SANAL ÖZEL AĞ TASARIMI VE GERÇEKLEMESİ

Ender YÜKSEL[1] **Bülent ÖRENCİK**[2]

[1,2]Bilgisayar Mühendisliği Bölümü
Elektrik-Elektronik Fakültesi
İstanbul Teknik Üniversitesi, 80626, Maslak, İstanbul

[1]e-posta: ender@cs.itu.edu.tr [2] e-posta: orencik@cs.itu.edu.tr

Anahtar sözcükler: Sanal Özel Ağ, Asıllama, Sayısal İmza, Kapsülleme, Tünelleme, Şifreleme

ABSTRACT

In this study a Virtual Private Network (VPN) software is designed and implemented on the physical infrastructure of Istanbul Technical University. Project is developed on Microsoft Windows Platform using Microsoft's Libraries and an open source library called WinPCap. The objective was to accomplish the main tasks of a VPN such as encapsulation, encryption and authentication. The aim is achieved and the VPN, including authentication, data signing, key exchange, encryption and encapsulation is implemented.

1. GİRİŞ

Güvenli bir ağa gereksinimi olan kurumların iletim hatları kiralamak gibi bir seçeneği vardır. Bu seçenek güvenli gibi görünse de pahalı olması, esnek olmaması ve uzaktan bağlanan kullanıcılara destek kolaylığı sağlamaması gibi yönlere de sahiptir. Alternatif bir çözüm ise gittikçe daha fazla benimsenen Sanal Özel Ağ'dır (SÖA) [1].

SÖA'nın gücü, verileri mevcut güvensiz genel iletişim altyapısı üzerinden güvenli ve güvenilir bir şekilde iletebilmesinden gelmektedir. Veriler, paket anahtarlamalı güvensiz bir ağ üzerinde şifrelenmiş ve noktadan noktaya tünellenmiş olarak iletilirler. Alıcı tarafında deşifre edilen veriler, gerekiyorsa filtrelenir ve veri bütünlükleri kontrol edilir. SÖA, ağ kullanıcılarına ucuz, güvenli ve ölçeklenebilir bir güvenlik çözümü sağlar [1-3-4-11].

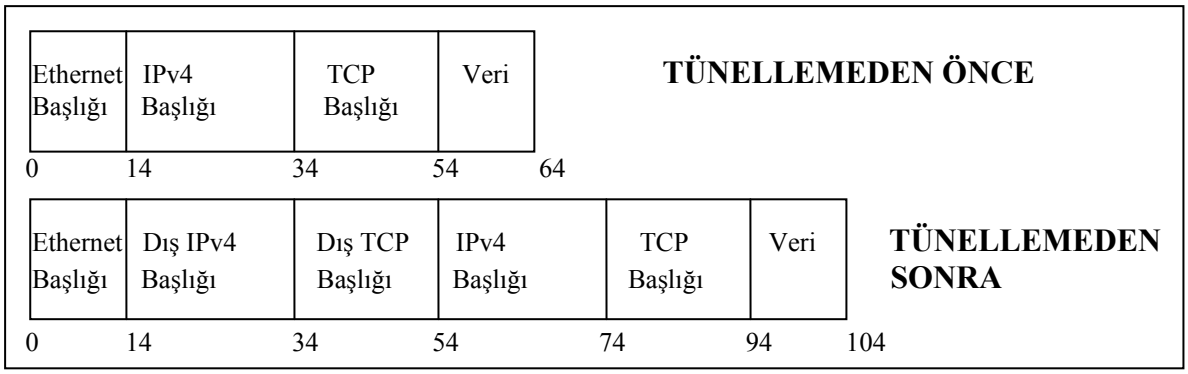
Bu çalışmada, İstanbul Teknik Üniversitesi fiziksel altyapısı üzerinde bir Sanal Özel Ağ Yazılımı tasarlanmış ve gerçeklenmiştir. Proje, Microsoft Windows Platformunda, Microsoft kütüphaneleri ve WinPCap adlı açık kaynak kodlu bir kütüphane kullanılarak geliştirilmiştir. Projenin amacı, Sanal Özel Ağ'ı oluşturan şifreleme, asıllama ve kapsülleme gibi ana unsurları başarmaktır. Bu çerçevede veri imzalama, anahtar değişimi, asıllama, şifreleme ve kapsülleme yapabilen bir Sanal Özel Ağ oluşturulmuştur.

2. SANAL ÖZEL AĞ TASARIMI

2.1 TÜNELLEME

Tünelleme, SÖA'nın sanal kısmını oluşturan yapıdır. Paketin başlıklarının önüne yeni başlıklar ekleyerek, taşındığı ağdaki olası izleyicilerden gizlenmesini sağlar. Bunun sonucunda, yönlendiriciler sadece sonradan eklenen başlıkları görür ve onlara göre paketi yönlendirir.

Tünelleme SÖA sunucu uygulamasında gerçekleşecek şekilde tasarlanmıştır. Gelen paketler yeni bir TCP-IP paketinin içine gömülür ve alıcı SÖA sunucusuna gönderilir. Kapsülleme işleminde yeni paketin eskisinden 2 başlık boyu kadar büyük olacağı açıktır. Yeni paketin ethernet başlığında kaynak SÖA sunucusunun ve ağ geçidinin MAC adresleri bulunur. Ipv4 başlığında ise yine SÖA sunucularının IP adresleri bulunur ve toplam sınaması değeri tekrar hesaplanır. TCP başlığının da SÖA sunucularının senkronize çalışabilecekleri şekilde biçimlendirilmesi gerekir. Orijinal paket ise IPv4 ve TCP başlıklarıyla beraber yeni paketin veri kısmını oluşturur [7-8-9-10-12] (Şekil 1).



Şekil 1 - Tünelleme

2.2 ASILLAMA

Güvenli iletişimde haberleşen tarafların birbirlerinin kimliğini bilmesi esastır. Asıllama, haberleşen tarafların kimliklerinin doğrulanması işlemini içerir.

Asıllama, anahtar değişimini gerçekleştiren yazılımda çalışacak şekilde tasarlanmıştır. SÖA sunucularının asıllanması için sayısal imza kullanılmıştır. İki taraftaki SÖA sunucusunun her birinde veri imzalama işlemi için kaynak koduna gömülü bir veri bulunur. İmzalama işleminden önce bu verinin özü alınır, ardından bu öz kullanılarak imza elde edilir. Asıllama işlemi anahtar değişiminden hemen önce gerçekleştirilmektedir.

2.3 ŞİFRELEME

Kriptografik işlemlerin temel öğelerinden biri de kriptografik anahtarlardır. Oturum anahtarları ve açık/gizli anahtar çiftleri olmak üzere iki tip kriptografik anahtar vardır. Oturum anahtarları simetrik anahtarlardır bir başka deyişle, şifreleme ve şifre çözme için aynı anahtar kullanılır. Simetrik algoritmalar, açık anahtarlı algoritmalardan çok daha hızlıdır ve büyük boyuttaki verilerin şifrelenmesinde kullanılırlar. RC2, RC4, DES, 3-DES gibi simetrik algoritmalar yaygın olarak kullanılmaktadır. Açık/gizli anahtar çiftleri ise daha güvenli bir şifreleme metodu olan açık anahtarlı şifrelemede kullanılan anahtarlardır. Gizli anahtarın gizli ve özel olması gerekirken açık anahtar, isteyen her kullanıcıya dağıtılabilir. Anahtar çiftindeki açık anahtar genellikle sayısal sertifika yoluyla dağıtılır. Anahtar çiftindeki bir anahtarla şifrelenen mesajı çözebilmek için diğer anahtar gereklidir. Bir kullanıcının açık anahtarıyla şifrelenen mesajı sadece kullanıcının kendisi çözebilir. Bir kullanıcının açık anahtarı ile çözülebilen mesaj ise mesajı o kullanıcının şifrelediğini gösterir. Ancak açık anahtarlı algoritmalar simetrik algoritmalara göre oldukça (kabaca 1000 kere) yavaştır bu yüzden büyük verilerin şifrelenmesinde kullanılamazlar. Pratikte açık anahtarlı algoritmalar oturum anahtarlarının şifrelenmesinde kullanılır. Şifreli iletişimde ilk adım açık anahtarların değiştirilmesidir. Ancak bundan sonra kullanıcılar şifrelenmiş ve imzalanmış veri iletişiminde bulunabilirler [5].

Şifreleme, güvenlik açısından önemli bir konudur. SÖA sunucuları arasındaki veri iletiminin simetrik şifreleme ile güvenli hale getirilmesi gerekmektedir. Bunun için her sunucuda aynı gizli anahtarın bulunması gerekir. Üstünde çalışılan tasarımda SÖA sunucuları (Challenger ve Dublin olarak isimlendirilmişlerdir) veri şifrelemek için bir gizli anahtar kullanırlar. Bu noktada önemli bir konu daha belirir: Anahtar Değişimi. Sunucuların şifreleme ve şifre çözmede kullanacakları anahtarın güvenli bir şekilde değiş tokuş edilmesi gerekir çünkü şifre kötü niyetli birisinin eline geçtikten sonra güvenli bir iletişimden bahsedilemez. Tasarımda anahtar değişimi asimetrik şifreleme kullanılarak yapılmaktadır.

3. GERÇEKLEME

Bu çalışmada, hiçbir atanmış donanım kullanmadan sadece yazılım kullanarak gerçek bir ağ ortamı olan İTÜ yerel ağında bir sanal özel ağ gerçeklenmiştir. Piyasada kullanılan gerçeklemelerin çoğunda özel ağ geçitleri, özel güvenlik duvarları ve atanmış bilgisayarlar gibi özel donanımlar kullanıldığından, sadece yazılım kullanan bu çalışma ekonomik bir yaklaşımı sergilemektedir.

SÖA uygulamalarında platform seçimi önemli bir konudur ve bu tip uygulamalar genellikle Linux tabanlı işletim sistemlerinde geliştirilir. Bizim çalışmamızda ise farklı bir yol seçilerek Windows tabanlı sistemler için WinSock gibi Windows tabanlı teknolojiler kullanılmıştır [2].

SÖA gerçeklemek için bazı temel noktaları kotarmak gerekir. Bunlar: Kapsülleme, Şifreleme ve Asıllama'dır.

Kapsülleme, paketi alıp yeni başlık dosyaları eklemek gerektirdiğinden önemli ve zor bir konudur. Paketi alabilmek için ağ geçidine erişebilmek, ağ geçidinin üzerinde çalışabilmek veya ağ geçidini gerçeklemek gerekir. Fakat bu

durum programı ciddi anlamda kısıtlar, esnekliği ve gerçeklemeyi zorlaştırır. Bu çalışmada, sözü edilen sorunun önüne geçmek için paketlerin yakalanması yoluna gidilmiş ve herhangi bir ağ geçidi yapısıyla çalışabilecek bir program ortaya çıkarılmıştır. Paket yakalamak için bir paket yakalayıcı yazılmış, kapsülleme de dahil tüm bu işlemler Politecnico di Torino Üniversite'sinde geliştirilmiş açık kaynak kodlu bir paket yakalama kütüphanesi olan WinPCap kullanılarak gerçeklenmiştir.

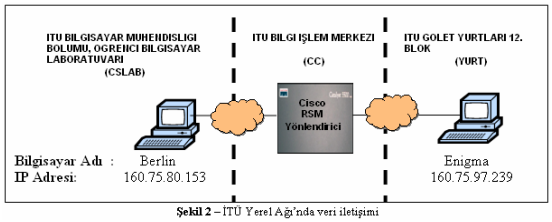
SÖA uygulamasında şifreleme, güvenlik açısından çok önemlidir. Sunucuların şifreleme ve şifre çözmede kullanacakları anahtarın güvenli bir şekilde değiş tokuş edilmesi gerekir. Anahtar değişimi asimetrik şifreleme kullanılarak gerçeklenmiştir ve tüm bunlar için Windows CryptoAPI kullanılmıştır. Gerçeklemede sunucuların özellikleri göz önüne alınarak daha uzun anahtarlar üretebilen Challenger isimli sunucu oturum anahtarına sahip taraf olarak belirlenmiştir. Challenger, şifreli veri iletimi için oturum anahtarını Dublin'e göndermelidir. Dublin'in açık anahtarını alan Challenger oturum anahtarını bu anahtarla şifreleyerek Dublin'e gönderir. Dublin kendi gizli anahtarı ile şifreli veriyi çözer ve Challenger'in önerdiği oturum anahtarını elde eder. Anahtar değişimi ve şifreleme işlemleri MS WinSock ve MS CryptoAPI kullanılarak gerçeklenmiştir. Yeni paketin içindeki veriler, daha doğrusu eski paketin tamamı SÖA sunucularında RC4 ile şifrelenmektedir [6].

Asıllama, yetkilendirilmemiş kullanıcıları SÖA'nın dışında tutar. Bu çalışmada, asıllamayı gerçeklemek için sunucular arasında sayısal imza kullanılmıştır. Tasarıma özel olarak istemci konaklar ve sunucular yerine sadece sunucular arası asıllama yapılmıştır. Ayrıca başarımı arttırmak için sadece anahtar değiş tokuşunda sayısal imza kullanılmıştır. Sayısal imza aynı zamanda bir asimetrik şifreleme uygulamasıdır ve bunun gerçeklemesinde de Windows CryptoAPI kullanılmıştır.

Programın testleri için paket koklayıcı (sniffer), uzaktan erişim programları gibi hazır yazılımlar kullanılmış ve istenilen şekilde paketler yaratmak ve yollamak için bir de test programı yazılmıştır.

4. İTÜ SÖA SENARYOSU

Aşağıdaki basit duruma bakarak İTÜ yerel ağında veri iletişimi ayrıntılarını görebiliriz (Şekil 2):

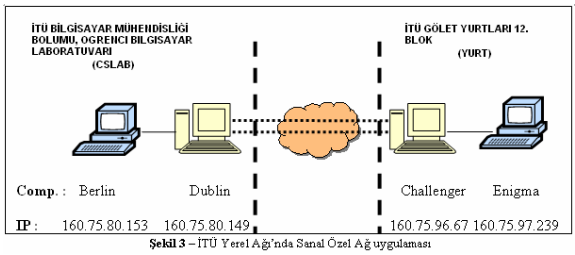


Şekil 2 – İTÜ Yerel Ağı'nda veri iletişimi

CSLAB ve YURT'ta bulunan Berlin ve Enigma adlı iki bilgisayar bağlantı kurup veri iletişimine geçtiklerinde aralarındaki açık ağı kullanırlar ve aralarında pek çok anahtar ve yönlendirici bulunmaktadır. Hiçbir güvenlik önlemi alınmadan yapılan veri iletişimi, ağ üzerindeki bilgisayarlar tarafından izlenebileceğinden ve çeşitli saldırılara karşı korunmasız kalacağından oldukça risklidir.

Güvenlik açısından SÖA oluşturulması örnek durumumuz için iyi bir çözüm olacaktır. Şu halde bilgisayarlar arası bir tünel oluşturulması gereklidir. Bir SÖA'yı "sanal özel" yapan şey tüneldir. Internet trafiği içinde, SÖA tüneli, uçlar arası farklı yollar izleyebilir. SÖA iletimini tünel yapan, sadece uçların tünel içindeki gerçek verileri görebilmesidir.

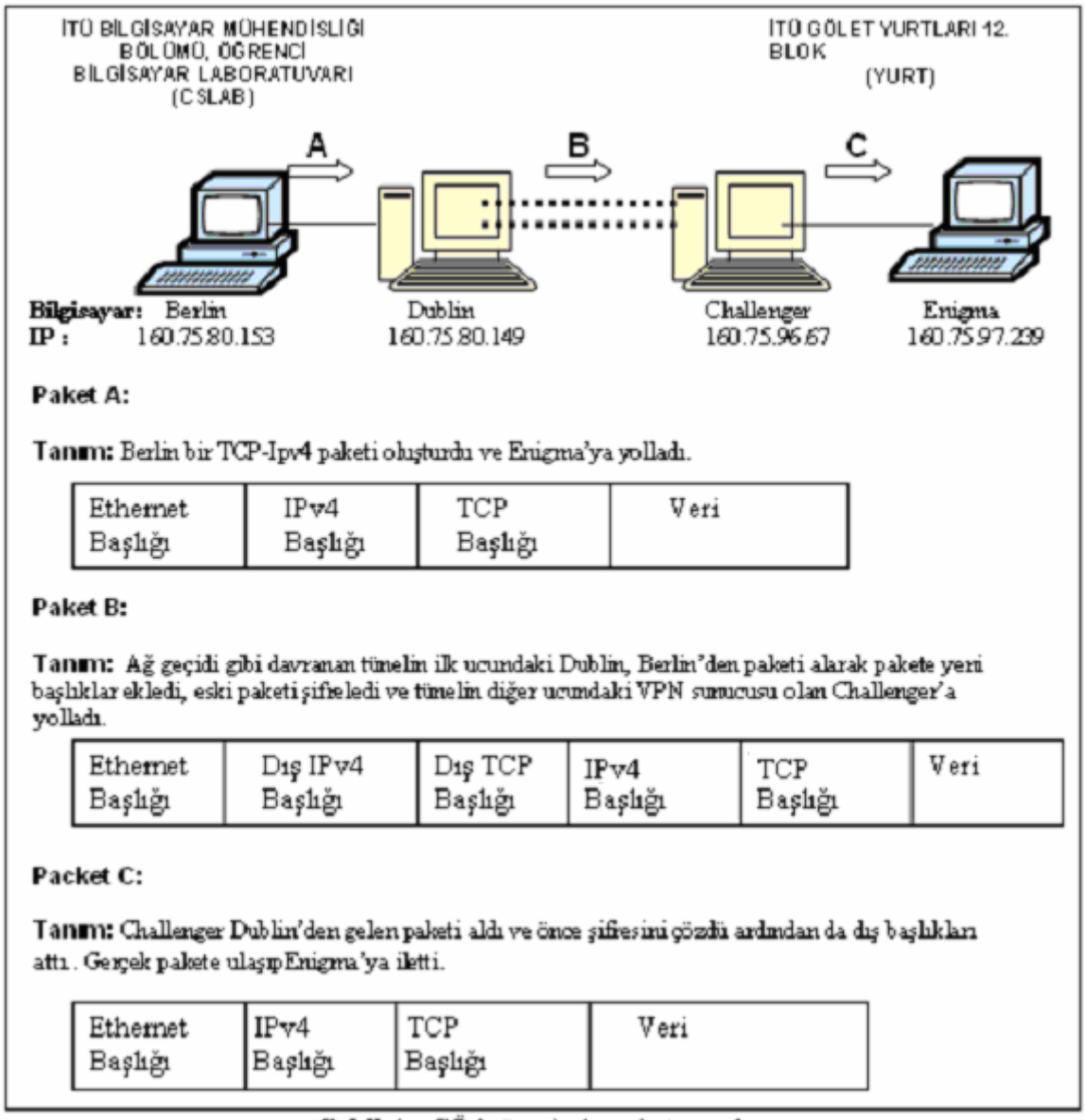
Tünel gerçeklemek için, tünelin iki ucunda birer tane SÖA sunucusu olması gerekir. Şekil 3'te görüldüğü gibi, Dublin ve Challenger bizim örneğimizdeki SÖA sunucularıdır.



Şekil 3 – İTÜ Yerel Ağı'nda Sanal Özel Ağ uygulaması

Dublin CSLAB'ta, Challenger da YURT'ta bulunan farklı bilgisayarlardır. Bu bilgisayarlar, istemci bilgisayarlar olan Berlin ve Enigma'ya dağıtıcılar (hub) üzerinden bağlıdırlar. SÖA sunucu bilgisayarları, paketler üzerinde yoğun işlemler yaptıklarından görece daha güçlü ve hızlı bilgisayarlardır.

Berlin ve Enigma arasında SÖA üzerinden veri iletişimi Şekil 4'te ayrıntılandırılmıştır. Bu örnekte sadece tek yönlü iletişim gösterilmiştir; ancak çift yönlü iletişim de benzer şekilde gerçekleştirilebilir.



Şekil 4 – SÖA üzerinde paket yapıları

5. SONUÇ

Bu çalışmada İTÜ yerel ağı için bir sanal özel ağ tasarımı yapılıp gerçeklenmiştir. SÖA, güvenli olmayan ağda herhangi bir ağ cihazı veya özel bilgisayar kullanmadan, yazılımla gerçeklenmiştir.

SÖA, düşük maliyetli, güvenli ve esnek olması dolayısıyla özel ağlarla rekabet edebilen bir teknolojidir. SÖA, güvensiz bir açık ağı özel ağa dönüştürebilmektedir. SÖA'nın bu başarısının ardında kapsülleme ve şifreleme yatar. Ayrıca, asıllama ve anahtar değişimi gibi konular da SÖA için önem taşır.

Altı aylık bir çalışmanın ürünü olan bu projede Microsoft Windows platformunda CryptoAPI, WinPCap ve PlatformSDK kullanımıştır. Kullanılan ana programlama dili C'dir.

Bu çalışma, daha yüksek başarım ve daha yüksek güvenlik seviyesi sağlamak üzere genişletilebilir ve şu an manuel olarak yapılan biçimlendirme ve kısıtlı olan görsel arayüz kullanıcı yanlısı olacak şekilde iyileştirilebilir. Çalışmanın başarım çözümlemesi ve hız sınamaları halen devam etmektedir.

KAYNAKLAR

[1] Fowler D., Virtual Private Networks, Morgan Kaufmann Publishers, California, 1999

[2] Jones A., Ohlund J., Network Programming for Microsoft Windows, Second Edition, Microsoft Press, 2002

[3] Comer D. E., Computer Networks and Internets with Internet Applications, Third Edition, Prentice Hall, 2001

[4] Örencik B., Çölkesen R., Bilgisayar Haberleşmesi ve Ağ Teknolojileri, Papatya Yayıncılık, 2002
[5] Schneier B., Applied Cryptography, Second Edition, John Wiley & Sons, Inc., 1996
[6] Microsoft Developer Network Library, http://msdn.microsoft.com
[7] Kent S., Atkinson R., RFC 2401: Security Architecture for the Internet Protocol, 1998.
[8] Simpson W., RFC 1853: IP in IP Tunneling, 1995
[9] Perkins C., RFC 2003: IP Encapsulation within IP, 1996
[10] Gleeson B., Lin A., Heinanen J., Armitage G., Malis A., RFC 2764: A Framework for IP Based Virtual Private Networks, 2000
[11] Virtual Private Network Consortium, http://www.vpnc.org
[12] Internet Engineering Task Force (IETF) Drafts, http://www.ietf.org/ID.html