T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

SANAL ÖZEL AĞ (VPN) BAĞLANTI MANTIĞI (VPN TEKNOLOJİSİ) VE TOKEN GÜVENLİĞİNİN PİN KODU İLE ARTTIRILMASI

(Yüksek Lisans Tezi)

Tezi Hazırlayan : **Nurdoğan AYDOĞDU**

İstanbul, 2014

T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

SANAL ÖZEL AĞ (VPN) BAĞLANTI MANTIĞI (VPN TEKNOLOJİSİ) VE TOKEN GÜVENLİĞİNİN PİN KODU İLE ARTTIRILMASI

(Yüksek Lisans Tezi)

Tezi Hazırlayan :

Nurdoğan AYDOĞDU

Öğrenci No:

110820008

Danışman:

Yrd. Doç. Dr. Ediz ŞAYKOL

İstanbul, 2014

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum "Sanal Özel Ağ (VPN) Bağlantı Mantığı (VPN Teknolojisi) ve Token Güvenliğinin Pin Kodu ile Arttırılması" başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım**. 24/ 05/ 2014**

Nurdoğan AYDOĞDU

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi 110820208...no'lu Nurdoğan Aydoğdu..'in 24/06/2014 tarihinde yapılan tez savunma sınavı[1] sonucunda..50 dakika süreyle sunduğu ve savunduğu tezi hakkında[2] oybirliğiyle, KABUL kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
Programı : BİLGİSAYAR MÜHENDİSLİĞİ
Tez Başlığı[3] : Sanal Özel Ağ (VPN) Bağlantı Mantığı (VPN Teknolojisi) ve Token Güvenliğinin Pin Kodu ile Arttırılması

Tez Sınav Jürisi	Öğretim Üyesi	İmza
Danışman	: Yrd. Doç. Dr. Ediz Saykol	
Üye	: Doç. Dr. Göksel Biricik	
Üye	: Doç. Dr. Kazım Sarı	

---

[1] Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayı tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

[2] Tez sınavının tamamlanmasından sonra jüri, tez hakkında "kabul", "düzeltme" veya "red" kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişiği kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişiği kesilir. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

[3] İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

Sanal Özel Ağ (VPN) Bağlantı Mantığı (VPN Teknolojisi) ve Token Güvenliğinin Pin Kodu ile Arttırılması

Tezi Hazırlayan: **Nurdoğan AYDOĞDU**

ÖZET

VPN sistemi kullanım amacı güvenli fakat erişim için kullanılan tokenlerin pekte güvenli olmadığı görülmüştür. Bu nedenle pin kodu ile güvenlik arttırılması gibi yeni yöntemlere ihtiyaç duyulmaktadır.Bu ve buna benzer yöntemler sayesinde son zamanlarda kötü amaçlı atakların ve bilgi sızıntılarını önlemek adına geliştirilmesi gereken önemli bir durum haline gelmiştir.VPN gibi güvenli görülen erişim sistemlerinde eski yöntemlerdeki sabit parola yada token ile üretilen (60 saniye sonra yeni bir kod üretme özelliği) pin kodu girişi algoritmasının kaynak sağlayıcı firmadan çalınması ihtimali sebebiyle buna ek olarak önüne dört haneli sadece kullanıcının oluşturarak bileceği ayrıca pin oluşturulması güvenliği arttırmaya yeterlidir.

Anahtar Kelimeler: VPN**,** Pin.

Virtual Private Network (VPN) Connection Logic (VPN Technology) And Security Token Boosting With The Pin Code

Presented by: **Nurdoğan AYDOĞDU**

ABSTRACT

VPN system used to access the intended use is safe, but many of the tokens have been shown to be not safe.Pin code so as to increase the security for new configurations are needed.Through this and similar methods of attack and malicious recently in order to prevent information leakage has become an important issue that needs to be developed.VPN as a secure access system seen in the old method or the fixed password generated by the token (after 60 seconds a new feature code generation) pin input algorithm source code stolen from suppliers because of the possibility of additionally creating only the user will know before the four digit pin creation also is sufficient to improve safety.

Key Words: VPN**,** Pin

İÇİNDEKİLER

Sayfa No.

ŞEKİLLER LİSTESİ

Sayfa No.

KISALTMALAR

AES : Gelişmiş Şifreleme Standardı.

CAN : Kampüs Alan Ağı.

EBC : Elektronik Kod Kitabı.

IPSEC : Ağ Katmanı Güvenliği Protokolü.

IPV4 : İnternet Protokolü Versiyon Dört.

LAN : Yerel Alan Ağı.

L2TP : Katman İki Tünel Protokolü.

MAN : Şehir Alan Ağları.

PPTP : Noktadan Noktaya Tünel Protokol.

SAN : Depolama Alan Ağı.

SECURID : Güvenlik Kimliği.

TCP / IP : İletim Denetimi Kuralları.

UDP : Kullanıcı Veribloğu Kuralları.

VPN : Sanal Özel Ağ.

XML : Genişletilebilir İşaretleme Dili.

WAN : Geniş Alan Ağları.

1.GİRİŞ

VPN sistemi kullanım amacı güvenli olması ile beraberinde erişim için kullanılan tokenlerin pekte güvenli olmadığı görülmüştür. Mevcut bilinen sistemlerde kullanıcı adı ve token cihazının oluşturduğu kod ile giriş yapılmaktadır. Muhtemel hırsızlık yada farklı amaçtaki üçüncü şahısların bilgisayarda kullanıcı adı zaten log olarak tutulduğu için basit bir şekilde bulunabilir ve cihazda ellerinde olduğu takdirde kolayca kullanıcı adı ve token kodu ile yetkisiz kullanılabilir. Tüm bunlar göz önünde bulundurulur ise çok güvenli gördüğümüz VPN sisteminde erişim esnasında oluşabilecek bu güvenlik açığı farklı bir yöntem kullanarak güvenli hale getirilebilir. Bu yöntemin çalışması zor olmamakla beraber kullanıcı tarafından kafa karışıklığına neden olmayacak ve basit bir mantıkla çalışmaktadır. Token cihazının ürettiği kod ile beraber sisteme ilk girişte kullanıcının kendi belirleyeceği (yalnızca kullanıcının kendi bileceği) dört hane ön pin oluşturarak sonraki denemesinde ve ileride bundan sonraki tüm sistem erişiminde kullanacağı bu basit güvenlik geliştirmesi ile pin artı token kodu (pin + token kodu = güvenli giriş kodu) birleşik yazılarak sisteme erişebilecektir. Bu mantık kullanıcı tarafından oluşturulan ve sadece kullanıcının bileceği pinini token kodu ile bütünleşik yazmaz ise güvenlik açısından erişime izin vermeyecektir. Kullanıcı pinini unuttuğu takdirde sistemin yöneticisi tarafından new pin mode a (yeni pin moduna) alınarak tekrar kullanıcı kendisi sistemden yeni pin belirleyerek erişimine güvenli bir şekilde devam edebilecektir.

2. VPN: VIRTUAL PRIVATE NETWORK (SANAL ÖZEL AĞ)

VPN sisteminden bahsetmeden önce kullanım alanı ağ paylaşımının güvenliği amaçlandığı için bu sistemin özündeki ağ (network) ne demek ve ağ yapısından kısaca bahsetmek gerekirse ;

2.1. Ağ (Network)

Genel olarak haberleşme (communication) sistemi , bilgiyi bir noktadan başka bir noktaya en az hata ile iletme olayıdır. Teknolojinin çok hızlı ilerlemesi ile genel haberleşme sistemi içerisinde bilgisayarlı haberleşme sistemi de yerini aldı. Birden çok bilgisayarın birbirine bağlanması ile oluşturulan çalışma düzenine bilgisayar şebekesi yada diğer adı ile bilgisayar ağı (Computer Network) denir.

Bilgisayar ağı oluşturmanın en önemli nedenleri ;

- Yazılım (dosyalar, uygulama programları vb.) ve donanım (yazıcı, tarayıcı vb.) gibi unsurların paylaşımını sağlamak.
- Kullanıcılar arasında veri iletişimini (mesaj ve dosya gönderme / alma vb.) kolaylaştırmaktır.
- En önemlisi ise verilerin yönetimi ve güvenliğidir.

2.2. Ağlar ölçeklerine göre temelde üç toplamda ise yedi çeşittir

- Kişisel Alan Ağları (Personal Area Network “PAN”) ; Bilgisayar aygıtları arasında iletişimi kurmak için kullanılan ağdır.
- Yerel Alan Ağları (Local Area Network “LAN”) ; Bir grup, bir oda yada kat içerisinde, bir bina yada işletme içerisindeki basit temel yapıdır.
- Şehir Alan Ağları (Metropolitan Area Network “MAN”) ; 50 yada 100 KM’ye kadar bir ölçekteki genelde şehir büyüklüğündeki alan kadar olan ağlardır.
- Geniş Alan Ağları (Wide Area Network “WAN”) ; Birden fazla şehir, ülke hatta dünya ölçeğindeki ağlardır. İnternet dünya çapında

milyonlarca bilgisayar, sunucu ve ağ cihazlarının bağlı olduğu bir WAN'dır.

- Sanal Özel Ağlar (Virtual Private Network "VPN") ; Herkesin erişimine açık olan ortak ağı (yani interneti) kullanıp özel ağa bağlanarak güvenli veri transferi gerçekleştirmeye yarayan bir ağ bağlantı çeşitidir.
- Kampüs Alan Ağı (Controller / Campus Area Network "CAN") ; Belirli bir coğrafi bölge ile yerel ağların ara bağlantısının yapıldığı bilgisayar ağıdır. Örneğin bir kuruluş, devlet, üniversite gibi yerlerde kullanılır.
- Depolama Alan Ağı (Storage Area Netwotk "SAN") ; Genelde büyük ağ kullanıcılarına hizmet vermek üzere veritabanı sunucuları ile birlikte farklı tipteki veri depolama cihazlarını birbirine bağlanmasını özel amaç edinen yüksek hızlı bir ağ bağlantı çeşitirdir.

Yukarıda belirtmiş olduğum ağ çeşitleri toplamda yedi olsada aslında temel olarak üç çeşittir (1- Yerel Alan Ağları, 2- Şehir Alan Ağları, 3- Geniş Alan Ağları).

Ağ protokoller sayesinde katmanları arasında iletişimi sağlar. En önemlisi protokollerin kullanımı veuygulanmasındaki seçim özellikleridir. Ağ yapısındaki en öne çıkan protokol TCP / IP (İletim Denetimi Kuralları = Transmission Control Protocol / İnternet Kuralları = Internet Protokol) olmakla beraber bunun nedeni ise UDP (Kullanıcı Veribloğu Kuralları = User Datagram Protocol) protokolündeki gibi veri karşı tarafa ulaşıp ulaşmadığını kontrol eder. TCP protokolünde güvenilir veri gönderimi tercih edilme sebebi iken UDP protokolü genelde ses ve görüntü iletimi bağlantı kurmadan gönderir.

Ağ iletişiminde TCP / IP yollanacak veriler katmanlara göre paketlenerek gönderilir ve alıcı bu paketleri teker teker açarak veriyi ulaştığını teğit eder. Bu karşılıklı kontrol mekanizması güvenilir veri aktarımını sağlar.

Kablolu yada kablosuz ağ bağlantıları genel olarak herhangi bir aşamasında TCP / IP protokolü ile networkte paketlerini diğer istemciye göndermektedir. Buna

bir örnek olarak VPN sistemindeki tünelleme protokolü işlevi için TCP / IP protokolü kullanılmaktadır. Paketler güvenli bir şifreleme metodu ile tünel oluşturularak karşı istemciye iletilmektedir.

Ağ hakkındaki bu genel ve kısa bilgilendirme sonrası VPN sisteminin aşağıda yer alan detaylarını daha iyi anlayabiliriz.

2.3. VPN: Virtual Private Network (Sanal Özel Ağlar)

Açılımı Virtual Private Network (Sanal Özel Ağ) olan VPN teknolojisi, herkesin erişimine açık olan ortak ağı (yani interneti) kullanıp özel ağa bağlanarak güvenli veri transferi gerçekleştirme imkanı sunmaktadır.

Türkçe'ye Sanal Özel Ağ olarak çevrilen VPN ağları tünelleme protokolü kullanarak transferi gerçekleşen her data paketini şifrelediği için son derece güvenlidir.VPN istemcisi, internet üzerinden bağlantı kurmak istediği kaynakla sanal bir noktadan noktaya bağlantı kurar, kaynak yada uzaktan erişime geçmek istediği sunucu kimlik bilgilerini kontrol eder ve doğruladıktan sonra VPN istemcisiyle uzaktan erişime geçtiği sunucuyla veri akışı gerçekleşir.Veriler VPN ile veri transferi sırasında transferi gerçekleştirecek data paketleri güvenli olmayan ve herkes tarafından kullanılan çalışma ağları üzerinden transfer edilmeden önce şifrelenmektedir. Ayrıca söz konusu transferin gerçekleştiği network ağları da şifrelenmektedir.

Temelde iki tip VPN teknolojisi vardır. Uzaktan Erişim VPN (Remote Access VPN) ve İki Ağ Arasında VPN (Site to Site VPN) olarak adlandırılırlar.

2.3.1 Uzaktan Erişim VPN (Remote Access VPN)

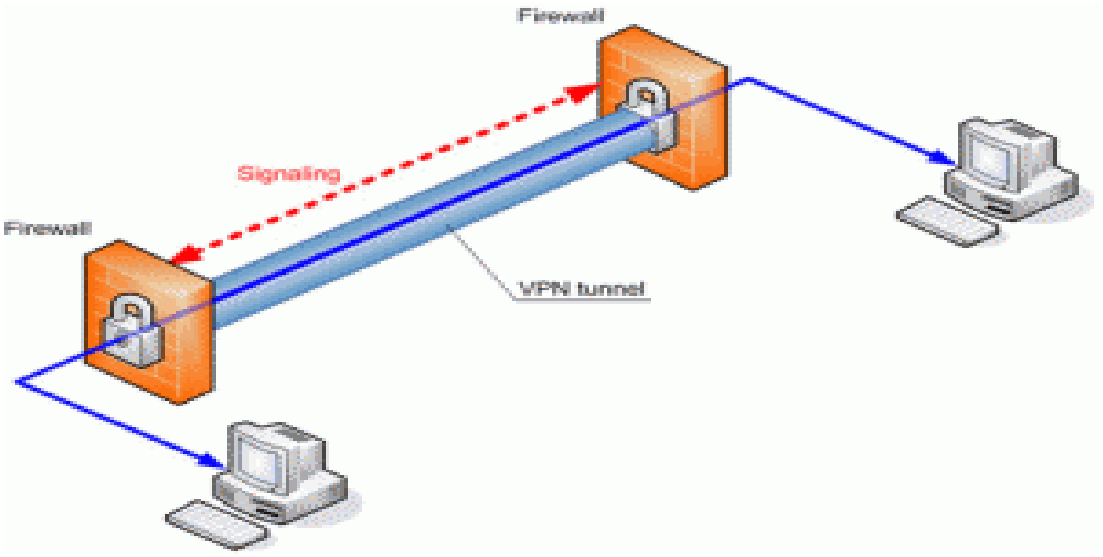
Evinde çalışan yada seyahat esnasında ofiste olmayan kullanıcıların internet üzerinden özel ağ üzerindeki sunucuya erişime imkan sağlar. VPN, istemcisiyle uzaktan erişim sunucusu arasında noktadan noktaya bir bağlantıdır. Ayrıca veriler özel bir ağ üzerinden gönderiliyormuş gibi görünmektedir. Bu yüzden ortak ağın gerçek alt yapısı önemli değildir.

2.3.2 İki Ağ Arasında VPN (Site to Site VPN)

Farklı ofisler arasında veya farklı kuruluşlar arasında ortak bir ağ üzerinden güvenli bir şekilde iletişimi sağlamaz. VPN bağlantısı WAN (Wide Area Network) bağlantısı gibi çalışır. WAN bağlantısı şehirler, ülkeler gibi uzun mesafeler arasında iletişimi sağlayan ağ çeşididir. Ağlar, internet üzerinden verileri bir yönlendirici ile başka bir yönlendiriciye iletir. Yönlendiricilere göre VPN bağlantısı, veri bağlantısı olarak işlev görmektedir.

Siteden Siteye VPN bağlantısı özel bir ağın iki bölümünü birbirine bağlar. VPN sunucusu,bağlı olduğu ağa bağlantı sunarken,yanıtlayan diğer sunucu yada yönlendirici (VPN sunucusu), yanıtlayan yönlendiricinin (VPN istemcisi) kimlik bilgilerini doğrular ve karşılıklı doğrulama sağlanır. Ayrıca siteden siteye VPN bağlantısı üzerindeki iki sunucuda gönderdikleri veri transferlerinin başlangıç noktaları tipik olarak yönlendiriciler veya sunucular değillerdir.

Aşağıdaki şekilde bir VPN türü olan Site to Site örneğini görebiliriz:

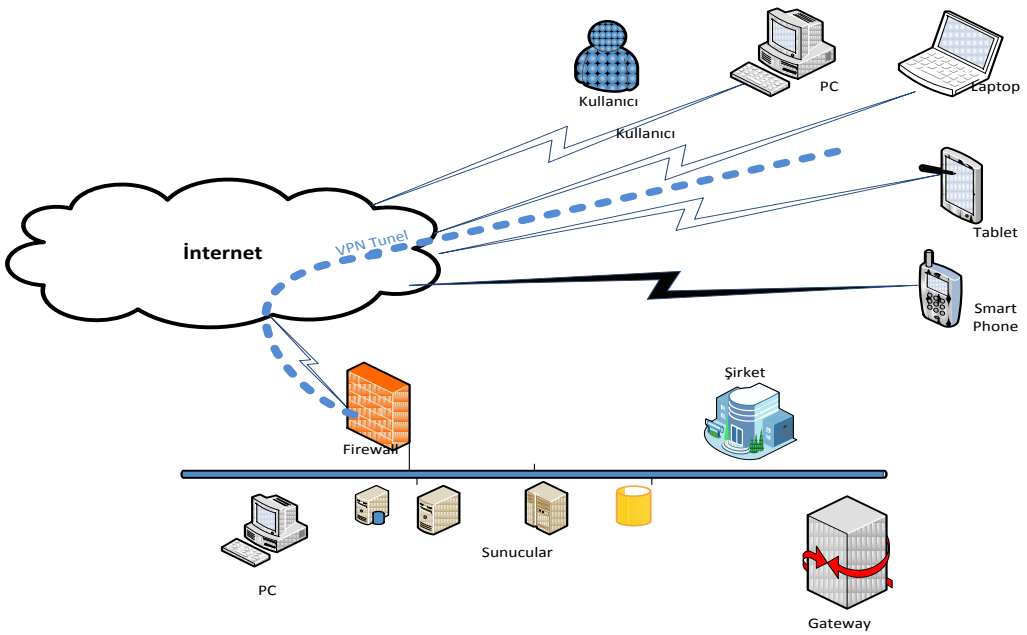


Şekil.1. Site to Site VPN

VPN ağları kullanım alanlarına göre Access VPN, Intranet Tabanlı VPN ve Extranet (internet) Tabanlı VPN olmak üzere üç çeşit olup Access VPN gerçek kişiler tarafından bireysel kullanım amacıyla tercih edilirken İntranet ve Extranet Tabanlı VPN ağlar tüzel kişiler (Şirketler, Üniversiteler gibi kurum ve kuruluşlar) tarafından tercih edilmektedir.

VPN'in kullanım alanlarını daha net algılayabilmek için örneklendirmek gerekirse ;

- Evden ofis bilgisayarına bağlanarak çalışabilir, proje ve dosyalara göz atabilir, sanki fiziksel olarak ofisteymiş gibi çalışma imkanı sunmaktadır.
- Bir firmanın şubeleri ile arasındaki iletişim ve denetimini sürdürebilmesi için farklı yerlerde ki şubelerin yüksek güvenlik protokolü altında firmaya bağlanmasını sağlamaktır.
- Hiç bir kısıtlama olmaksızın şirketin intraneti üzerindeki tüm içeriğe ulaşma imkanı sağlamaktadır.



Şekil.2. VPN Kullanım Alanları

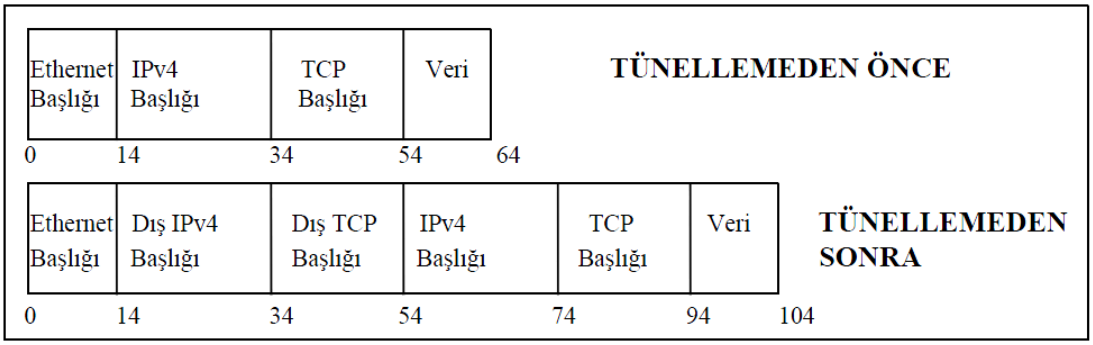
Yukarıdaki şekilde görüldüğü üzere VPN'in kullanım alanları ve çeşitliliğini kapsayan bir örnektir.

VPN bağlantılarının özellikleri kapsülleme, kimlik doğrulama ve son olarak ise veri şifrelemedir;

- **Kapsülleme(Tünelleme):** VPN ağında veriler bir üstbilgi kapsüllenirler. Bu üst bilgi, verileri geçiş ağı sırasında çapraz geçmelerine izin verecek bilgileri içerir. 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.

  Kapsülleme işlemini ileride tam olarak bahsedeceğim VPN tünel protokolleri bölümünde daha iyi anlaşılacaktır. Aşağıda yer alan tünelleme hakkındaki kısa bilgi ve görsel şekilden ön bilgi olarak bahsetmek gerekirse ;

  Tünelleme sunucu uygulamasında gerçekleşecek şekilde tasarlanmıştır. Gelen paketler yeni bir TCP – IP peketinin içine gömülür ve alıcı sunucusuna gönderilir. Kapsülleme işleminde yeni peketin eskiden iki başlık boyu kadar büyük olacağı açıktır. Yeni paketin ethernet başlığında kaynak sunucusunun ve ağ geçidinin MAC adresi bulunur. Ipv4 başlığında ise yine sunucuların IP adresleri bulunur ve toplam sınaması değeri tekrar hesaplanır. TCP başlığınında sunucularının senkronize çelışabilecekleri şekilde biçimlendirilmesi gerekir. Orjinal paket ise Ipv4 ve TCP başlıklarıyla beraber yeni paketin veri kısmını oluşturur.



Şekil.3. Kapsülleme (Tünelleme)

Virtual Private Network, aynı özel ağda bulunmayan bir veya birden fazla ağ cihazı arasından güvenli bir şifreleme metodu kullanarak kapsüllenmiş veri akışı yapar. Bu kapsülleme özelliği aslında tünelleme olarakta bilinmektedir.

Güvenli şifreleme metodunun amacı, verilerin özel ya da kamusal alandaki diğer ağ cihazlarından gizlenmesidir.

VPN'nin asıl amacı kullanıcı internet akışını başka bir ağa aktararak kullanıcı kimliğini gizlemektedir. Bu sayede kullanıcıların kimlikleri bilinmeksizin İnternet'te gezinmeleri sağlanmaktadır. Birçok işletim sistemi VPN'e bağlanma desteği vermektedirler.

Firmalar tarafından yaygın olarak tercih edilen VPN, yöneticilerin, uzak ofislerin, bayi, acenta satış temsilcilerinin güvenli bir şekilde özel ağlara bağlanmalarını sağlar. Örnek olarak bilgisayarı veya akıllı telefonları (I-phone , Android bazlı telefonlar vb.) bir VPN'e bağladığında bilgisayar, bağladıkları VPN'in herhangi bir bilgisayarı gibi davranır. Tüm ağ trafiği güvenli bir bağlantı üzerinden VPN'e iletilir. Çünkü bilgisayarla VPN'in bağlandığı tüm kaynaklara erişebilmeyi sağlanmıştır.

Örneğin ; bir Web sitesine bağlanmak gerektiğinde bilgisayarın ilk olarak güvenli bir şekilde VPN'e bağlanması yapılır. Ardından VPN kaynaklarını kullanarak ulaşmak istenen Web sitesine güvenli bir şekilde bağlanmaya yardımcı olur. Mesela Almanya'daki VPN ile

İnternet'e bağlanmak denendiğine IP adresi dâhil tüm bağlantı özellikleri Almanya'daki ağla entegre olur.

- **Kimlik doğrulama:** Ağa erişmeye çalışan kişinin buna yetkili olup olmadığı, dışarıdan müdahele edilemeyecek şekilde, yani şifreli olarak HTTPS protokolü ile yapılır ve izini olanlar ağa alınır.

  Üç farklı kimlik doğrulama metodu mevcuttur. İlk olarak kullanıcı düzeyinde PPP kimlik doğrulamasında VPN bağlantısında bağlanan VPN sunucusu, VPN istemcisinin kimliğini Noktadan Noktaya (PPP) kullanıcı düzeyinde kimlik doğrulama metodlarıyla doğrular. Sonra VPN istemcisinin yetkili kişi olmasını onaylar. Eğer, karşılıklı kimlik sorgusu yapılırsa VPN istemcisi de VPN sunucusunun kimliğini doğrular. Bu sayede bilgisayarlara karşı da bir koruma sağlanmış olur. İkinci olarak İnternet anahtar değişimi (IKE) ile bilgisayar düzeyinde kimlik sorgulama işlemi, VPN istemcisi ve VPN sunucusu İnternet Protokolü güvenliği (Ipsec) ile güvenlik oluşturmak üzere bilgisayar sertifikası veya önceden paylaşılan bir anahtar değişimi için IKE protokolünden yararlanırlar. Bu iki durumda da VPN istemcisi ve VPN sunucusu, karşılıklı olarak bilgisayar düzeyinde kimliklerini doğrular. Bilgisayar düzeyinde kimlik doğrulama, kullanıcı düzeyinde kimlik doğrulamaya oranla daha güvenlidir. Bu yüzden bilgisayar düzeyinde kimlik doğrulama daha çok önerilir. Bilgisayar düzeyinde kimlik doğrulama, sadece L2TP/IPsec bağlantıları için kullanılır. Üçüncü olarak veri kaynağı için kimlik doğrulama ve veri bütünlüğü yer almaktadır. VPN bağlantısında veri gönderimi sırasında gönderilen verilerin bağlantının diğer ucundan gönderildiğini ve verilerin gönderim aşaması sırasında değişmediğini sağlamak için veride yalnızca gönderenin ve o alanın bildiği bir şifreleme anahtarına uygun bir şifreleme toplamı vardır. Veri kaynağı için kimlik doğrulama ve veri bütünlüğü, sadece L2TP/IPsec bağlantılarında kullanılır.

- **Veri şifreleme:** Verilerin dışarıdan ağdan geçen bilgileri dinleyenlerin çözümleyemeyeceği biçimde şifrelenerek dışarıdakiler için anlaşılmaz hâle getirilmesidir.

  Veriler, paylaşılan veya ortak geçiş ağından çapraz geçer ve gizliliğin sağlanması için gönderici tarafından şifrelenir. Daha sonra bu şifreler, alan tarafından çözümlenir. Şifreleme ve alan tarafından şifre çözme işlemi, gönderecinin ve alanın kullandığı şifreleme anahtarına bağlıdır.

  VPN bağlantısı üzerinden veri transferi yapıldığında şifreleme anahtarı olduğu için bilgiler ele geçirilse bile dışarıdaki için bir anlam ifade etmez. Ayrıca şifreleme anahtarının uzunluğu da güvenlik anlamında önemli bir faktördür. Çünkü şifreleme anahtarını belirlemek adına hesaplama işlemleri yapıldığında bu tür metodlar, şifreleme anahtarı büyüdüğünde daha uzun hesaplama zamanı alır. Bu yüzden güvenliği sağlamak adına, yani veriyi mümkün olduğu kadar gizli tutmak amacıyla şifreleme anahtarını mümkün olan en büyük boyutta seçmek veya kullanmak önemli bir faktördür.

  Verilerin hack edilme durumunu önlemek için şifrelenmesi ağ üzerinde bir yavaşlığa neden olabilir.Genel olarak 128 bit şifreleme yöntemi yaygın olarak kullanılmaktadır.Diğer 256 – 512 yada 1028 bit şifreleme yöntemleri daha güvenli olsada aşırı yavaşlığa neden olabilir ve uygulamalar daha yavaş çalışabilir.

  Şifrelemede yazılım ve donanım olarak iki yöntem kullanılır.Yazılımsal yöntem donanımsal (Concentrator Machine) yöntemden daha yavaş çalışmaktadır.Donanımsal yöntem maliyeti daha fazla olsada şifreyi çözme metodunda chip kullanarak giriş alanındaki şifreli paketlerin çıkışta işlenip paketler şifresiz gönderilir.Yazılımsal yöntemde sistem üzerinde mevcut data trafiğine ek olarak paketler tekrar işlenerek şifreler çözüldüğü için yavaş çalışmaktadır.

2.4. VPN (Virtual Private Network) Tünel Protokolleri

İlk olarak tünel oluşturma, bir protokol türündeki paketin başka bir protokol türünde kapsüllenmesini sağlar. Örnek olarak VPN, IP paketlerini ortak bir ağ üzerinden kapsüllemek için PPTP protokolünü kullanır. Noktadan Noktaya Tünel Protokolü (Point to Point Tunnel Protocol "PPTP"), Katman İki Tünel Protokolü (Layer Two Tunnel Protocol "L2TP") veya Güvenli Yuva Tünel Protokolü (Secure Socket Tunneling Protocol "SSTP") gibi protokoller, bir VPN çözümüdür. PPTP, L2TP ve SSTP protokolleri, Noktadan Noktaya Protokolü (PPP) için belirlenen özellilkleri esas alır. PPP, çevirmeli veya noktadan noktaya bağlantılar üzerinden veri transferi yapmak için yapılmıştır. IP kullanımı için PPP, IP paketlerini PPP içinde kapsüller. Kapsüllenen PPP IP paketleri noktadan noktaya bağlantılar üzerinden aktarır. PPP, çevirmeli istemci ve ağ erişimi sunucusu arasında kullanılan protokoldür.

2.4.1 PPTP Protokolü (Noktadan Noktaya Tünel Protokolü = Point to Point Tunnel Protocol)

PPTP, birden çok protokolün şifrelenmesi ve IP ağı veya ortak IP üzerinden gönderilen verilerin IP üstbilgisi ile kapsüllenme işleminin yapılmasını sağlamaktadır. PPTP, uzaktan erişim veya siteden siteye VPN bağlantıları için kullanılmaktadır. PPTP, etkin bir VPN sunucusudur. Çünkü ortak ağ üzerinden olan İnternet ve intranet arasında bulunan etkin bir sunucudur.

PPTP, ağ üzeri aktarım yaptığında PPP çerçevelerini IP datagramları içinde kapsüller. PPTP, tünel yönetimi için Genel Yönlendirme Kapsüllemesi'nin (GRE) değiştirlmiş bir sürümünü kullanır. Ayrıca kapsüllenen PPP çerçeveleri şifrelenebilir veya sıkıştırılabilir.

PPP çerçevesi, MS-CHAPv2 veya EAP-TLS gibi şifreleme anahtarlarıyla Microsoft Noktadan Noktaya Şifreleme (MPPE) ile şifrelenir. PPP çerçeve yüklerinin şifrelenebilmesi için MS-CHAPv2 veya EAP-TLS gibi kimlik doğrulama protokollerinin kullanılması gerekmektedir. PPTP, önceden şifrelenen ve PPP şifrelenmesinin kapsüllenmesinden faydalanılır.

2.4.2 L2TP Protokolü (Katman İki Tünel Protokolü = Layer Two Tunnel Protocol)

L2TP, birden çok protokol trafiğinin şifrelenmesi ve sonrada IP gibi noktadan noktaya veri transferi teslimini destekleyen herhangi bir medya üzerinden iletilmesini sağlar. L2TP, Cisco Sytems Inc. tarafından geliştirilmiştir. Ayrıca PPTP ve Katman İki İletme (L2F) protokollerinin birleşiminden oluşmaktadır. L2TP, PPTP ve L2F'nin en donanımlı özelliklerine sahiptir.

Microsoft'un L2TP uygulaması PPTP gibi davranmaz. PPP, veri iletiminin (datagramların) şifrelenmesinde MPE'yi kullanmaz. L2TP, şifreleme esnasında Aktarım Modunda İnternet Prtokolü (Ipsec) güvenliğini kullanır. L2TP ve IPsec'in birleşimine L2TP/Ipsec denilmektedir. Ek olarak L2TP, TCP/IP protokolüyle birlikte yüklenir.

L2TP/IPsec paketlerinin kapsüllenmesi iki katmandan oluşmaktadır. Birinci katman, PPP çerçevesi L2TP ve UDP üstbilgisiyle sarılmaktadır. İkinci katman ise Ipsec güvenlik yükü dediğimiz (ESP) üstbilgi ve altbilgi olmak üzere iletiyi ve kimlik doğrulamayı destekleyen Ipsec kimlik doğrulama altbilgisi ve IP üstbilgisiyle sarılır. IP üstbilgisinde VPN sunucusuna karşılık gelen IP adresi yer almaktadır.

2.4.3 SSTP Protokolü (Güvenli Yuva Tüneli Protokolü = Secure Socket Tunneling Protocol)

SSTP (Güvenli Yuva Tüneli Protokolü), TCP bağlantısı üzerinden (443 numaralı) HTTPS protokolünü kullanan yeni bir tünel protokolüdür. Bu protokol, trafiğin güvenlik alanından PPTP ve L2TPsec trafiğini engelleyen We proxylerden geçimesine yardımcı olmaktadır. Ayrıca SSTP ve PPP trafiğini HTTP protokolünün SSL (Güvenli Yuva Protokolü) üzerinde kapsülleme işlemi yapmak için bir işleyiş sağlar. PPP'de aynı EAP-TLS gibi kimlik doğrulama metodlarını sağlar. SSL ise gelişmiş anahtar antlaşması, şifreleme ve bütünlük denetimi kullanarak transfer ânında güvenliği sağlar. Eğer istemci SSTP tabanlı VPN bağlantısı oluşturmak isterse ilk başta SSTP, SSTP sunucusunda çift yönlü bir HTTPS katmanı oluşturur. Paketler de HTTPS katmanı üzerinden veri yükü olarak ilerler. Kapsülleme

işleminde SSTP, PPP çerçevelerini IP datagramları içinde kapsüller. Ayrıca SSTP ve PPP veri çerçevelerinin aksine tünel yönetimi için TCP bağlantısı (443 numaralı bağlantı noktası) kullanır. Şifrelemede ise HTTPS protokolü, SSL ile SSTP iletisini şifreler.

Tünel Protokolleri arasında seçim yapılabilirken PPTP, L2TP/Ipsec ve SSTP ile VPN çözümlerinde dikkat edilmesi gereken hususlar:

- PPTP, Microsoft işletim sistemleriyle birlikte (Microsoft 2000, WindowsXP, Windows Vista vb.) ve aynı zamanda farklı Microsoft istemcileriyle kullanılabilir. PPTP, L2TP/Ipsec gibi ortak anahtar altyapısı (PKI) kullanımını gerektirmez. Bunun üzerine PPTP, veri akışındaki güvenliği sağlar, yani veri bir başkasının eline geçtiğinde şifreleme anahtarı olmadan ulaşılamaz. Fakat veriler, PPTP tabanlı VPN bağlantılarının akış sırasıda verilerin değişimi konusunda ve kimlik doğrulama, yani veriyi gönderen kişinin yetkili kullanıcı olup olmadığı konularında kanıt sağlayamaz.
- L2TP/Ipsec, sadece Microsoft 2000, Windows XP ve Windows Vista ve üstü çalıştıran bilgisayarlarla kullanılabilir. L2TP/Ipsec, kimlik doğrulama yöntemi için önceden paylaşılan anahtarları veya bilgisayar sertifikalarını sağlar. Kimlik doğrulama kısımında bilgisayar sertifikalarını sağlamak için PKI'ya başvurur. L2TP/Ipsec, VPN bağlantıları kimlik doğrulama, veri gizliliği ve veri bütünlüğü için IPsec'i kullanır.
- SSTP, Windows Vista Service Pack 1 (SP1) veya Windows Server 2008 ve üstü işletim sistemleriyle çalışan bilgisayarlarda kullanılır. SSTP VPN bağlantılarında SSL kullanılarak veri gizliliği, kimlik doğrulaması ve veri bütünlüğü sağlanır. Ayrıca, bu üç tünel türünde de kimlik doğrulama, İnternet Protokolü sürüm 4 (Ipv4) ve İnternet Protokolü 6 (Ipv6) antlaşması ve ağ erişim koruması (NAP) gibi PPP özellikleri, bu üç tünelde de değişmez bir kıstastır.

2.5. VPN (Virtual Private Network) Çalışma Prensibi ve Güvenlik Amacı

Sanal Özel Ağ (VPN), komşu ağlar arasında gizli ve özel bir bilgi akışını sağlamaya yönelik kurulur. Paketler internet üzerinden gitse dahi, tünelleme ve kullanılan güvenlik yazılımları sayesinde ağ dinlense bile şifrelenmiş paketlere saldıran kişiye elde ettiği bilgiler hiçbir anlam ifade etmeyecektir.

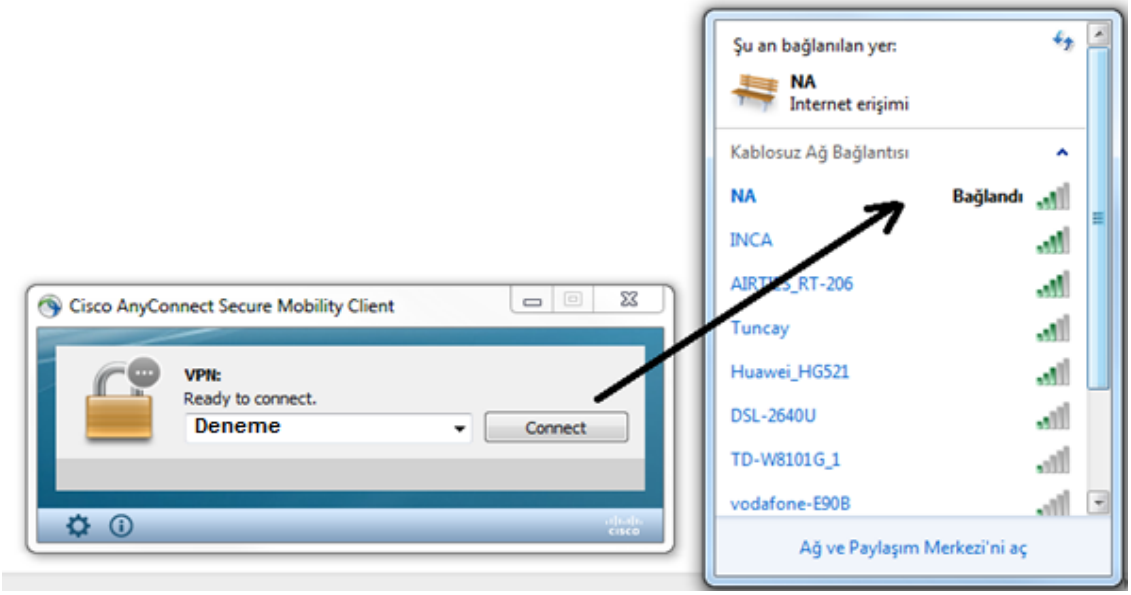
Üçüncü şahısların özel ağlara bağlanmasını engellemek için PPTP (Point to Point Tunelling Protocol = Noktadan Noktaya Tünel Protokolü), L2TP (Layer Two Tunel Protocol = Katman İki Tünel Protokolü), IPSEC (Ağ Katmanı Güvenliği Protokolü) gibi güvenlik protokolleri kullanılmaktadır. Çok düşük bir ihtimal olmakla birlikte söz konusu transferi sağlanan data paketleri 3. Şahıslar tarafından ele geçirilse bile şifrelendiği için içeriğin görüntülenmesi ve kullanılması mümkün değildir.

Tüm bu güvenlik özelliklerinin yanı sıra büyük şirketlerin kendi bünyesi içerisindeki uzak network iletişiminde internetten faydalanmaksızın, özel ağ kurmalarının çoğu yerde gereksiz olduğu düşünülebilir.

VPN sunucuları iki yerel alan ağ arasında ve internet üzerinde gerçekleşen veri iletişimini açık yada kapalı bir algoritma kullanarak şifreler. Aynı biçimde VPN istemcileri ile internet üzerinde bir noktadan yerel alan ağına güvenli bağlantı oluşturmak mümkündür. Şifreleme, internette dolaşan veri paketlerinin içeriğinin üçüncü şahıslar (kötü niyetli şahıslar) tarafından anlaşılmasını engellediğinden, sanal bir tünel işlevi görür.VPN kullanımındaki bahsettiğimiz bu tünel işlevi gönderilen verilerin üçüncü şahıslar tarafından ele geçmesini engelleyerek güvenlik amacını sağlamaktadır.

Şuan bilinen en güvenilir VPN erişim sistemi Cisco AnyConnect Secure Mobility Client uygulamasıdır.Sistem devreye girdiğinde yani token ile gerekli sisteme erişildiğinde ekli şekilde görüldüğü üzere bilgisayar, laptop yada tabletinizin windows (işletim sistemi) wireless sağlayıcısı kullanılmaz konuma geçerek tüm internet çıkışını bu uygulama denetlemektedir.Bilgisayarınızda internet sayfasından (google chrome, mozilla firefox, yandex gibi internet explorer görevindeki

arayüzlerden) herhangi bir siteye erişemeyecek, herhangi bir atak yada saldırıya mağal vermiyecek şekilde VPN erişimi sağlanmaktadır.



Şekil.4. Cisco AnyConnect Secure Mobility Client Uygulaması

2.6. VPN (Virtual Private Network) Kullanım Alanları

Büyük yada Küçük Orta Boy İşletmelerin (KOBİ) kendilerine ait bilişim paketleri ile satış esnasında çalışanların her aşamada birbirine koordineli, hızlı ve doğru karar verebilen daha verimli bir yapıya sahip olması gerekmektedir.Bu verimliliğin artması için sahada yada şubelerinde VPN uygulanması kullanılması sayesinde hem verimli hemde kaynak yönetimi bütçesi açısından tasarruf sağlayarak güvenli bir şekilde işlerini süredürebilir.

Ayrıca Büyük yada Küçük Orta Boy İşletmelerin (KOBİ) yeni satış şubeleri açmak ve büyümek istediğinde Sanal Özel Ağ(Virtual Private Network) olan VPN erişimi sayesinde güvenli internet bağlantısı kurarak merkez şube ve satış şubesi arasında bilgi paylaşımını kolaylaştırabilir.Bu sistem kullanılmadan işlemlerin yapılabilmesi için masrafları arttıracak server, birçok cihaz ve altyapı gereksinimleri

oluşacaktır.Anlaşıldığı üzere VPN bağlantısı kullanımı ucuz, güvenli, kurulum ve sistemlere uygulanabilirliği hızlıdır.

VPN'in kullanım alanlarına en güncel örneğini vermek gerekir ise yakın zamanda bazı internet sitelerinin kullanımının yasaklanması sonrası alternatif olarak bağlanma akla geldiğinde DNS ayarları değişikliğine ek olarak VPN ihtiyacının oluştuğu görülmektedir.Buda yeni trend olarak VPN'in iş ihtiyacı dışında normal kullanıcılar tarafından yaygın kullanımını sağlamıştır.Birçok kişi önceden VPN ne demek olduğunu bilmez iken ihtiyacı doğrultusunda araştırarak şahsi bilgisayarlarına yada kullandıkları akıllı cep telefonlarına VPN uygulamasını yükleyerek etkin bir şekilde kullanmaya başlamıştır.

VPN ile ilgili ifade ettiğimiz bu bilgilerden sonra Token kullanımı ile VPN erişim nasıl yapılacağına dair bilgiye bir sonraki bölümde detaylı olarak ekran görüntüleriyle birlikte görebileceğiz.

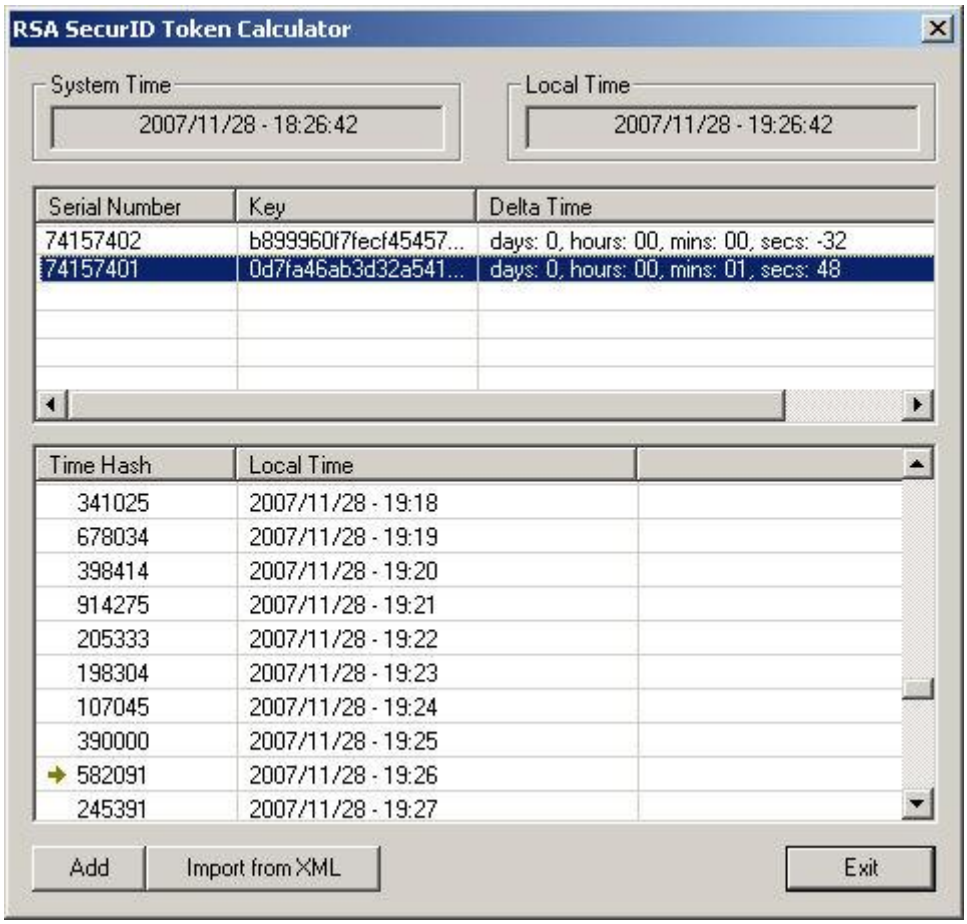
3. TOKEN KULLANILARAK VPN (VIRTUAL PRIVATE NETWORK) ERİŞİMİ

3.1. Token (Jeton,Giriş Anahtarı)

Token kelime anlamı olarak jeton yada giriş anahtarı şeklinde ifade edilmektedir.Token cihazı içerisindeki gömülü işletim sistemi tarafından üretilen kodları cihazın ekranına vererek kullanıcının sisteme giriş yapabilmesi için kimlik doğrulama işlevini gerçekleştirmeyi sağlar.

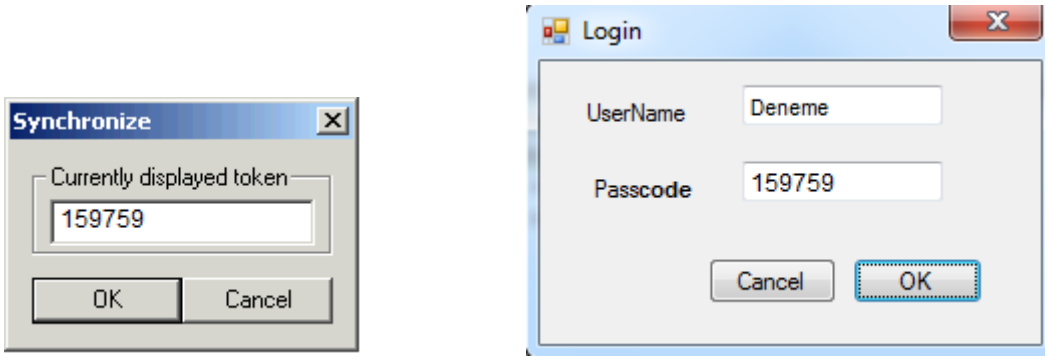
Şekil.5. Token (Jeton,Giriş Anahtarı)

Token cihazı yani şekilde görülen SecurID Anahtarlık her 60 saniyede bir değişen rastgele oluşturulmuş erişim kodlarını görüntüler.Bu kod Time Hash (karmaşık zaman) alanında görüldüğü üzere 60 saniyede bir alakasız rastsal sayılardan oluşmaktadır.Zaman aralığına göre cihazın sistemindeki saat bilgisi ile token kodunu deneyen kişinin bulunduğu yerel saat aynı olmayabilir.Bunun doğruluğunu cihazın algoritmasına göre tarih ve dakika bazında sıralanan kodun karşılık geldiği alan ile o an denenen bölgenin yerel saati ne ise birbirini eşleştirerek iki ayrı saat kavramı arasındaki delta süresi göz önüne alınarak zamanı eşitler.



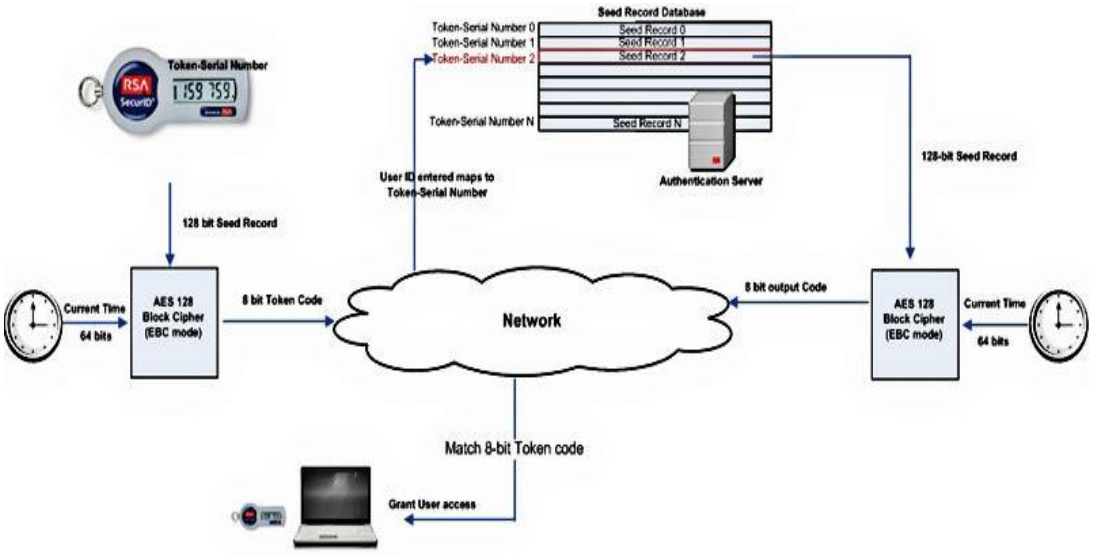
Şekil.6. RSA (Remote System Admin) SecurID Token Calculator

Bu bilgiler XML dosyaları ile tutulmaktadır. SecurID Anahtarlık iki faktörlü kimlik doğrulaması sağlar.Kullanıcı SecurID belirteci üzerinde görüntülenen geçerli kod (Token üzerindeki geçici kod) tarafından sistemde gizli ve sadece kullanıcının bildiği kişisel kimlik numarası (user name) girerek iki faktörlü kimlik doğrulamayı sağlamış olmaktadır.İki faktörlü kimlik doğrulamada birinci faktör Token üzerindeki geçici kod belirteci sistemde kullanıcı kimliğini doğrulamak için login ekranı ile girişi yapılarak eşleştirilebilir.



Şekil.7. SecurID Anahtarlık İki Faktörlü Kimlik Doğrulaması

Token algoritmasını anladıktan sonra algoritmasının akış diyagramına bakarak sistemin nasıl çalıştığını daha net görebiliriz.



Şekil.8. Token Algoritması Akış Diyagramı

128 bit değerindeki tokendan girilen kod AES (Advanced Encryption Standard) 128 Block Cipher (blok şifreleme) EBC Mode (Electronic Book Code Mode) sisteminden 8 bit olarak değiştirilerek token seri numarası ve kullanıcı kimliklerinin olduğu doğrulama sunucusuna (authentication server) giderek burada doğrulanma işlemi yapılır.Doğrulama işlemi başarı ile gerçekleştikten sonra kod 128

bit olarak tekrar AES (Advanced Encryption Standard) 128 Block Cipher (blok şifreleme) ECB Mode (Electronic Code Book Mode) sisteminden geçerek 8 bit olarak değiştirilip kullanıcı erişim izni sağlanmış olarak sisteme giriş yapılabilir.

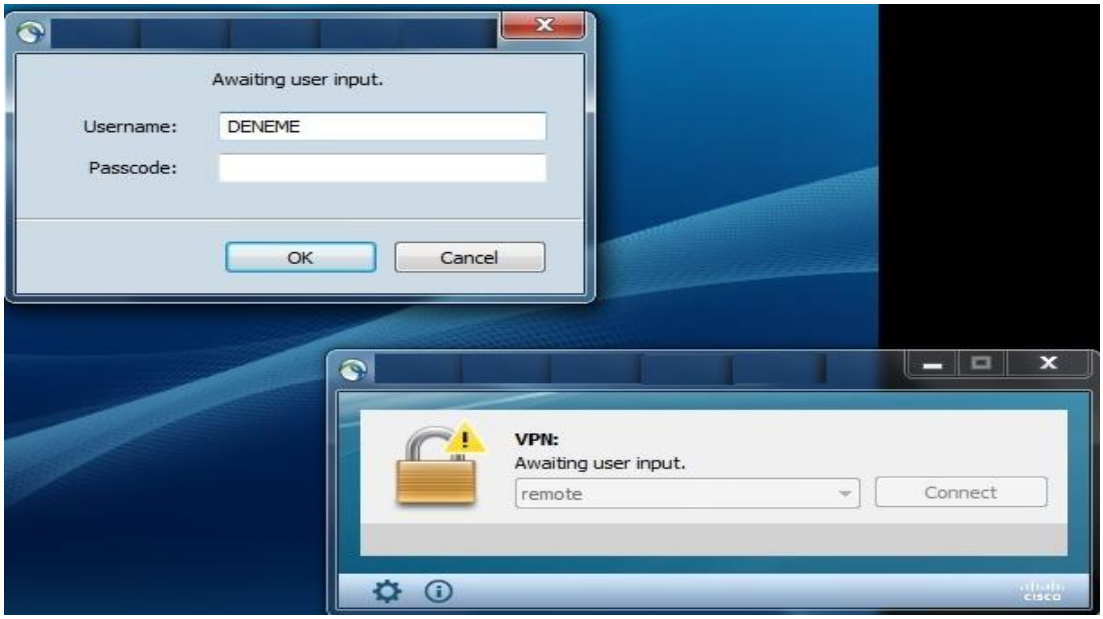
3.2. Token Kullanılarak VPN (Virtual Private Network) Erişimi

VPN sistemine erişimin ilk bilindiği metodu sistem yöneticisi tarafından tanımlanan kullanıcının sabit belirlenen şifresi ile her defasında aynı şifreyi kullanarak sisteme giriş yapmasıdır.Bu yöntem pekte güvenli olmamakla beraber kolay bir şekilde şifre ele geçirilebilir.Bu yöntemde belirli periyotlarla (kullanıcıyı ayda bir yada üç ayda bir şifre değişimine zorlayarak) şifre değişimi yapılarak güvenlik sağlanması amaçlansada pek faydası olmadığı görülmüştür.

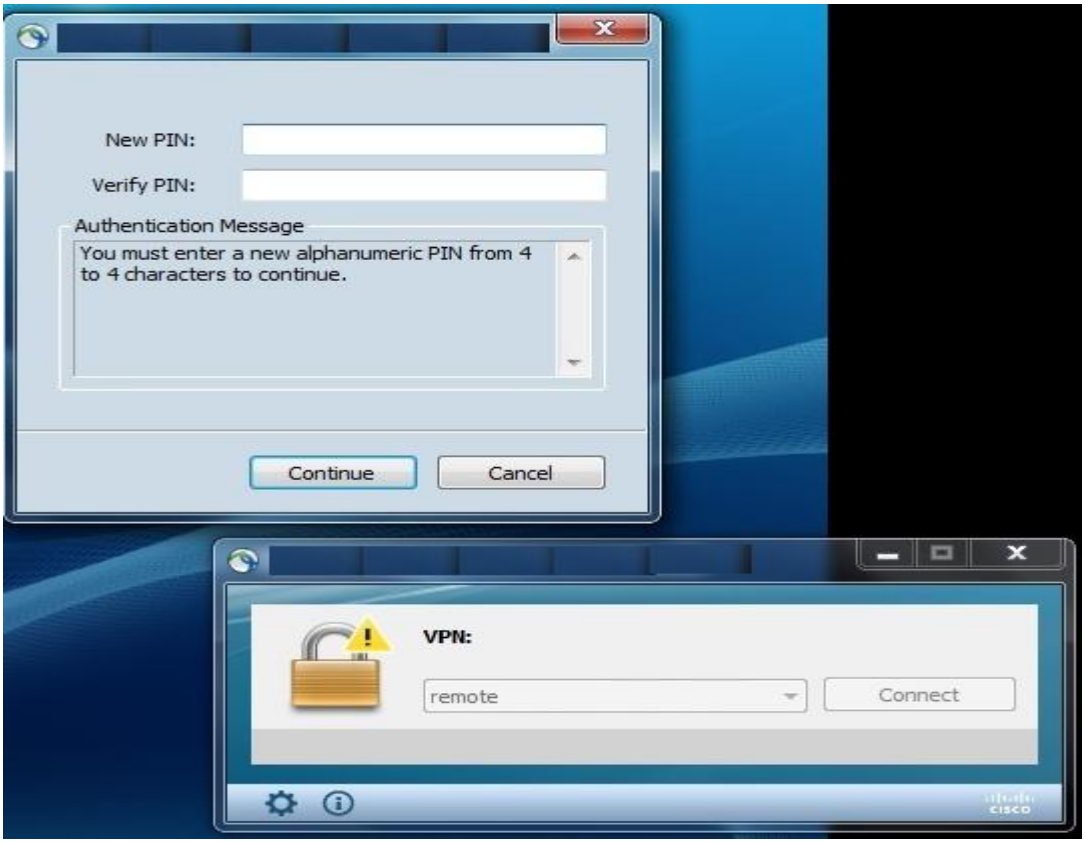
VPN erişim yöntemlerinde birçok farklı metod kullanılması ile beraber en güvenilir olarak nitelendirilen Token ile giriş sağlanmasıdır.Token cihazında ve yönetildiği sistem içerisindeki belirlenen algoritma sayesinde zaman ayarlı sistem tarafından bilinen ve cihaz ekranında aynı anda senkronize çalışarak görülebilen kodun belirli periyotlar ile değişerek sabit kod ile girişi daha güvenilir hale getirilmiştir.Cihazın elde olmayan sebeplerden ötürü kaybolması yada çalınması ile beraber üçüncü kişilerin eline geçerek kullanımında bu yöntemde güvenilirliğin tam olmadığını göstermektedir.

Tezimde savunduğum ve güvenliğinin çift kontrol sayesinde arttırılmış olduğu görülen yöntemin erişimi şu adımlarla gerçekleşmektedir;

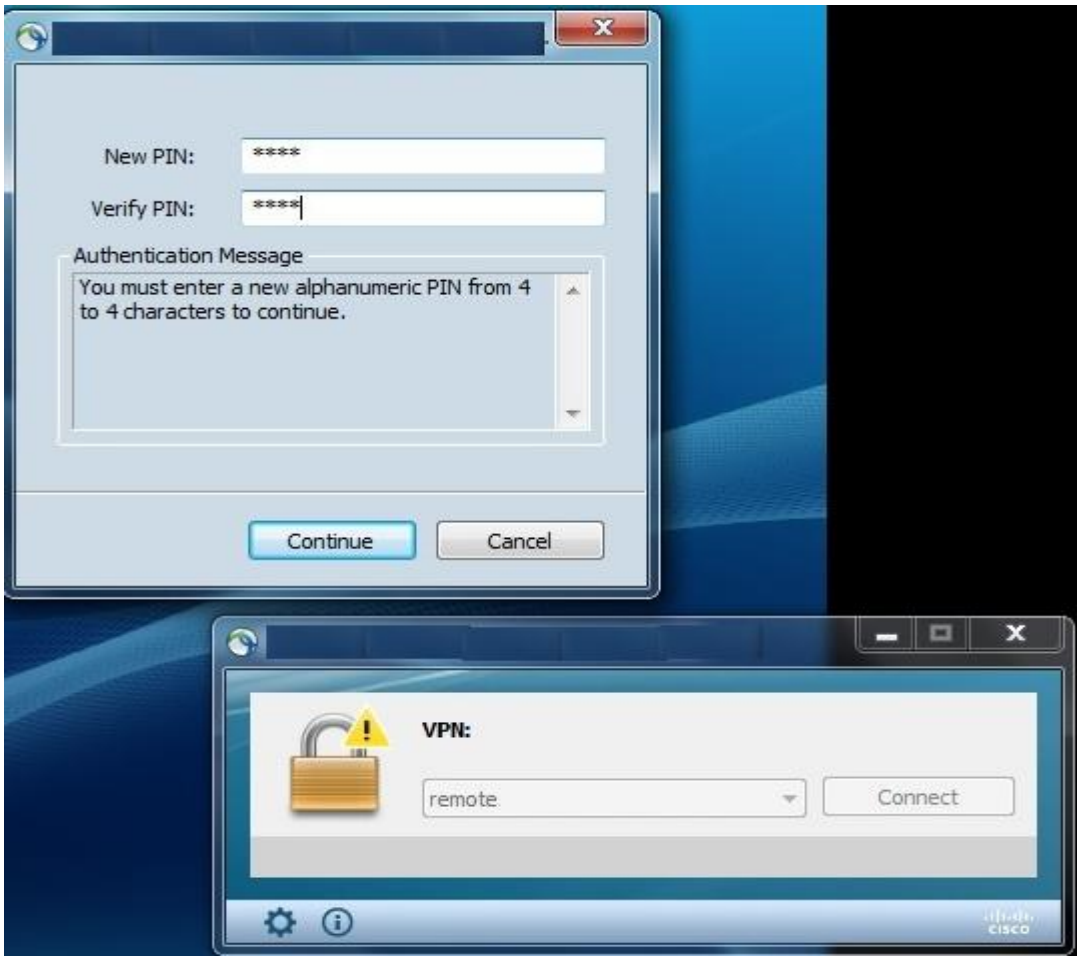
Öncelikle tanımlanan kullanıcının Token'ı üzerindeki altı haneli kod hiçbir zaman sisteme erişimde tek başına yeterli olmayacak sadece kullanıcının sisteme ilk erişimde geçici bir özellik taşıyarak (dört haneli pin oluşturmak için girilir) ekrana girildiğinde sonraki menüde pin kodu oluşturunuz mesajı alarak ilerlemesini sağlayacaktır.Bknz **Şekil.9.** Token Kod Giriş Ekranı ve **Şekil.10.** Token Kod New Pin Oluşturma Ekranı.



Şekil.9. Token Kod Giriş Ekranı

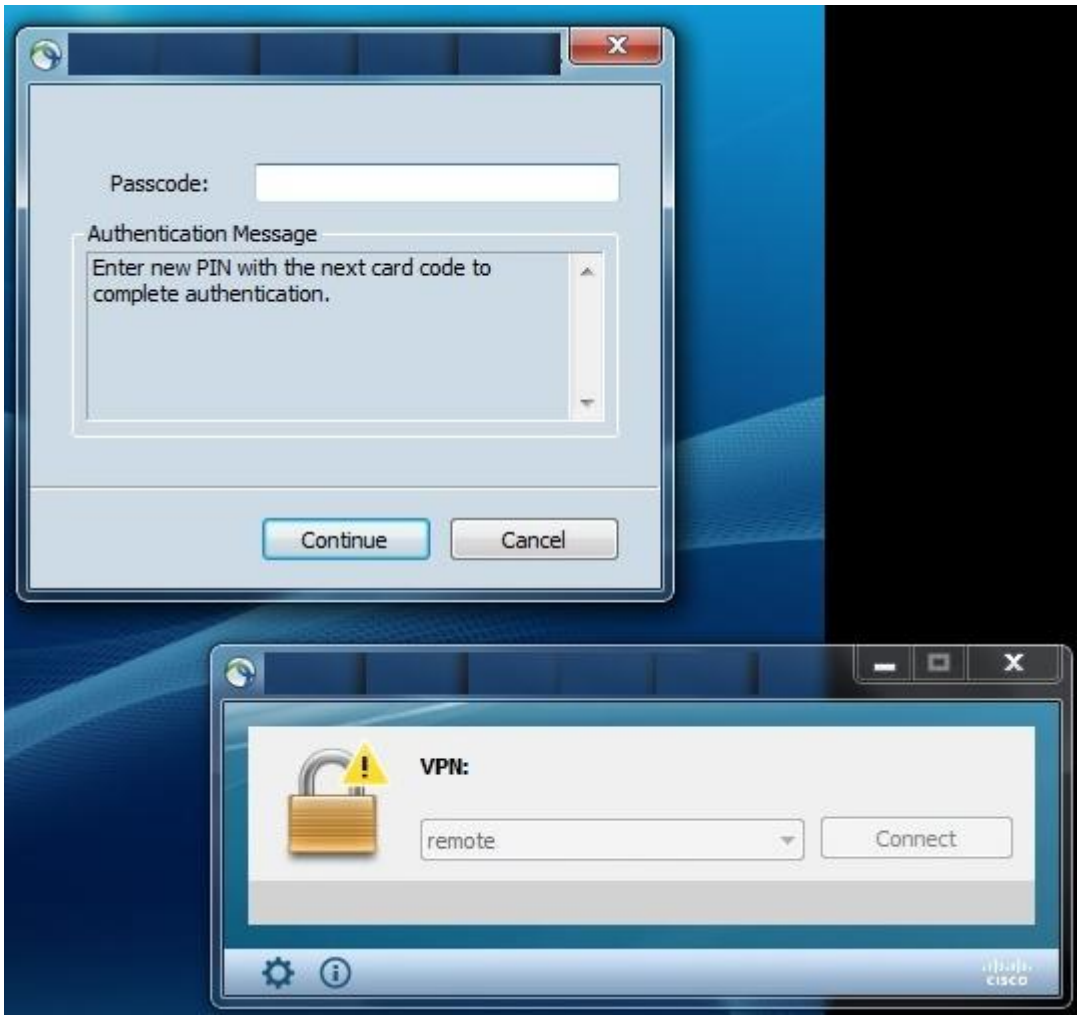


Şekil.10. Token New Pin Oluşturma Ekranı



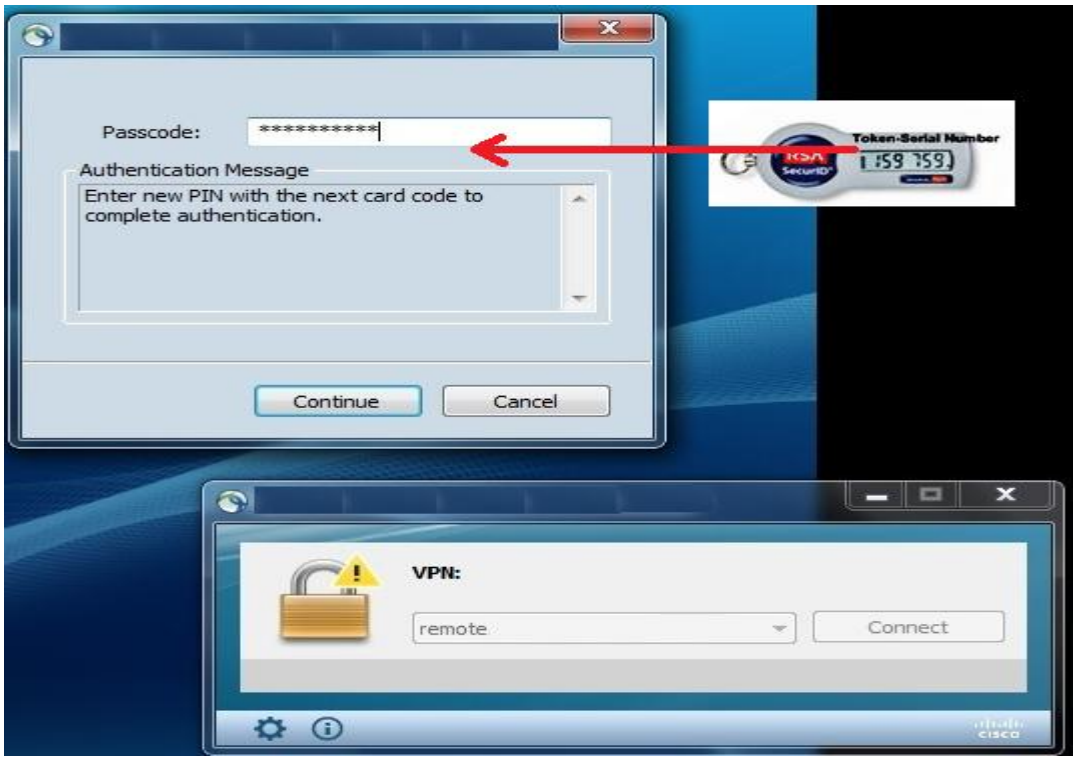
Şekil.11. Token New Pin Oluşturulmuş Görünümü

Pin kodu (dört haneli) oluşturulduktan sonra sistem kullanıcıyı Passcode (Pin kodu + Token kodu = Passcode) ekranına yani hem oluşturduğu dört haneli pin ve ardından boşluk bırakmadan bitişik olarak altı haneli token kodunu (token üzerinde görülen altı haneli ve her altmış saniyede bir değişen kod) yazması gerektiği menüye yönlendirmektedir.

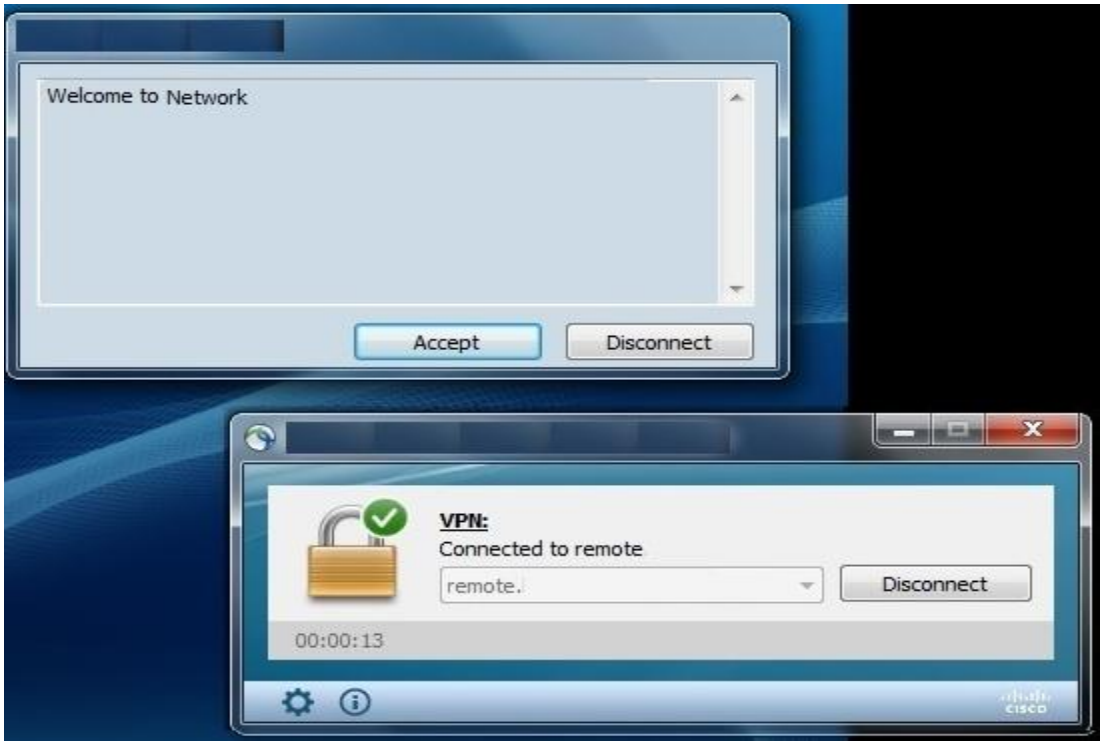


Şekil.12. Token Pin Oluşturuldukran Sonraki Passcode Giriş Menüsü

Belirtilen şekilde passcode aşağıdaki gibi 4 haneli pin kodu + 6 hane token üzerindeki kodu ile toplam 10 hane olarak yazıldıktan sonra sisteme bağlantı sağlanarak VPN erişimini güvenli bir şekilde tamamlandığı anlaşılmaktadır.



Şekil.13. Passcode Girişi Ekran Görünümü

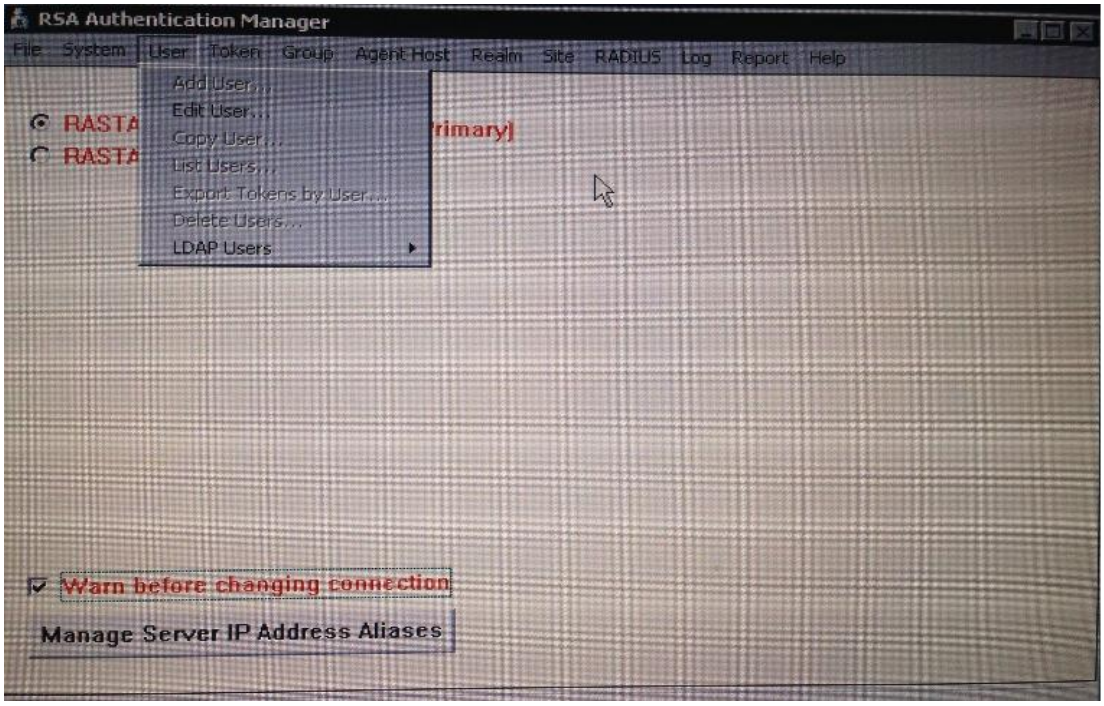


Şekil.14. Passcode Girişi Sonrası Bağlantı Sağlandı Ekran Görünümü

3.3. Token Kullanımı Başarılı Olmayan Durumlarda VPN (Virtual Private Network) Sistem Erişim Ayarlarının Kontrolleri

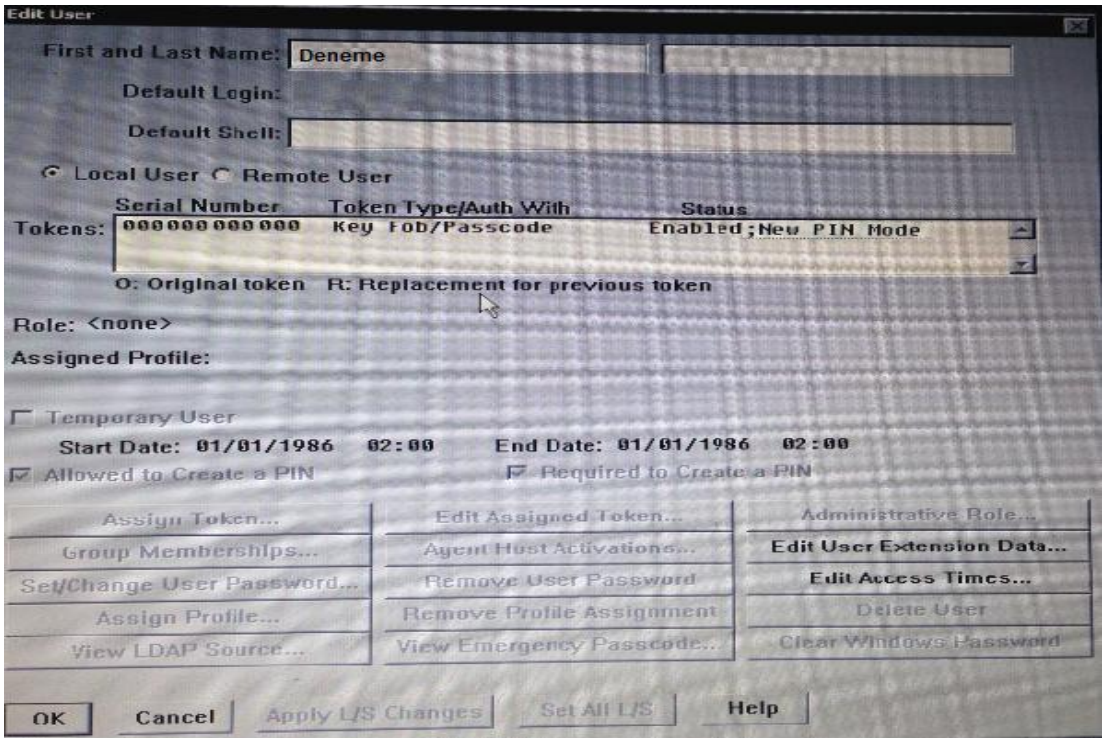
Kullanıcı bu erişim denemesinde hata ile karşılaşırsa ekli menülerden sistem yönetici tarafından user name, token seri no gibi kontroller sonrası cihazın sistem ile konuşarak bilgileri eşleştirip eşleştirmediği kontrolü yapılabilir.

Öncelikle kullanıcının adı bilgisinin doğru olup olmadığı kontrol edilmelidir.Bazı sistemlerde ad + soyad , adın ilk harfi + soyadın tamamı, ad + soyadın ilk harfi, kullanıcı id (sicil numarası vb) numarası gibi farklı özelliklerde kullanıcı adları kullanılmaktadır.



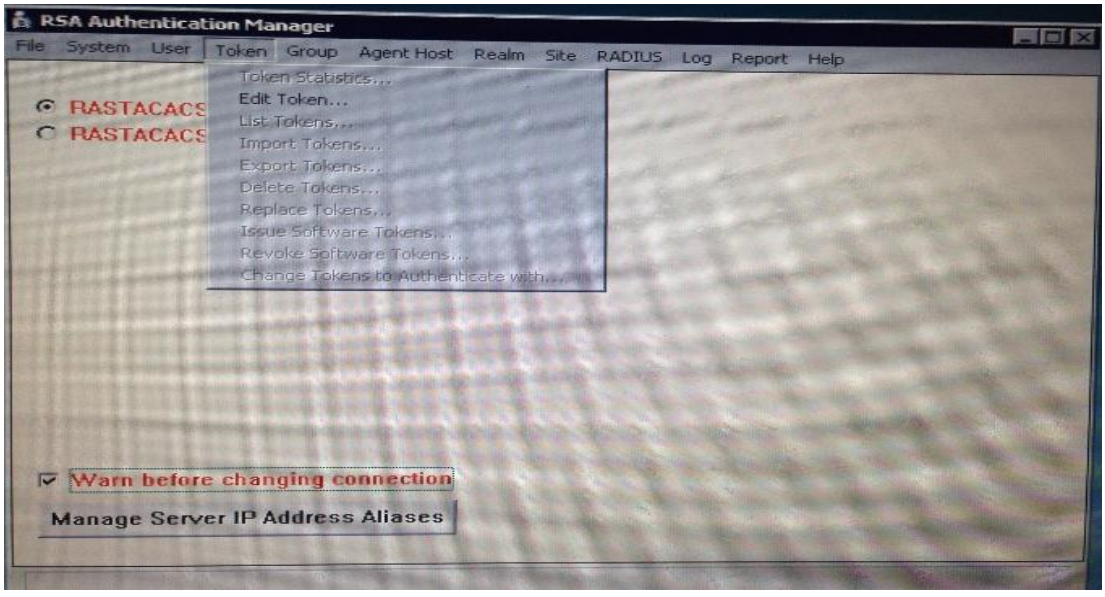
Şekil.15. RSA(Remote Admin Server) Authenticvation Manager Ekranında Edit User Kontrolü

Kullanıcı Enable New Pin Mode'da yani sorunsuz bir şekilde kullanıcı adı doğru görülüp neden hata aldığı incelenmesi için diğer bir kriter olan token seri numarası kontrol edilmelidir.



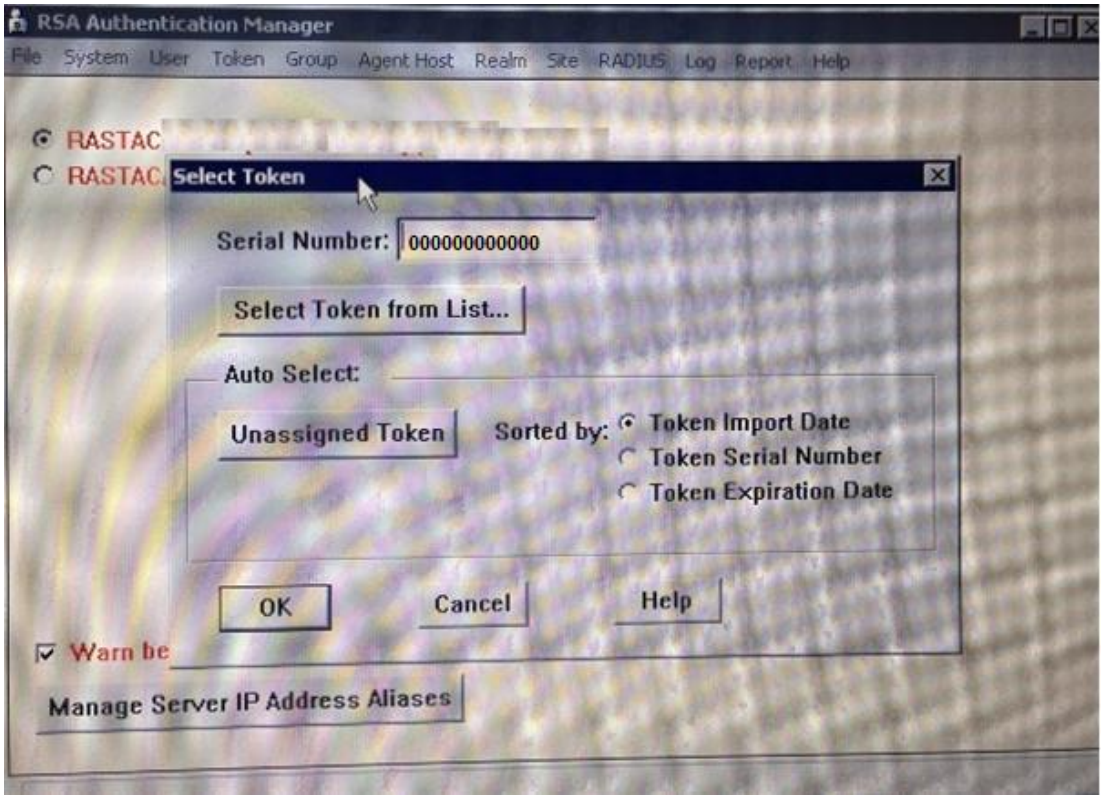
Şekil.16. Edit User Kontrolü

Token cihazının arkasındaki seri no kullanıcıdan istenerek aşağıdaki menülerden sistemde tanımlı token kodu eşleşip eşleşmediği görülebilir.



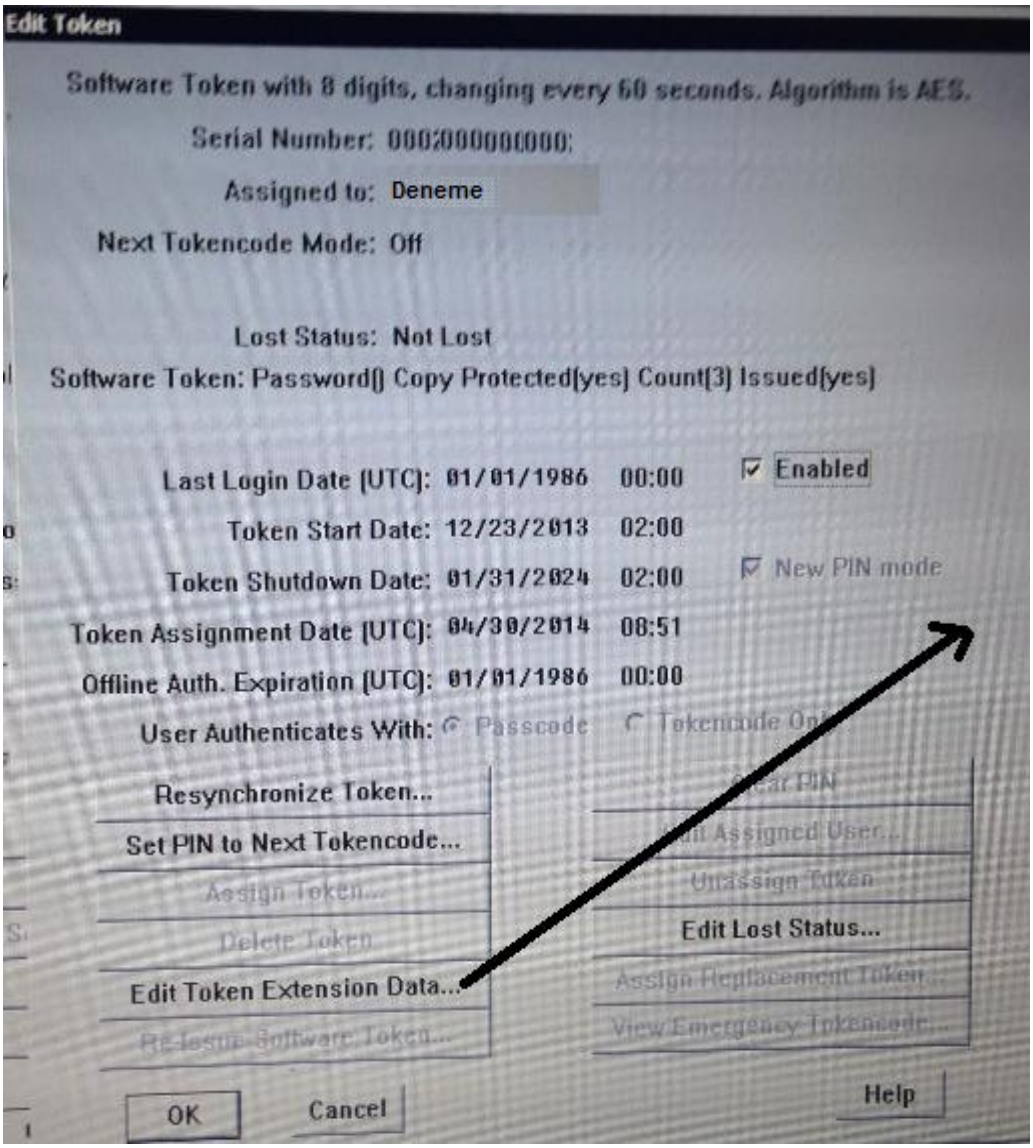
Şekil.17. Edit Token Kontrolü

Kullanıcı Enable New Pin Mode'da yani sorunsuz bir şekilde kullanıcı adı doğru görülüp neden hata aldığı incelenmesi için diğer bir kriter olan token seri numarası kontrol edilmelidir.Token cihazının arkasındaki seri no kullanıcıdan istenerek aşağıdaki menüden sistemde tanımlı token kodu eşleşip eşleşmediği görülebilir.



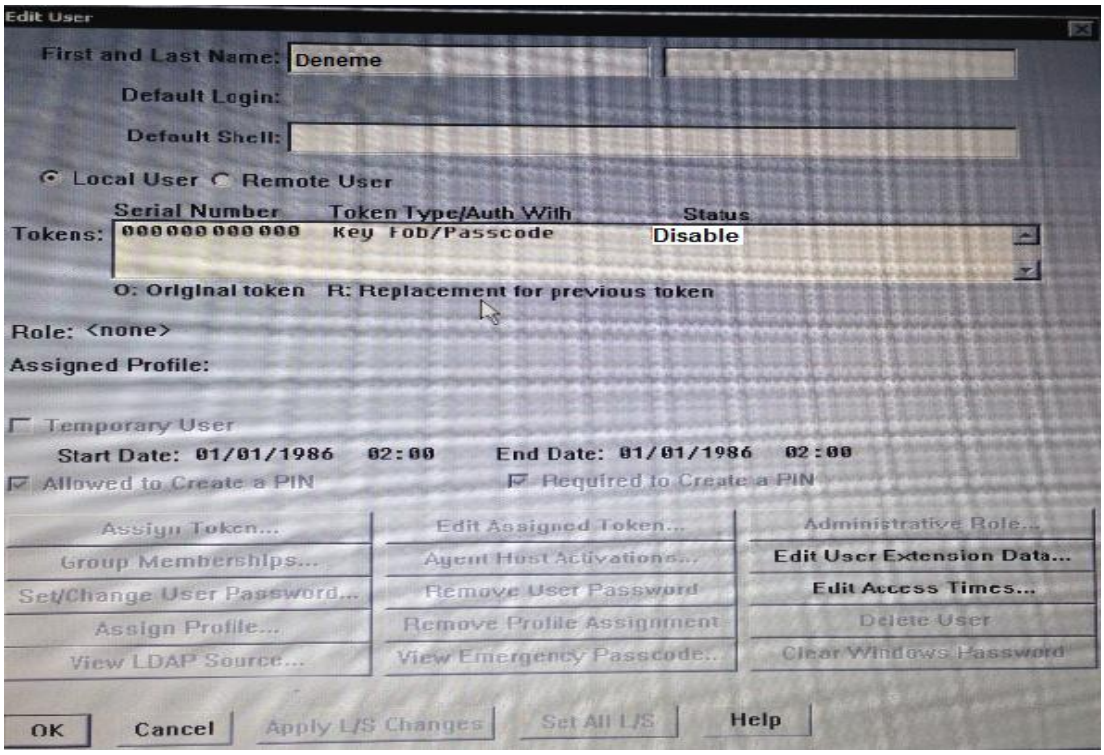
Şekil.18. Token Seri Numarası Kontrolü

Belirtilen token seri numarasıda sistemde doğru fakat kullanıcı doğru kullanıcı adı ve doğru tanımlı token ile aldığı kod ile deneyip giriş yapamayarak hata alıyorsa geriye tek bir ihtimal kalmaktadır.Bu ihtimal cihaz yani token ile sistem üzerindeki algoritma senkronize olmadığı için hata alınmaktadır. Edit Token Extention Data menüsünden bakılması gerekmektedir.



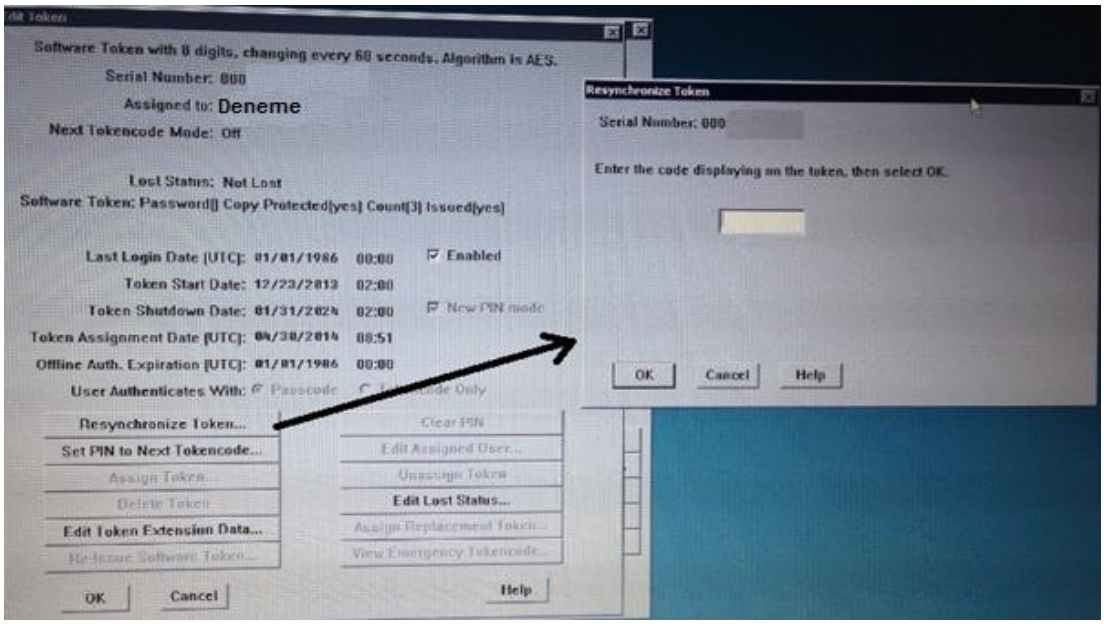
Şekil.19. Edit Token Extention Data Menüsü

Edit Token Extention Data menüsünden bakıldığında token Disable görülecektir.



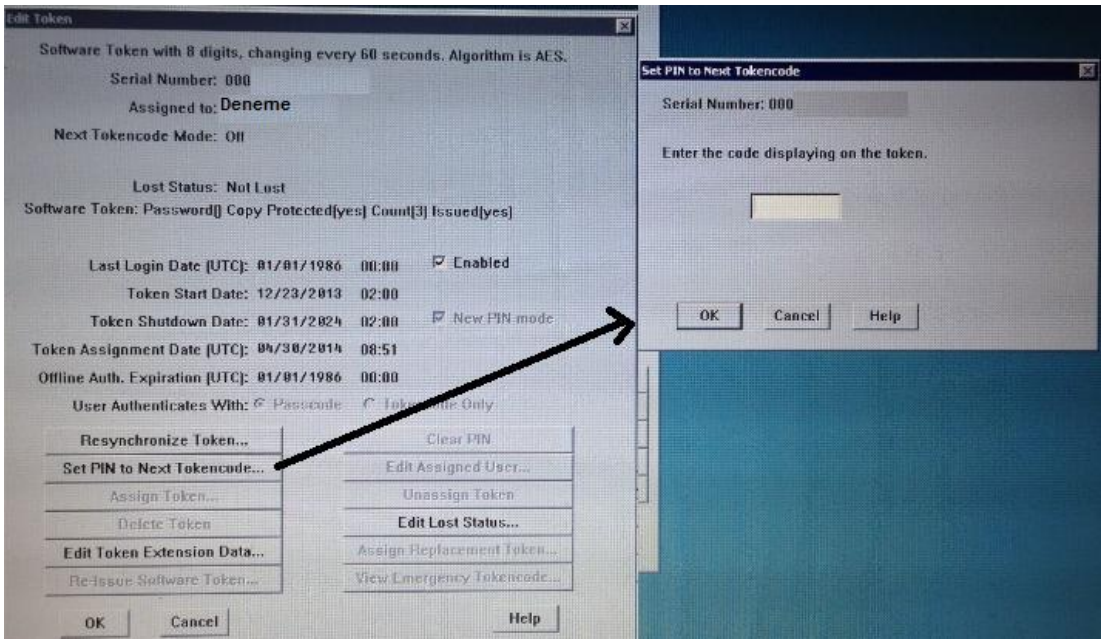
Şekil.20. Edit Token Extention Data Menüsü User Status

Token cihazı sistem ile senkronize çalışmıyor anlamına gelen bu hata sonrası manuel senkronize etme özelliği denenebilir.



Şekil.21. Reseynchronize Token Menüsü

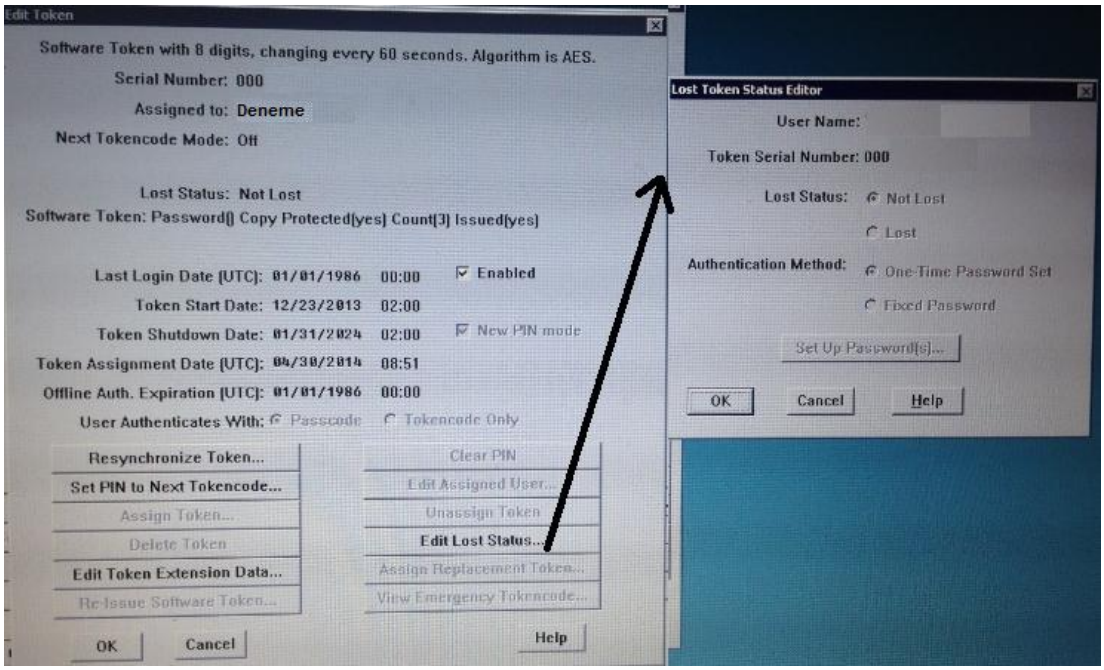
Resynchronize Token menüsüden kullanıcıdan token ekranındaki gördüğü o anki güncel token kodu istenerek bu alana girişi yapılır ve senkronizenin manuel tamamlanması sağlanır. Manuel senkronize işlemi bittikten sonra kullanıcı halen sorun yaşar ise kullanıcı sistemden silinip tekrar tanımlanması yerine son bir seçenek olan pin kodu manuel sistem üzerinden belirlenebilir.



Şekil.22. Set Token Pin Menüsü

Bu işlem yani pin kodu manuel sistem üzerinden belirlenmesi sistem yöneticisi tarafından yapılmaktadır. Kullanıcının bir sonraki denemede kendisi değiştirebilecek özellikte (set new pin next logon) olması güvenli bir şekilde VPN erişimini gerçekleştirmesini sağlayabilmektedir.

Tüm bu kontroller sonrası eğer halen kullanıcı sisteme erişemiyor ise token cihazının içerisinde üretilen kod özelliği bozulmuş yada yazılım kopyalanmış anlamına gelebilir.Bu şüphe ile cihaz LOST statüsüne getirilerek kullanıcıya yeni bir token cihazı tanımlanmalıdır.



Şekil.23. Lost Token Status Editor Menüsü

4. PİN KODU ve ÖNEMİ

4.1. Pin Kodunun Güvenlik Açısından Önemi

Pin kodu ile güvenlik arttırılması son zamanlarda kötü amaçlı atakların ve bilgi sızıntılarını önlemek adına geliştirilmesi gereken önemli bir durum haline gelmiştir.VPN gibi güvenli görülen erişim sistemlerinde bile eski yöntemlerdeki sabit parola yada token ile üretilen (60 saniye sonra yeni bir kod üretme özelliği) pin kodu girişi algoritmasının kaynak sağlayıcı firmadan çalınması ihtimali sebebiyle buna ek olarak önüne dört haneli sadece kullanıcının oluşturarak bileceği ayrıca pin oluşturulması güvenliği arttırmaya yeterlidir.Geçmişte buna örnek bir token cihazı üretici firmasının nakliye sırasında tüm yazılım ve cihazların seri numaralarına ait bilgileri üçüncü kişiler tarafından ele geçmesi sonucu firma büyük zorluk yaşamıştır.Mevcut piyasadaki ürünlerini toplatarak müşteri memnuniyetini kaybettiği ölçüde geri kazanamasada yeniden ürettiği cihazları müşterilerine gecikmeli olarak ulaştırmıştır.Bu aşamada bu firma ile çalışmama kararı olan şirketler olmuş ve firma büyük bir prestij kaybı yaşamıştır.

Mevcut token kodu uzunluğu altı hanelidir ve eğer önüne dört haneli (kullanıcı tarafından ilk sisteme erişimde zorunlu alan olarak gelip belirleyeceği) pin daha eklendiği taktirde erişim güvenliği kırılamaz şekilde çifte güvenlik katmanı oluşturulabilir.

4.2. Pin kodu neden dört haneli olmalı ?

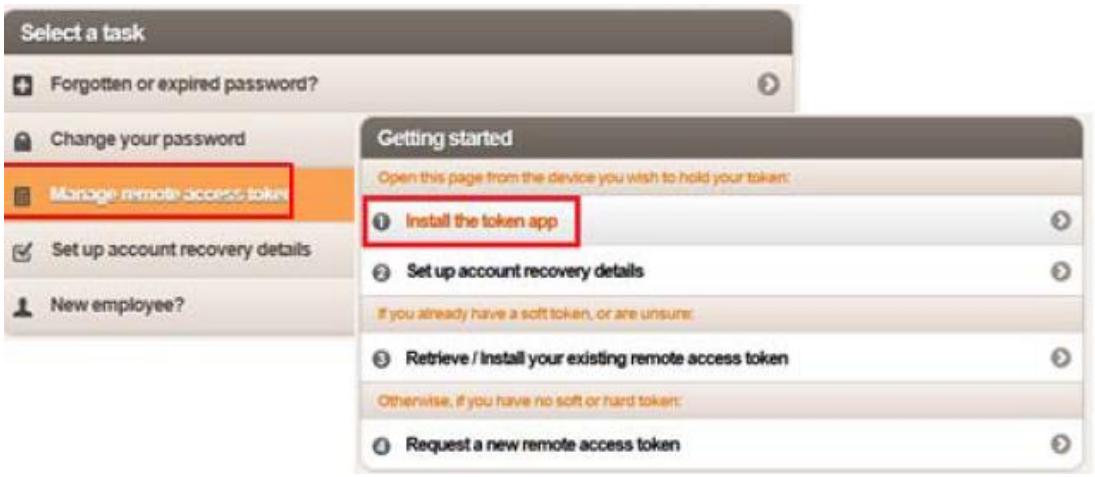
Pin kodu neden 4 hane daha uzun karakterde bir pin yapısı güvenliği arttırmada yüksek seviyeye ulaşmasını sağlamaz mı ? gibi oluşacak sorulara mantıklı bir açıklama gerekir ise bunun nedeni çok basittir.Kullanıcıların birçok alanda şifre kullanım alışkanlıkları dört haneli olarak benimsendiği için dört haneden uzun belirlenmesi isteneceği durumda bu hiçte kullanıcı dostu bir sistem olmayabilir.Banka kredi kartı şifresi, telefon sim kart pin kodu karakter uzunluğu, cep telefonu ekran güvenlik şifreleri vb. gibi birçok şifre özelliği karakteri dört hane olduğu için teknolojinin kullanıcıyı zorlamadan kullanıcı dostu olma özelliği amacı ile dört haneli belirlenmesi ve akılda kalıcı olması amaçlanmaktadır.VPN erişimi

esnasında dört haneli pin ve hemen ardından altı haneli tokendan elde edilen kodu yazarak toplamda on hane girerek (4 hane pin kodu + 6 hane token kodu = 10 haneli passcode) kullanıcının sisteme güvenli erişimi amaçlanırken aynız zamanda daha uzun karakterler girerek kullanıcı dostu olmayan bir yaklaşımı önlemek açısından dört haneli pin kodu olması mantıklı bir yaklaşımdır.

5. LİTERATÜR TARAMASI

5.1. Token ile Uzaktan Erişimde Geçici Pin ve Passcode Beraber Kullanımı

Token ile erişimde sadece token kodu güvenlik açısından yetersiz kalacağı düşüncesi ile Pin kodu sayesinde güvenliğin sağlanmasının bir örneği GSK adındaki bir uygulamada görülmektedir. Uygulamanın internetteki https://password.gsk.com sayfasındaki arayüzünden 30 dakikalık geçici pin alarak aynı sayfa üzerinden cep telefonunuza hard token (fiziki token) yanısıra soft token (yazılımsal token) yazılımı indirilerek token ile uzaktan erişim sağlanabilmektedir.

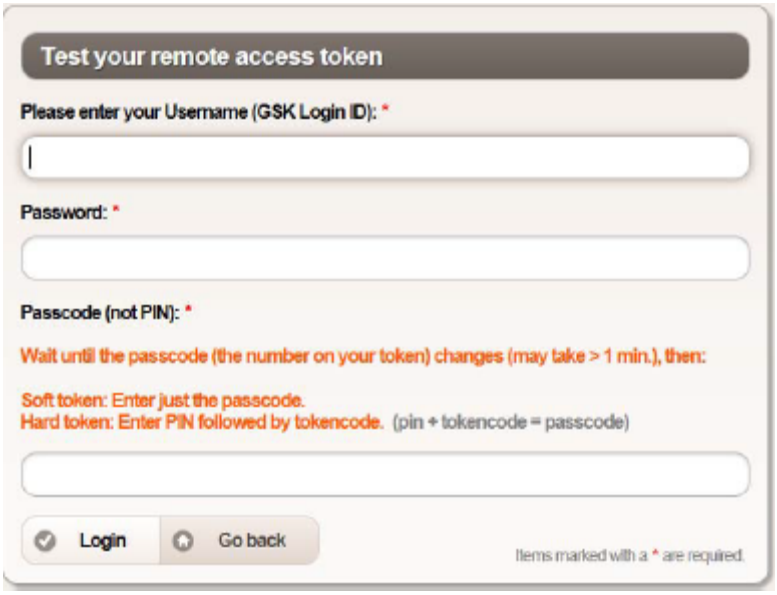


Şekil.24. GSK Soft Token Yazılımı İndirme Menüsü

Uygulamada passcode üretimi için siteden alınan 30 dakikalık geçici pin kodu (mail olarak yada sms ile tarafınıza 30 dakika geçerliliği olan pin kodu gelmektedir) girişinden sonra alınan soft token passcode üretir.

Şekil.25. GSK Soft Token Passcode Menüsü

Soft token ekranında üretilen passcode son olarak sitenin login alanında yazılarak uzaktan erişim gerçekleştirilmektedir. Uygulamanın hard token (fiziki token) kullanımında ise ekli belirtildiği şekilde pin + tokencode = passcode girişi yapılarak uzaktan erişim gerçekleştirilmektedir.



Şekil.26. GSK Uzaktan Erişim Testi Menüsü

Kullanıcı adı ve soft tokendan üretilen passcode girilerek yada hard token ile girişi doğrulandığı taktirde ekli şekildeki uzaktan erişim sağlandı uyarısı görülmektedir ve bu uzaktan erişimin başarılı bir şekilde gerçekleştiğini ifade etmekteddir.

Şekil.27. GSK Uzaktan Erişim Gerçekleşti Onay Menüsü

Yukarıda belirttiğimiz üzere token girişinde kullanılan pin sistemden mail yada sms ile alınarak aslında pekte güvenli olmadığı görülmektedir. Tamamen güvenli olabilmesi için kullanıcıya pin kodunun mail yada sms gibi ele geçirilebilecek şekilde gönderilmeden sistem üzerinden kullanıcının kendisi belirleyerek daha doğru ve güvenli bir yöntem olacaktır.

SONUÇ

VPN gibi güvenli görülen erişim sistemlerinde eski yöntemlerdeki sabit parola yada token ile üretilen (60 saniye sonra yeni bir kod üretme özelliği) pin kodu girişi algoritmasının kaynak sağlayıcı firmadan çalınması ihtimali sebebiyle buna ek olarak önüne dört haneli sadece kullanıcının oluşturarak bileceği ayrıca pin oluşturulması güvenliği arttırmaya yeterli olmuştur. Bu yöntemin çalışması zor olmamakla beraber kullanıcı tarafından kafa karışıklığına neden olmayacak ve basit bir mantıkla çalışmaktadır. Token cihazının ürettiği kod ile beraber sisteme ilk girişte kullanıcının kendi belirleyeceği (yalnızca kullanıcının kendi bileceği) dört hane ön pin oluşturarak sonraki denemesinde ve ileride bundan sonraki tüm sistem erişiminde kullanacağı bu basit güvenlik geliştirmesi ile pin artı token kodu (pin + token kodu = güvenli giriş kodu) birleşik yazılarak sisteme güvenli bir şekilde erişebilmiştir.

VPN erişim için kullanılan tokenlerin pekte güvenli olmadığı görülmektedir. Bu nedenle pin kodu ile güvenlik arttırılması gibi yeni yöntemlere ihtiyaç duyulmaktadır.

KAYNAKLAR

Ağ Protokolleri,
http://web.karabuk.edu.tr/ahmetcinkara/dersler/A%C4%9F%20sistemleri/agtemelleri.pdf

Cisco AnyConnect Secure Mobility Client,
http://www.cisco.com/c/en/us/products/security/anyconnect-secure-mobility-client/index.html

RSA Token Algoritması ve Akış Diyagramı,
http://www.google.com.tr/imgres?imgurl=http%3A%2F%2Fm.eet.com%2Fmedia%2F1120410%2Ffreersasecuridfig2large.jpg&imgrefurl=http%3A%2F%2Fwww.embedded.com%2Fdesign%2Fsafety-and-security%2F4230483%2FUnderstanding-the-security-framework-behind-RSA-SecurID&h=320&w=800&tbnid=5sYzYjBVycMtMM%3A&zoom=1&docid=Z--xAcSN7Xr-GM&ei=GEeGU4aOKKet7QaI-oDgAQ&tbm=isch&ved=0CLMBEDMoWTBZ&iact=rc&uact=3&dur=238&page=4&start=71&ndsp=28

SSL VPN,
http://www.google.com.tr/patents/US20080034416?printsec=drawing&hl=tr&dq=ssl+vpn

TCP IP Protokol,
http://www.msxlabs.org/forum/donanim/51750-bilgisayar-agi-network-nedir.html

Token Cihazı İçerisindeki Programın Java Yazılım Kodu,
http://www.java2s.com/Open-Source/Java/6.0-JDK-Modules-sun/security/sun/security/pkcs11/Token.java.htm

VPN Şema Örneği,
http://www2.compute.dtu.dk/~ender/Site/Publications_files/3.EnderABG2005-ip.pdf

Literatür Taraması GSK Hard ve Soft Token Örneği,
http://www.gsk.com/gsk_filestore/OWA/OWA%20Section%208.pdf

EKLER

Ek 1. Token Cihazı İçerisindeki Programın Yazılım Kodu

Program Java kod dilinde yazılmıştır.

```
package sun.security.pkcs11;

import java.util.*;
import java.io.*;
import java.lang.ref.*;

import java.security.*;
import javax.security.auth.login.LoginException;

import sun.security.jca.JCAUtil;

import sun.security.pkcs11.wrapper.*;
import static sun.security.pkcs11.wrapper.PKCS11Constants.*;

/**
 * PKCS#11 token.
 *
 * @author  Andreas Sterbenz
 * @version 1.11, 05/05/07
 * @since   1.5
 */
class Token implements Serializable {

    // need to be serializable to allow SecureRandom to be serialized
```

```
private static final long serialVersionUID = 2541527649100571747L;

// how often to check if the token is still present (in ms)
// this is different from checking if a token has been inserted,
// that is done in SunPKCS11. Currently 50 ms.
private final static long CHECK_INTERVAL = 50;

final SunPKCS11 provider;

final PKCS11 p11;

final Config config;

final CK_TOKEN_INFO tokenInfo;

// session manager to pool sessions
final SessionManager sessionManager;

// template manager to customize the attributes used when creating objects
private final TemplateManager templateManager;

// flag indicating whether we need to explicitly cancel operations
// we started on the token. If false, we assume operations are
// automatically cancelled once we start another one
final boolean explicitCancel;

// translation cache for secret keys
final KeyCache secretCache;
```

```
// translation cache for asymmetric keys (public and private)
final KeyCache privateCache;

// cached instances of the various key factories, initialized on demand
private volatile P11KeyFactory rsaFactory, dsaFactory, dhFactory, ecFactory;

// single SecureRandomSpi instance we use per token
// initialized on demand (if supported)
private volatile P11SecureRandom secureRandom;

// single KeyStoreSpi instance we use per provider
// initialized on demand
private volatile P11KeyStore keyStore;

// whether this token is a removable token
private final boolean removable;

// for removable tokens: whether this token is valid or has been removed
private volatile boolean valid;

// for removable tokens: time last checked for token presence
private long lastPresentCheck;

// unique token id, used for serialization only
private byte[] tokenId;

// flag indicating whether the token is write protected
private boolean writeProtected;
```

```
  // flag indicating whether we are logged in
  private volatile boolean loggedIn;

  // time we last checked login status
  private long lastLoginCheck;

  // mutex for token-present-check
  private final static Object CHECK_LOCK = new Object();

  Token(SunPKCS11 provider) throws PKCS11Exception {
this.provider = provider;
this.removable = provider.removable;
this.valid = true;
p11 = provider.p11;
config = provider.config;
tokenInfo = p11.C_GetTokenInfo(provider.slotID);
writeProtected = (tokenInfo.flags & CKF_WRITE_PROTECTED) != 0;
// create session manager and open a test session
SessionManager sessionManager;
try {
    sessionManager = new SessionManager(this);
    Session s = sessionManager.getOpSession();
    sessionManager.releaseSession(s);
} catch (PKCS11Exception e) {
    if (writeProtected) {
  throw e;
    }
    // token might not permit RW sessions even though
    // CKF_WRITE_PROTECTED is not set
```

```
    writeProtected = true;
    sessionManager = new SessionManager(this);
    Session s = sessionManager.getOpSession();
    sessionManager.releaseSession(s);
}
this.sessionManager = sessionManager;
secretCache = new KeyCache();
privateCache = new KeyCache();
templateManager = config.getTemplateManager();
explicitCancel = config.getExplicitCancel();
  }

  boolean isWriteProtected() {
return writeProtected;
  }

  // return whether we are logged in
  // uses cached result if current. session is optional and may be null
  boolean isLoggedIn(Session session) throws PKCS11Exception {
// volatile load first
boolean loggedIn = this.loggedIn;
long time = System.currentTimeMillis();
if (time - lastLoginCheck > CHECK_INTERVAL) {
    loggedIn = isLoggedInNow(session);
    lastLoginCheck = time;
}
return loggedIn;
  }
```

```
  // return whether we are logged in now
  // does not use cache
  boolean isLoggedInNow(Session session) throws PKCS11Exception {
 boolean allocSession = (session == null);
 try {
    if (allocSession) {
  session = getOpSession();
    }
    CK_SESSION_INFO info = p11.C_GetSessionInfo(session.id());
    boolean loggedIn = (info.state == CKS_RO_USER_FUNCTIONS) ||
      (info.state == CKS_RW_USER_FUNCTIONS);
    this.loggedIn = loggedIn;
    return loggedIn;
 } finally {
    if (allocSession) {
  releaseSession(session);
    }
 }
  }

  // ensure that we are logged in
  // call provider.login() if not
  void ensureLoggedIn(Session session) throws PKCS11Exception, LoginException
{
 if (isLoggedIn(session) == false) {
    provider.login(null, null);
 }
  }

  // return whether this token object is valid (i.e. token not removed)
```

```
  // returns value from last check, does not perform new check
  boolean isValid() {
if (removable == false) {
    return true;
}
return valid;
  }

  void ensureValid() {
if (isValid() == false) {
    throw new ProviderException("Token has been removed");
}
  }

  // return whether a token is present (i.e. token not removed)
  // returns cached value if current, otherwise performs new check
  boolean isPresent(Session session) {
if (removable == false) {
    return true;
}
if (valid == false) {
    return false;
}
long time = System.currentTimeMillis();
if ((time - lastPresentCheck) >= CHECK_INTERVAL) {
    synchronized (CHECK_LOCK) {
  if ((time - lastPresentCheck) >= CHECK_INTERVAL) {
      boolean ok = false;
      try {
```

```
    // check if token still present
    CK_SLOT_INFO slotInfo =
      provider.p11.C_GetSlotInfo(provider.slotID);
    if ((slotInfo.flags & CKF_TOKEN_PRESENT) != 0) {
        // if the token has been removed and re-inserted,
        // the token should return an error
        CK_SESSION_INFO sessInfo =
          provider.p11.C_GetSessionInfo
          (session.idInternal());
        ok = true;
    }
      } catch (PKCS11Exception e) {
    // empty
      }
      valid = ok;
      lastPresentCheck = System.currentTimeMillis();
      if (ok == false) {
    destroy();
      }
  }
    }
}
return valid;
  }

  void destroy() {
valid = false;
provider.uninitToken(this);
  }
```

```
  Session getObjSession() throws PKCS11Exception {
return sessionManager.getObjSession();
  }

  Session getOpSession() throws PKCS11Exception {
return sessionManager.getOpSession();
  }

  Session releaseSession(Session session) {
return sessionManager.releaseSession(session);
  }

  Session killSession(Session session) {
return sessionManager.killSession(session);
  }

  CK_ATTRIBUTE[] getAttributes(String op, long type, long alg,
    CK_ATTRIBUTE[] attrs) throws PKCS11Exception {
CK_ATTRIBUTE[] newAttrs =
     templateManager.getAttributes(op, type, alg, attrs);
for (CK_ATTRIBUTE attr : newAttrs) {
   if (attr.type == CKA_TOKEN) {
  if (attr.getBoolean()) {
     try {
   ensureLoggedIn(null);
     } catch (LoginException e) {
   throw new ProviderException("Login failed", e);
     }
```

```
  }
  // break once we have found a CKA_TOKEN attribute
  break;
    }
}
return newAttrs;
  }

  P11KeyFactory getKeyFactory(String algorithm) {
P11KeyFactory f;
if (algorithm.equals("RSA")) {
    f = rsaFactory;
    if (f == null) {
  f = new P11RSAKeyFactory(this, algorithm);
  rsaFactory = f;
    }
} else if (algorithm.equals("DSA")) {
    f = dsaFactory;
    if (f == null) {
  f = new P11DSAKeyFactory(this, algorithm);
  dsaFactory = f;
    }
} else if (algorithm.equals("DH")) {
    f = dhFactory;
    if (f == null) {
  f = new P11DHKeyFactory(this, algorithm);
  dhFactory = f;
    }
} else if (algorithm.equals("EC")) {
```

```
    f = ecFactory;
    if (f == null) {
  f = new P11ECKeyFactory(this, algorithm);
  ecFactory = f;
    }
} else {
    throw new ProviderException("Unknown algorithm " + algorithm);
}
return f;
  }

  P11SecureRandom getRandom() {
if (secureRandom == null) {
    secureRandom = new P11SecureRandom(this);
}
return secureRandom;
  }

  P11KeyStore getKeyStore() {
if (keyStore == null) {
    keyStore = new P11KeyStore(this);
}
return keyStore;
  }

  private synchronized byte[] getTokenId() {
if (tokenId == null) {
    SecureRandom random = JCAUtil.getSecureRandom();
    tokenId = new byte[20];
```

```
    random.nextBytes(tokenId);
    serializedTokens.add(new WeakReference<Token>(this));
}
return tokenId;
  }

  // list of all tokens that have been serialized within this VM
  // NOTE that elements are never removed from this list
  // the assumption is that the number of tokens that are serialized
  // is relatively small
  private static final List<Reference<Token>> serializedTokens =
    new ArrayList<Reference<Token>>();

  private Object writeReplace() throws ObjectStreamException {
if (isValid() == false) {
    throw new NotSerializableException("Token has been removed");
}
return new TokenRep(this);
  }

  // serialized representation of a token
  // tokens can only be de-serialized within the same VM invocation
  // and if the token has not been removed in the meantime
  private static class TokenRep implements Serializable {

private static final long serialVersionUID = 3503721168218219807L;

private final byte[] tokenId;
```

```
 TokenRep(Token token) {
    tokenId = token.getTokenId();
 }

 private Object readResolve() throws ObjectStreamException {
    for (Reference<Token> tokenRef : serializedTokens) {
  Token token = tokenRef.get();
  if ((token != null) && token.isValid()) {
      if (Arrays.equals(token.getTokenId(), tokenId)) {
    return token;
      }
  }
    }
    throw new NotSerializableException("Could not find token");
 }
  }

}
```

ÖZGEÇMİŞ

25 Haziran 1983 tarihi, İstanbul ili Bayrampaşa ilçesi doğumluyum. Liseyi İstanbulda, Bayrampaşa Rıfat Canayakın YDA Süper Lisesinde 2000 yılında tamamladıktan sonra aynı yıl Süleyman Demirel Üniversitesi Fen Fakültesi Bilgisayar Programcılığı Bölümü'ne kaydoldum. Lisans eğitimini 2011 yılında Anadolu Üniversitesi İktisat Fakültesi İktisat Bölümü'nde tamamladım. 2012 Yılında Beykent Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı Yüksek Lisans programına kaydoldum. 2004 Yılından beri Bilgi Teknolojileri sektöründe özel bir bankada çalışma hayatımı halen sürdürmekteyim.

Çalışma hayatımdaki ilgi alanlanlarım Erişim Yönetme Stratejileri, Uygulama ve Altyapı Güvenliğidir.

Yabancı dilim İngilizcedir.

Aday: Nurdoğan AYDOĞDU