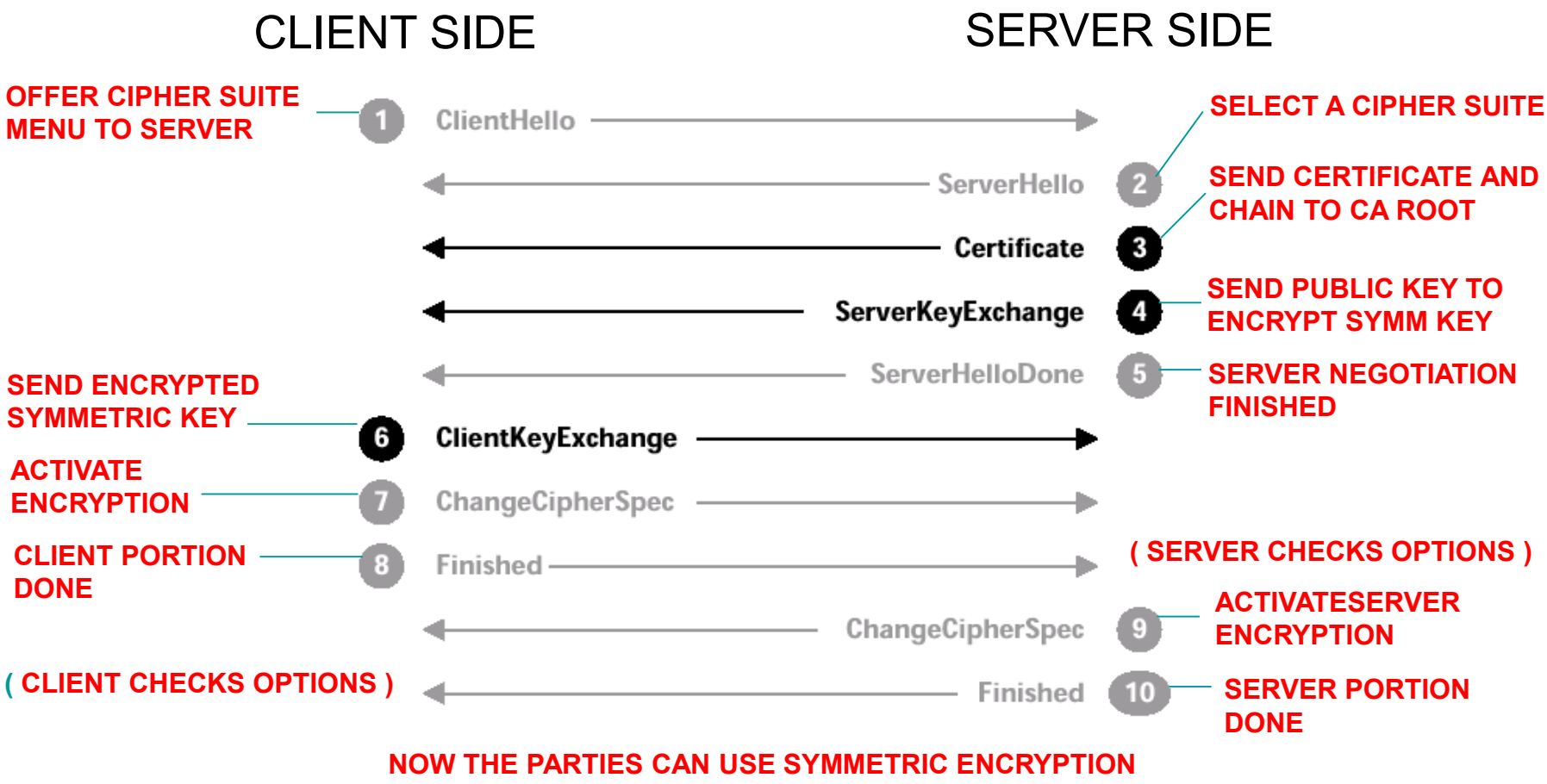
- Alarm Protokolü SSL ile ilgili alarmların uçlara taşınması için kullanılır.

- Bu protokolde her mesaj iki Byte'dan oluşur. İlk Byte, Uyarı - 1 *(Warning) veya* Ölümcül - 2 *(Fatal) değerleri ile mesajın önceliğini belirtir. Eğer değer 2 ise, SSL* bağlantıyı hemen keser. Aynı oturumda kurulmuş diğer bağlantılar devam edebilir ancak yenileri kurulmayabilir. İkinci Byte ise datayı belirten bir kod içerir.

Uzlaşı (Handshake) **Protokolü**

- SSL'in en karmaşık bölümü Uzlaşı Protokolüdür. Bu protokol, kullanıcı ve sunucunun birbirlerini doğrulamalarını, SSL kaydı içinde gönderilecek verinin korunması için kullanılacak kriptolama, MAC algoritması ile kriptografik anahtarların belirlenmesini sağlar.
- Uzlaşı Protokolü, herhangi bir uygulama veri iletilemeden önce kullanılır. Bu protokol sunucu ve kullanıcı arasında alıp verilen bir dizi mesajdan oluşur.
- Bu protokol yapısı dah çnce açık bie şekilde açıklanmıştır.

SSL Mesajlarının



SOURCE: THOMAS, *SSL AND TLS ESSENTIALS*

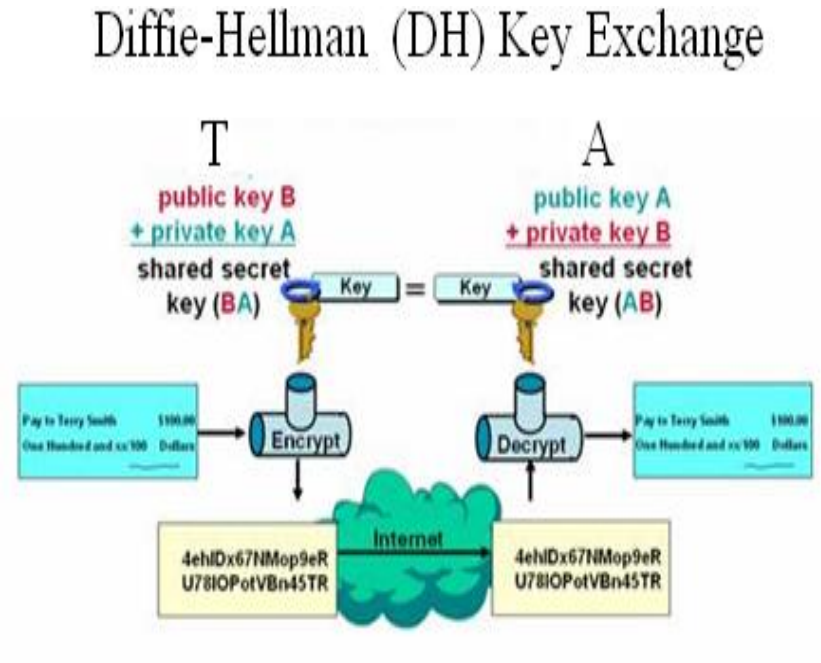
örnek

Tarayıcı SSL kullanan site adresini girer. Örneğin

1. https://www.firma.com
2. Tarayıcı ve sunucu aşağıdaki adımları içeren SSL el sıkışmayı gerçekler.

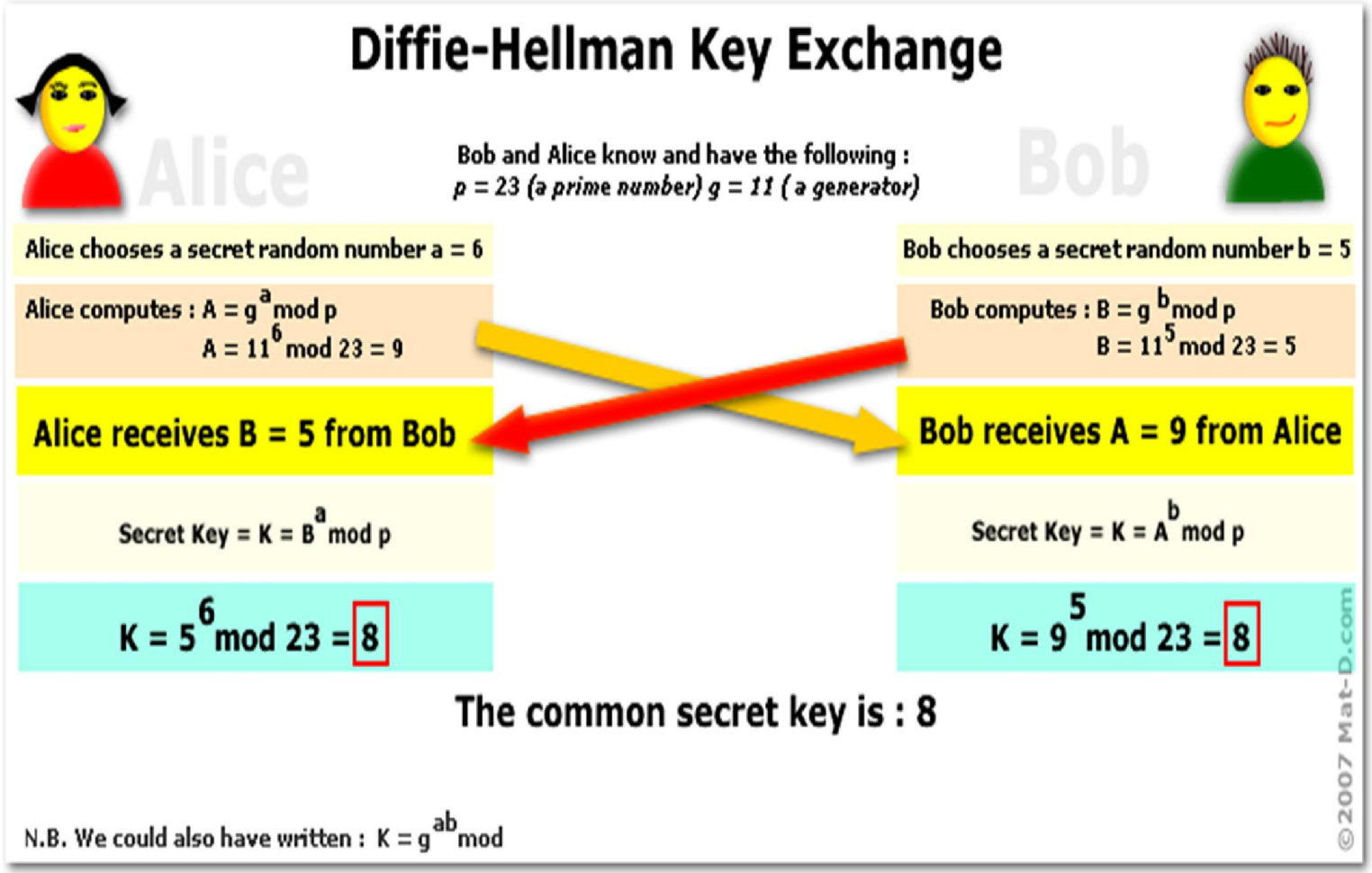
- Tarayıcı ve sunucu hangi şifreleme setini kullanacakları üzerinde anlaşırlar.
- Sunucu sertifkasını tarayıcıya yollar. Tarayıcı sertifika üzerinden sunucunun kimlik denetimini yapar, Sertifikada aynı zamanda sunucunun puplic key'ide vardır.
-Tarayıcı simetrik şifreleme anahtarını yaratır ve sunucuya yollar. Bu anahtar tarafların birbirlerine gönderdikleri verileri şifrelemeleri için kullanılır.

-Client/server arasındaki bu bir seferlik anahtar değişimi için RSA veya Diffi-Hellman algoritmaları kullanılır.



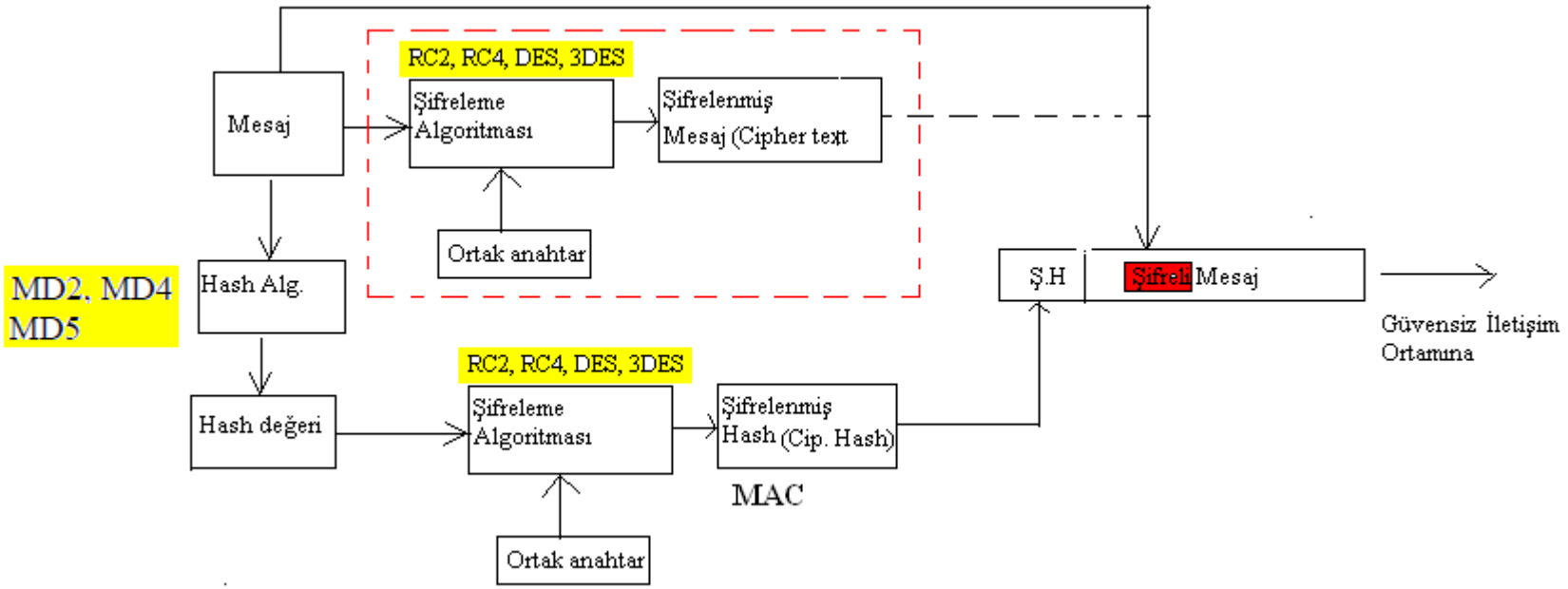
Yaratılan bu tek anahtarla şifreleme ve deşifreleme işlemi SİMETRİK ŞİFRELEME'dir. Simetrik Şifreleme basit olduğundan hızlı çalışır. Güvensiz bir ortamda seyahat eden verileri şifrelemek için kullanılır. Şekilde T A' ya veri göndermek istiyor. A, T'ye Public anahtarını gönderiyor. T kendi özel anahtarı (Private Key) ile A' nin kendisine gönderdiği Public anahtarı karıştırıp bir ortak anahtar elde ediyor (Shared Secret). T' bu anahtarı, A'nın public key'i ile şifreleyip A'ya gönderir. A bunu kendi privite anaharı ile açıp gönderilen bu anahtara sahip olur. Artık A ile T arasındaki şifreleşme bu ortak anahtarla olacaktır.

Diffie-Hellman gizli iletişimlerde kullanılabilecek ortak gizli anahtar üretir. Bu anahtar da ortak ağlarda (güvenli olmayan kanaldan) güvenli veri alışverişini sağlar. Alice ve Bob'un kendi gizli sayılarını belirli bir algoritmaya göre değiştirirler.. Sonunda her iki taraf matematiksel olarak arada dinleyen başka bir kişi tarafından geri döndürülmesi zor olan (bugünkü süper bilgisayarların mantıklı bir zamanda geri döndürememesi) aynı anahtarı elde eder. Bu aşamadan sonra Alice ve Bob oluşturmuş oldukarı ortak gizli anahtarla aralarındaki veri alışverişini şifrelemek ve deşifrelemek için kullanırlar.



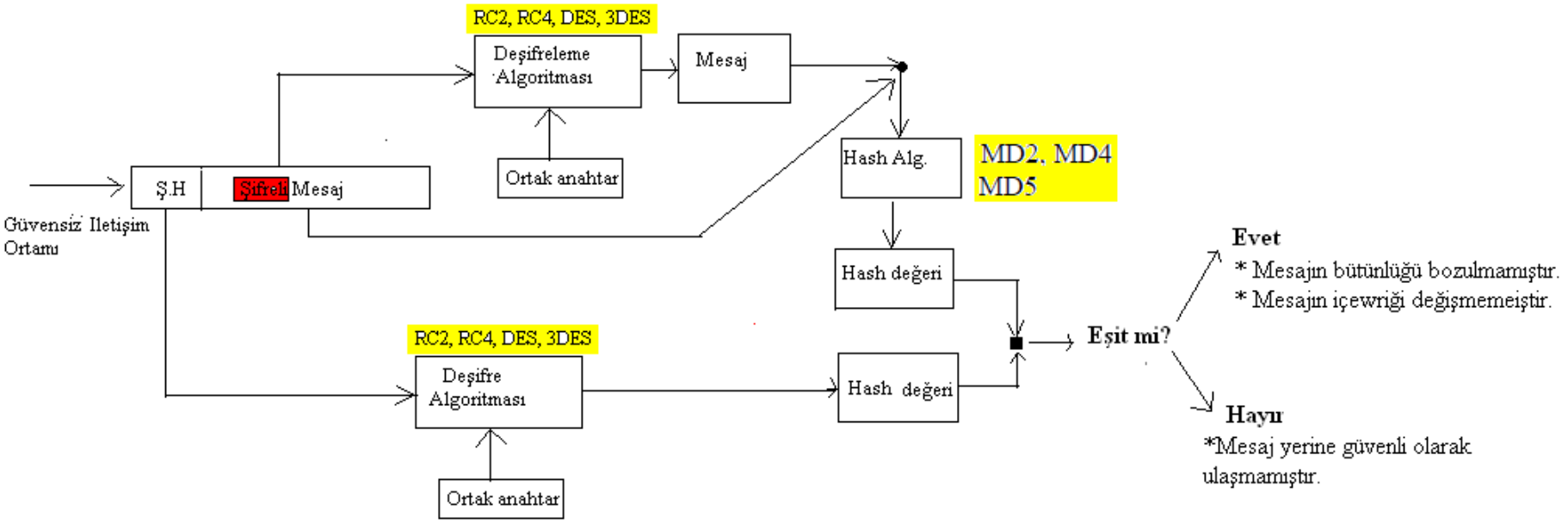
3. Sunucu , tarayıcıya istediği veriyi aşağıdaki aşamalardan geçerek yollar.

- Veri için hash değeri üretir
- Simetrik anahtar ile veriyi ve hash değerini şifreler
- Veriyi ve hash değerini tarayıcıya yollar.



4. Tarayıcı veriyi ve hash değerini alır ve aşağıdakileri yapar.

- Şifrelenmiş veriyi ve hash değerini çözer.
- Veri için hash değeri yaratır.
- İki hash değerini karşılaştırır ve aynı ise gelen veriyi gösterir.



SSL' Ne Kadar Güvenlidir?

I .Şifrenin Kırılması

- SSL çeşitli şifreleme tekniklerini destekler. RSA genel anahtar şifrelemesiyle oturum anahtarının değiş tokuşu ve istemci ile sunucunun doğrulanmaları sağlanır.
- Oturumun şifrelenmesi için kullanılan bir çok şifreleme algoritması vardır. Eğer bu kullanılan şifreleme yöntemlerine karşı başarılı bir saldırı mekanizması gerçekleştirilebilirse o zaman SSL in güvenliğinden bahsedilemez.
- Böyle bir saldırı mekanizması ise sadece oturumunun tamamıyla kopyalanması ve bu kopya üzerinde oturum şifrelerinin çözülmesi için uğraşmakla gerçeklenebilir. Fakat SSL böyle bir saldırı sonucunda elde edilecek faydanın bu saldırı için harcanacak emeğe, zamana ve paraya değmemesi için çalışmaktadır.

2. Replay

- Replay saldırı şekli esasında çok basit bir yapıya sahiptir. İstemci ve sunucu arasında gerçekleşen haberleşme oturumunun kopyalanarak daha sonra sunucuya bağlanıp oturum boyunca istemci tarafından gönderilen mesajların tekrar gönderilmesi esasına dayanır. SSL bu saldırı şeklini kullandığı bağlantı id'si ile karşı koyar. Bu bağlantı id'si kötü niyetli kişi tarafından bilinemediği için sunucuya gerekli cevapları veremez. Eğer geniş imkanlara sahip biri bu istemci ile sunucu arasında geçen birçok oturumu kopyalasa ve *Server-Hello* mesajında gönderilen bağlantı id'sini tahmin etmeye kalksa şansı gene çok azdır. Çünkü SSL bağlantı id'leri 128 bit büyüklüğündedir ve bu kötü niyetli kimse %50 şansının olabilmesi için en az 2^{64} oturum kopyalaması gerekmektedir.

3. Man in the Middle

The Man in the Middle saldırısı oturum boyunca üç kişinin (istemci, sunucu ve kötü niyetli kişi) yer alması ile gerçekleşebilir. Kötü niyetli kimse bu oturumda istemci ve sunucunun arasında bulunur ve istemci ile sunucu arasındaki mesaj trafiğini karıştırır. Kendini istemciye gerçek sunucuymuş gibi gösterir. Fakat bu saldırı şeklinin de SSL bağlantısında gerçekleşmesi imkansızdır. Çünkü SSL bağlantılarında sunucunun sahip olduğu bir sertifikası vardır. SSL Handshake protokolü esnasında bu sertifika istemci tarafından doğrulanır ve oturum daha sonra kurulur.