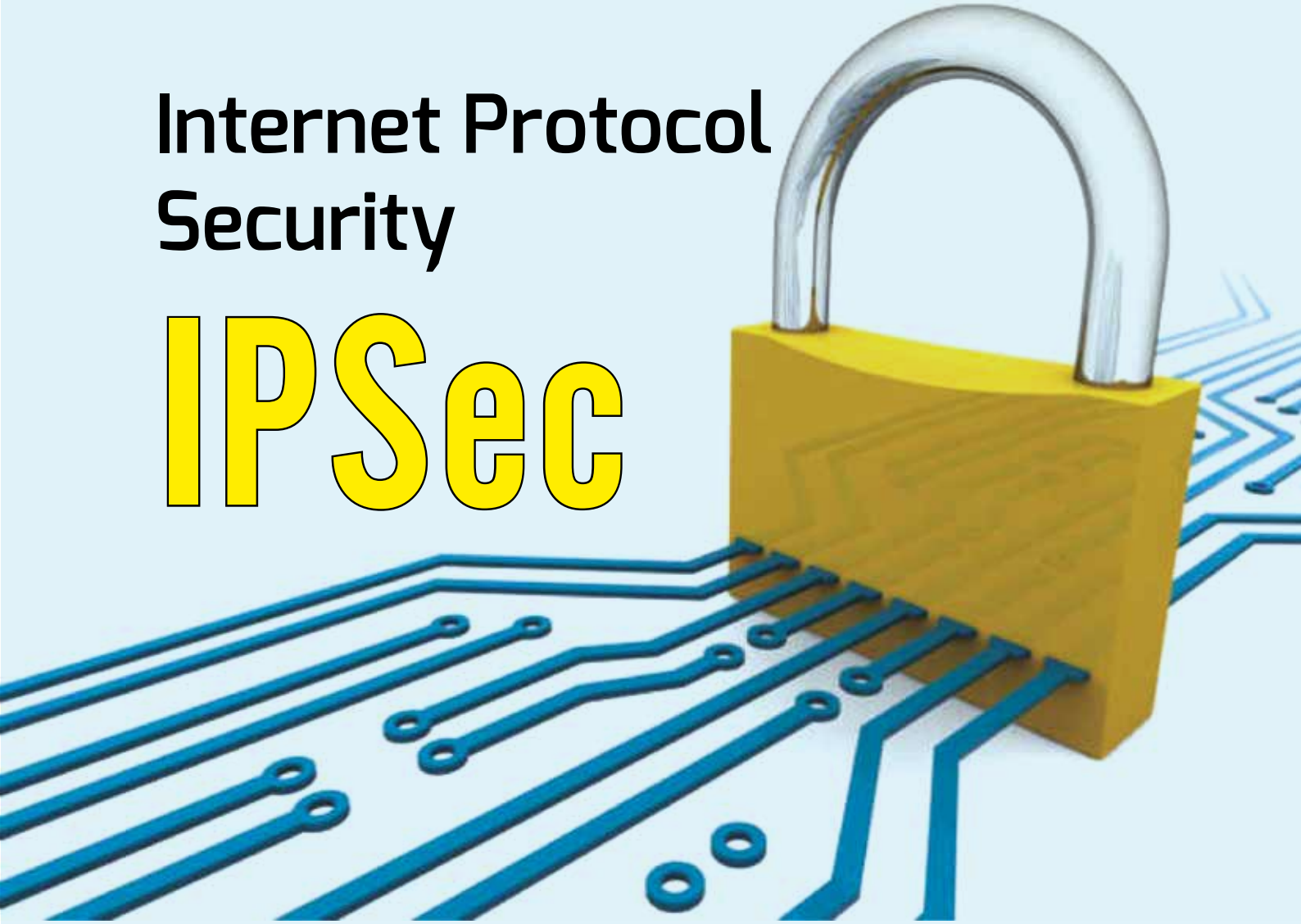
Internet Protocol Security

IPSec



Levent AKCASU *[Elektronik ve Haberleşme Müh.]*

Günümüzün dünyasında IP tabanlı haberleşme bir çığ gibi büyüyerek her yanımızı kapladı. Ses, video ve veri haberleşmesinde IP vazgeçilmez bir protokol haline gelirken, iletişimde güvenlik, gizlilik sorunları ve buna bağlı çözümleri de getirdi. Özellikle kişisel ve ticari bilgilerimizi iletişimimizde kullanma zorunluluklarımız, iletişim anında güvenlik ve gizlilik açısından riskler barındırmaktadır. Bir ticari faaliyet veya bankayla internet üzerinden haberleşmemiz esnasında gönderdiğimiz ve aldığımız verilerin gizliliği ve güvenliği hayati önem taşır. Bilgi güvenliğinin sağlanmasından anladığımız ise kimlik tanıma, gönderilen/ alınan bilginin gizliliği ve doğruluğudur. Tüm bunları sağlamak için IP iletişiminde internet Protokol Güvenliği geliştirilmiştir.

Internet Protokolu güvenliği (IPSec), şifreleme güvenlik hizmetlerini kullanarak Internet Protokolü (IP) ağları üzerinden özel, güvenli bir iletişim sağlamaya yönelik açık standartlar çerçevesidir.

IPSecurity (IPSec), veriyi şifreleyen (encryption), verinin bütünlüğünü sağlayan (integrity), göndericinin kimlik doğrulamasını (authentication) ve verinin network üzerinde güvenli iletimini (Secure transmission) sağlayan bir endüstri standardıdır. IPSec, 3. katmanda (layer)'da çalışır . Uçtan uca (Transport mod) çalışabildiği gibi network'den network'e de (Tunel mod) çalışabilir.

Bir veri alış verişinde, güvenlik anlamında beklediklerimiz;

- Karşımızdakinin yani iletişimde bulunduğumuz muhatabımızın kimliğinden emin olmak. Örnek; bir banka ile iletişime geçtiğimizde gerçekten iletişime geçtiğimizin gerçekten o banka olduğuna emin olmamız gerek. Keza banka içinde aynı şey geçerli, banka da sanal ortamda iletişimde olduğu kişiden emin olmalı. Sertifikasyon, kerberos, preshared key ve dijital imza yöntemleri kimlik tanıma yöntemleri olarak geliştirilmiştir.
- Bize gönderilen verinin/veya gönderdiğimiz verinin doğru, değiştirilmemiş ve tam kendisi olduğuna emin olmak. Bu işem, HASH algoritmaları ile CHECKSUM hesabı yapılarak sağlanır
- Bize gönderilen veya bizim gönderdiğimiz verinin gizliliğinden emin olmak. Yani iletişim esnasında aldığımız veya gönderdiğimiz verinin 3. Şahıslar tarafından ele geçirilmesi durumunda, çözülmesinin en azından belli süre içinde çözülmesinin olanaksız olması. Bu işlem verinin şifrelenmesi ile sağlanır. Belli süre kavramı muğlak bir kavram gözükse de burada amaç, gönderdiğimiz bir verinin işlevinin devam

ettiği süre içinde çözülememesi bizim için yeterlidir. Şifreleme uzmanlarına göre teorik olarak çözülemeyecek şifrenin olmadığı gerçeğini bilerek belli süre kavramını kullandım. Örnek verirsek; bir banka işleminde gönderdiğim verinin aylar hatta belki de saatler sonra çözülmesi bizim için bir tehdit değildir. Önemli olan o verinin işlevinin devam ettiği zaman içerisinde yani bankayla iletişim esnasındaki aynı oturum süresince çözülememesidir.

IP güvenliğinde bu 3 şartın nasıl sağlandığında geçmeden önce şifreleme tekniğinden kısaca bahsedelim. Verinin şifrelenmesi, hepimizin bildiği gibi verinin sadece alıcı ve verici tarafından anlaşılacağı, 3. Kişilerin eline geçse bile anlaşılması (belli sürede) olanaksız bir şekilde verinin üzerinde işlemler yapılmasıdır.

Eğer gönderici ve alıcı veriyi açmak için aynı metodu ve anahtarı kullanıyorsa buna simetrik şifreleme adını veririz. Bunun için daha önce alıcı ve verici anlaşmış olmalı ve anahtarları paylaşıp gizli tutmalılar. Örnek olarak, bildiğiniz gibi veriler aslında 1 ve 0 lardan oluşur. Eğer 1 ve 0 lardan oluşan bir anahtarımız varsa, gönderici gönderdiği veriyi, sadede alıcı ve vericinin bildiği bir anahtarla lojik terimi olan XOR ile işleme sokar, alıcı ise değiştirilmiş veriyi gene aynı anahtar ile XOR'layarak orijinal veriyi elde eder.

Örnek: Veri 10011010 olsun Anahtarımız ise 00110011

10011010 (XOR) 00110001 = 10101011 (şifrelenmiş veri)

Alıcı taraf da; (şifrelenmiş veri) (XOR) anahtar = 10101011 (XOR) 00110001 = 10011010 (orjinal veri)

Bu iş de zorluk ise; anahtar üzerinde önceden anlaşıp bu anahtarı gizli tutmaktır. Daha önceki haberleşmeden elde edilen bilgilerle yeterli zaman içinde bu şifre çözüldüğünde, yeniden aynı anahtarla haberleşmeye başlandığında 3. şahıslar tarafından veriler ele geçer. Bunu önlemek için ise anahtarı çok sık değiştirip belki her oturum başında anahtarı tekrar yenilemek gereklidir. Banka örneğine dönersek her bankaya bağlandığımızda yeni bir şifreleme anahtarını önceden bankayla konuşup yenilemek gibi bir zorunluluk getirir. Bunun zorlukları açık. Bu anahtarları belli bir mantığa göre değiştirmek bir çözüm gibi gözükse de üzerinde çalışmak için yeterli süre olduğu için kötü amaçlı 3. şahıslar bu mantığı da kolayca çözerler. Belki 2 taraf da olan ve senkron çalışan, her oturumda anahtarı rasgele değiştiren aletler veya yazılımlar çare olabilir. Böyle bir çözümde ise hepimizin her gizlilik isteyen işler için elimizde çok sayıda karşı tarafla senkron çalışan aletlerle veya yazılımlarla donatılmamız gerekir.

Diğer bir çözüm ise ilk başta pek akla uygun gelmese de asimetrik anahtarlamadır. Bu yöntemde bir veri bir anahtarla şifrelendiğinde, artık o anahtarla tekrar açılamaz sadece onun bir ikizi, eşleniği olan başka anahtarla açılabilir. Yani kapımız kilitliyoruz ama kapıyı kilitlediğimiz anahtar açamıyor sadece başka bir anahtar açıyor. Bu ilk başta matematiksel olarak olanaksız gibi gibi gözükse de mümkün. 1978`de bulunan RSA, yaratıcılarının baş harflerini almıştır: Ronald Rivest, Adi Shamir ve Leonard Adleman. RSA günümüzde en çok kullanılan açık anahtar algoritmasıdır. Ayrıca RSA en çok test edilen algoritmalardan biridir. RSA hem bilgi şifreleme de hem de dijital imza sistemlerinde kullanılır. Bu şifreleme mantığı büyük asal sayıların asal çarpanları ve modüler matematiğe dayanır. Sonuçta elimizde birbirinin eşleneği olan 2 anahtar var ve bu anahtarların birisi ile şifrelenen bilgi ancak diğer anahtarla açılabilir. Burada önemli olan bu anahtarların birisini bilenin diğer eşlenik anahtarı çıkartmasının makul sürede olanaksız olması. RSA sistemine göre elde edilen eşlenik anahtarlardan birisini bildiğimizde diğerinin ne olabileceğini hesaplamak yıllar sürecek bir uğraş gerektirdiği söylenir. Eşlenik anahtarlar sistemi yani asimetrik anahtarlamanın pratikteki faydası ise bu, bu anahtarların birisi açık anahtar olarak kullanılır ve bu anahtara isteyen herkes ulaşabilir. Fakat bu anahtara ulaşmak şifrelenmiş bilgiyi açmaya yetmez. Diğer anahtarı da bilmek gerekir. Örnek bir bankaya bağlandığınızda, banka size açık olan anahtarı rahatlıkla gönderir. Bunun başkalarının eline geçmesi tehlike içermez. Siz bilginizi bu anahtarla şifrelediğinizde 3. şahıslar diğer anahtarı bilmediği için bilginin orijinal halini göremez. Sadece bu anahtarı size gönderen (örnekte banka) bu bilgiyi kendisinde olan 2. anahtarla açabilir. Bu anahtarlardan herkese açık olan anahtara public (genel), diğer anahtara ise private (özel) anahtar denir.

veri <---> private key <---> sifrelenmis veri <---> public key <---> veri

Bu anahtar ikililerinin şifreleme veya e-imza olarak kullanımındaki aşamaları örneklerle açıklamak istersek; biraz beyin cimnastiği yapmamız gerekecek.

Örnek A kişisi, B kişisine şifrelenmiş bir mesaj atmak istiyor. Öncelikle A kişisi B kişisinin puplic anahtarını ister ve elde eder. Bu anahtarın mesajlaşmadan hemen önce B'den istenmesinin bir mahsuru yoktur çünkü bu anahtarı elde edebilecek 3. şahıslar olsa bile mesajlaşma anında buna makul süre diyebiliriz, diğer anahtarı elde etmesine olanak yoktur.

A ---> Mesaj --- Public_B ---> [Mesaj]Public_B -->B (mesaj, B'nin public anahtarıyla şifrelenerek B'ye gönderildi.)

[Mesaj]Public_B --- Private_B---> Mesaj (şifrelenmiş mesaj B'nin private anahtarı ile çözüldü)

Elektronik İmza ve Kimlik Doğrulama:

Elektronik imza, gelen bir mesajın (verinin) gerçekten gönderen kişiden geldiğinin doğruluğunu ispatlama (authentication) mekanizmasıdır. Hiçbirimiz kendi adımızın kullanarak başkası ile iletişim kurmasını istemez. Bu yüzden e-imza yöntemi geliştirilmiştir. A kişisi elektronik imzaya sahipse, özel anahtarı vardır ve mesajını özel anahtarıyla imzalar.

A ---> Mesaj --- Pri_A ---> [Mesaj]Pri_A

Şifrelenmiş mesajı ve genel (public) anahtarı bir paket yapıp B'ye gönderir

A ---> [[Mesaj]Pri_A + Pub_A] ---> B

Mesajı alan B, A'nın public anahtarını kullanarak mesajı açar ve mesajın A tarafından gönderildiğinden emin olur.

[Mesaj]Pri_A --- Pub_A ---> Mesaj

Fakat görüldüğü gibi A'nın gönderdiği mesaj, 3. şahısların eline geçtiğinde Public anahtar da geçeceği için bu mesaj deşifre edilir. Bunu önlemek için public/private anahtar ikilileri, simetrik anahtarlama ve hash algoritmaları birlikte kullanılır.

Hikayemizde A'nın, B'ye hem şifreli hem de imzalı bir mesaj göndermek istediğini düşünelim. Mesajın imzalanması ve mesajın şifrelenmesi aynı işlem içerisinde iki ayrı bölüm olarak incelenir.

A mesajı imzalamadan önce mesaj üstünde hash algoritması çalıştırarak mesajin bir kontrol toplamını (checksum) hesaplatır.

A ---> Mesaj --- hash alg. ---> Toplam = [Mesaj]Hash

Kontrol toplamı A'nın private anahtarı ile şifrelenir

Toplam --- Pri_A ---> [Toplam]Pri_A

Oluşturulan şifreli toplam, mesajın kendisi, A'nın public anahtarı ve kullanılan hash algoritmasının ismi hep beraber bir paket oluşturur. Kullanılan Hash algoritmasının bilgisi önemlidir çünkü bir çok yöntem vardır. (MD5, SHA1 vb)

[Mesaj + [Toplam]Pri_A +Pub_A + hash alg] ------->B

Bu şekilde gönderilecek mesajın imzalanma işlemi tamamlanmış olur. Bu oluşturulan paketin içerisinden B, A'nın public/private ikilisinden mesajı A'nın gönderdiğini anlar. Kendisine gönderilen paketten public anahtarı kullanarak toplamı elde eder. Gönderilen çıplak mesajdan da ikinci bir toplam elde etmek için, ismi gönderilen hash algoritmasını çıplak mesaj üzerinde tekrar çalıştırır. Elde bulunan iki toplamı karşılaştırarak mesajın değişip değişmediği bilgisini öğrenir. Bu iki toplam birbiri ile aynı ise mesaj değiştirilmemiştir ve A tarafından gönderilmiştir. İki toplam birbirinden farklı ise mesaj A tarafından gönderilmemiştir veya mesaj yolda değiştirilmiştir, diye düşünür ve mesajı dikkate almaz ama burada da mesaj 3. şahıslara açıktır.

Asimetrik anahtarlamada işlemler yavaş olmakta, o yüzden uzun mesajlaşmalarda simetrik anahtarlamaya ihtiyaç duyululmaktadır. Asimetrik anahtarlama karşı tarafa hem kimlik tanımlama, hash algoritmasıyla bilginin değişmezliğinin sağlanması ve simetrik anahtarlamayı göndermek için kullanılırsa ve ondan sonraki mesajlaşmalar simetrik anahtarlamayla yapılırsa, hem güvenlik hem de hızlı mesajlaşma sağlanır.

O yüzden toplam paketimiz önce simetrik anahtarla şifrelenir.

[Mesaj + [Toplam]Pri_A +Pub_A + hash alg]symK

Fakat bu toplam paketi açmak içinde B'ye simetrik anahtarı da göndermek gereklidir. Yani paketimiz

[Mesaj + [Toplam]Pri_A +Pub_A + hash alg]symK + symK olmalı.

Fakat hala 3. şahıslar için mesaj açıktır. Çünkü bu pakette 3. şahıslar simetrik anahtarı elde edebilirler ve tüm paket açılabilir. O yüzden önce B'den B'nin public anahtarı istenir ve simetrik anahtarlama bilgisi bu anahtarla şifrelenir.

A----->[Mesaj + [Toplam]Pri_A +Pub_A + hash alg]symK + [symK]Pub_B --->B

Bu mesajı alan herhangi biri B'nin private anahtarına sahip olmadığı için simetrik anahtara ulaşamayacak, yani gönderilen mesajı okuyamayacaktır. Bu mesajı alan B önce kendi private anahtarı ile simetrik anahtarı elde edecek, bu simetrik anahtarı kullanarak paketi açacak, paketle gelen A'nın public anahtarını ve hash algoritmasını kullanarak, mesajın değiştirilip değiştirilmediğini ve bu mesajı A'nın gönderip göndermediği bilgisini elde edecektir.

Bu yöntemi kullanarak, mesajın ağda değiştirilmesi, ağı dinleyen herhangi biri tarafından okunması, başkası adına mesaj gönderme gibi tehditler unsurlarına karşı önlem alınmış olur.

Sonuçta bu şekilde IPSEC'in 3 şartı:

- Verinin şifrelenerek 3. şahıslardan gizlenmesini
- Hash algoritmasıyla bilginin bütünlüğünün (eksik/fazla değiştirilip değiştirilmediği) sağlanması
- Kimlik doğrulanması

Sağlanmış olur.