Açık Anahtar Tabanlı Şifreleme Neden Zordur?

Albert Levi ve Mahmut Özcan
Sabancı Üniversitesi
Mühendislik ve Doğa Bilimleri Fakültesi
Orhanlı, Tuzla, İstanbul
levi@sabanciuniv.edu mahmut@su.sabanciuniv.edu

Özet

Bu bildiride açık anahtar tabanlı şifreleme teknikleri kullanan sistemlerdeki açık anahtarların dağıtım sorunu ele alınmıştır. Varolan sertifika ve PKI sistemlerinin bazı sorunlarından bahsedilmiş ve açık anahtar tabanlı şifreleme kullanan bir sistemin nasıl tasarlanması gerektiği konusunda bazı kriterler oluşturulmuştur. Bu kriterler ışığında Sabancı Üniversitesi'nde tasarlanan ve gerçekleme aşamasına geçilen Pratik ve Güvenli E-posta (PGE) sisteminin temel özelliklerinden bahsedilmiştir.

1. Giriş

Açık anahtar tabanlı şifreleme (Public Key Cryptography) sistemlerinin tarihi 1970'li yıllara dayanır. Diffie ve Hellman'ın [1] temellerini attığı bu sistemde zaman içinde birçok algoritma önerilmiştir. Rivest, Shamir ve Adleman'ın meşhur RSA algoritması [2] ve 80'li yıllarda parlamaya başlayan eliptik eğri tabanlı şifreleme sistemleri [3] halen kullanılmaktadır.

Her açık anahtar tabanlı şifreleme sistemi matematiksel zor problemlere dayanır. Örneğin RSA sisteminin güvenliği, büyük sayıların faktorizasyonunun zorluğuna dayanmaktadır. Ancak bu bildirideki zorluk kavramı bu tür bir zorluk değil, sözkonusu sistemlerin uygulamada getirdiği pratik zorluklardır.

Bu zorlukların en önemlisi açık anahtarların, sahiplerinin kimlikleri ile ilişkilendirilmiş bir şekilde dağıtılmasıdır. Bu amaçla sertifikalar ve PKI (Public Key Infrastructures – Açık Anahtar Altyapıları) kullanılmaktadır. Ancak bunların da kendine has sorunları vardır. Bildiride bu sorunlardan kısaca bahsedilecektir.

Bu bildiride eleştirisel bir bakış açısı sergilenmekle beraber, başarı kazanmış açık anahtar tabanlı şifrelemeye dayanan örnek uygulamalar da incelenip bir senteze varılacaktır. Temel amacımız, güvenli olmayı kendisine amaç edinmiş bir ürünün temel özelliklerini belirlenmesidir. Bu noktadan hareketle Sabancı Üniversitesi'nde geliştirilmeye başlanan güvenli e-posta uygulamasının temel özelliklerinden bahsedilecektir.

2. Hız sorunu

Açık anahtar tabanlı şifreleme algoritmaları ile yapılan işlemler (şifreleme, deşifreleme, sayısal imzalama ve imza doğrulama işlemleri) yavaş işlemlerdir. Kullanılan algoritma, anahtar uzunluğu ve uygulamanın koştuğu platform işlemlerin hızını belirleyen önemli faktörlerdendir. Ancak her ne şart altında olursa olsun, tek anahtarlı simetrik algoritmalar (DES, AES gibi) onlarca, hatta bazı durumlarda yüzlerce, kat daha hızlıdır. Buna rağmen gerek sunduğu kriptoanaliz direnci, gerekse de anahtar dağıtım kolaylıkları açısından açık anahtar tabanlı algoritmalar tercih edilmektedir.

İşlem süresini azaltmak için standartlaşmış hızlandırıcı mekanizmalar kullanılmaktadır. Sayısal imzalamada metnin kendisi değil de tek blok halinde özü (hash) imzalanmaktadır. Şifrelemede metin daha hızlı olan bir simetrik şifreleme algoritması ile şifrelenmekte, bu şifrelemede kullanılan anahtar ise açık anahtar tabanlı bir algoritma ile şifrelenmektedir. Böylelikle hem açık anahtar tabanlı sistemlerin hız sorunu kısmen de olsa aşılmış olmakta, hem de anahtar dağıtım avantajlarından yararlanılmaktadır. Yine de hızlı işlemin elzem olduğu, özellikle mobil uygulamalarda, açık anahtar tabanlı sistemler tercih edilmemektedir.

3. Açık Anahtar Dağıtım Sorunları

Açık anahtar tabanlı algoritmalarda iki anahtar vardır. Bunlar, (i) şifrelemede ve imza doğrulamada kullanılan ve herkesin bildiği açık anahtar, (ii) deşifrelemede ve imza atmakta kullanılan ve sadece sahibinin bildiği gizli anahtardır. Bu anahtarlar arasında bir matematiksel bağıntı bulunduğu halde açık anahtardan gizli anahtarı üretmek pratik olarak mümkün değildir.

Bu durumda anahtar dağıtım sorununun ortadan kalktığı düşünülebilir. Gerçekten de simetrik anahtar tabanlı (yani iki yönde de aynı anahtarı kullanan) sistemlere göre anahtar dağıtım sorunu daha azdır, ama sanılanın aksine ortadan kalkmamıştır. Açık anahtarlar herkese dağıtılabilir, ancak hangi anahtarın kime ait olduğundan da emin olunmalıdır. Bu konuda kişisel beyanlara güvenilemez. O yüzden sertifikalar kullanılmaktadır. Sertifika, en basit anlatımı ile, bir açık anahtar ile sahibinin kimliği arasındaki bağıntının belgesidir. Güvenilir sertifika otoriteleri (SO) tarafından üretilen bu sertifikaların içerdiği bilgilerin doğru olduğu varsayılır. Sertifika otoriteleri sayısal imzalarını kullanarak sertifika üretirler. Sertifika otoritesinin imzasını doğrulayan kişi aynı zamanda sertifika sahibinin açık anahtarını da öğrenmiş olur.

Sertifika sistemleri ve bağlı mekanizmaların oluşturduğu PKI (Public Key Infrastructure – Açık Anahtar Altyapısı) son yılların en gözde pazarlarından biri olmuştur. Bu konuda ticari ürünler geliştirildiği gibi milli PKI'lar da önerilmektedir. Türkiye için de bir PKI modeli önerilmiştir [4].

3.1. Sertifika ve PKI sorunun çözümü mü?

Sertifika sistemleri teknik olarak sorunu çözüyor gibi gözükse de yine de bazı noktalar açıkta kalmaktadır. Özellikle güven ve isim karışıklıkları ile ilgili bazı sorunlar Ellison ve Schneier tarafından [5]'te verilmiştir. Ancak daha genel sorunlar da vardır. Bunlar aşağıda anlatılmıştır.

3.1.1. Mahremiyetin korunamaması

Alınan sertifikalar Internet tarayıcı programlarla bütünleştikleri andan itibaren kullanımları kısmen de olsa sahibinin kontrolünden çıkmaktadır. SSL bağlantıları sırasında bazı sunucular istemci sertifikasını istemekte, istemci tarafında çalışan Internet tarayıcı programlar da bu sertifikaları sunucuya otomatik olarak göndermektedir. Böylelikle sertifika sahibinin kimliği ve e-posta adresi gibi bazı bilgileri sertifika ile beraber kontrolsüz şekilde Internet üzerinde dolaşmış olmaktadır. Bu durum mahremiyet savunucularının sertifika sistemlerine karşı en önemli saldırısını oluşturmaktadır.

3.1.2. Kayıt zorlukları

Sertifika üretilirken sertifika sahibinin kimliği SO tarafından doğrulanmalıdır. Bunun için ise kullanışlı olmayan, örneğin kişisel başvuru veya kimlik fotokopisi fakslamayı gerektiren, çevrim dışı yöntemler devreye girmektedir. Bu yöntemler sertifika sahibi olmayı zorlaştırmaktadır. Çevrim içi kayıt ve kimlik doğrulama yöntemlerini kullanan ve *class-1* sertifika olarak sınıflandırılan sertifika tipleri de vardır. Ancak bu tür sertifikalarda kullanılan kimlik doğrulama yöntemleri zayıf ve ataklara açıktır. Bu konuda daha geniş bilgi [6]'da bulunabilir.

3.1.3. Sertifikaların ücretli olması

Sertifikalar ücret karşılığı verilmektedir. Deneme amaçlı ücretsiz class-1 sertifikalar belli başlı SO'lar tarafından verilmektedir, fakat bunlar genelde geçici süreyle verilmekte olup süre bitiminde aynı anahtarın kullanımına devam etmek için ücretli sisteme geçmek gerekmektedir. Kaldı ki class-1 sertifikalarda kullanılan kimlik doğrulama yöntemleri yukarıda da belirtildiği üzere ataklara açık yöntemlerdir. Daha yüksek derecede güvenlik ve kimlik doğrulama prosedürleri gerektiren durumlarda (class-2, class-3 sertifikalar için) yıllık ücretler ödemek şarttır.

E-ödeme uygulamalarında son kullanıcı sertifikalarının maddi yükü, kredi kartlarının materyal masrafları gibi, bankaların üzerindedir. Bankaların toplamda yüklü bir miktar tutacak bu masrafı isteyerek üstlenmesini beklemek gereğinden fazla iyimserlik olur.

3.1.4. Güven sorunu

Sertifika üretip dağıtmak için gerekli yazılımları bulmak hiç de zor değildir. Kişisel bir SO kurup herkese ücretsiz sertifika dağıtmak da mümkündür. Ancak bu şekilde kurulan SO'ların var olan güven ağına girmeleri pek olası değildir. O yüzden bu SO'lar tarafından üretilen sertifikalar kullanıcılar tarafından kuşkuyla karşılanacaktır. SSL ve S/MIME sertifikaları için konuyu biraz daha derinleştirelim. SSL ve S/MIME sertifikalarını doğrulamak için gerekli kök SO sertifikalar Internet tarayıcı programlarla beraber gelmektedir. Kullanıcılar bu kök SO'lara güvenmek zorunda bırakılmaktadır. Çoğu kullanıcı, zaten sistemi anlayamadığı için, kök SO kavramından habersiz bir şekilde sistemi kullanmakta ve dolaylı olarak, ama farkında bile olmadan, bu SO'lara güvenmektedir. Sistemi az da olsa anlayan kullanıcılar ise isteyerek veya istemeyerek – en iyi ihtimalle "Internet tarayıcı program üreticisi bunlara güveniyorsa ben de güvenirim" veya benzeri bir mantıkla – kök SO'lara güvenmektedir. Güvenmek zorundadır, yoksa sistemi kullanamaz. Yazılımla beraber gelen kök SO'ların dışında bir SO ile karşılaşıldığında ise, Internet tarayıcı program sözkonusu SO'ya güvenilip güvenilmediğini soracaktır. Bu soruya cevap vermek çoğu kullanıcı için zordur. Çünkü sözkonusu yeni SO tanınmış biri değildir. Kullanıcıda, "güvenilir biri olsa zaten listede olurdu" şeklindeki bir düşünce hakim olacaktır.

3.1.5. Sertifika iptalinin getirdiği ek yükler

Sertifikaların sadece üzerindeki imzaları doğrulamak yeterli değildir. Tıpkı kredi kartlarında olduğu gibi üzerlerindeki son kullanma tarihleri aşılmamış bile olsa iptal edilip edilmediğinin sorgulanması gereklidir. Bu da sisteme ek yükler getirmektedir. Kullanıcılar ya çevrim içi bir sunucuya bağlanıp sertifikanın statüsü hakkında bilgi almak zorunda kalmakta, ya da periyodik olarak sertifika iptal listelerini (CRL – Certificate Revocation Lists) indirerek liste bazlı kontroller yapmaktadır.

3.1.6. Uygulama bazlı sorunlar

PKI sihirli bir değnek değildir. Sadece bir altyapıdır. Bu altyapıyı kullanacak uygulamalar olmak zorundadır. Varolan uygulamaların da bu altyapıya uygun hale getirilmeleri zorunludur. En fazla sorun da bu noktada yaşanmaktadır. Zaten hatırı sayılır bir yatırım yaparak PKI kuran kuruluşlar, bir de uygulamalarını değiştirmek zorunda kalmak ve bu konuda yatırıma gitmek istememektedirler.

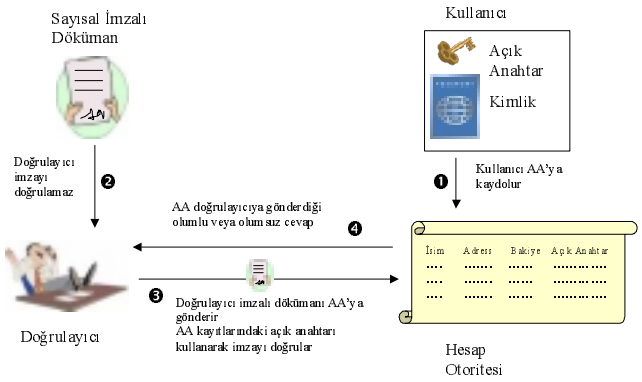
Uygulama ile ilgili başka bir sorun ise, sertifika kullanımını gerektiren e-ticaret gibi kişisel yanı da olan uygulamalarda, son kullanıcının bazı yazılımlar kurmak zorunda bırakılmasıdır. Kullanıcılar genellikle bu tür yazılımlara kuşkuyla yaklaşmaktadırlar. Gerek kurulumda karşılaşılabilecek bazı zorluklar, gerek zaman ve bant genişliği eksikliği, gerekse de virüs korkusu yazılım kurmayı gerektiren uygulamaların yaygınlaşması önünde önemli engellerdendir.

3.2. Sertifikasız da olur mu?

Açık anahtar tabanlı şifreleme sistemleri Diffie ve Hellman tarafından [1] ilk ortaya atıldığında önerilen yöntem, açık anahtarların yazmanın kısıtlı ama okumanın serbest olduğu "açık dosya" (public file) denilen bir dosyadan okunması idi. Sonradan bu kavram hiç kullanılmamıştır. Ancak daha sonra benzeri bir fikir, 90'lı yılların sonunda Wheeler tarafından, *Hesap Otoritesi Modeli* (Account Authority Model) olarak ortaya atılmıştır [7]. Sözkonusu model, açık anahtarların hesap otoritesi (HO) denilen güvenli sunucularda saklanması, gerektiğinde bu sunucuların açık anahtarları kullanarak sisteme destek vermeleri prensibine dayanmaktadır. Destek, sayısal imzaların HO tarafından doğrulanması veya gerektiğinde açık anahtarların HO'dan ihtiyaç sahibine çevrim içi aktarılması şeklinde olabilmektedir. Bu durumda klasik anlamda sertifika kullanımı gerekmemektedir, ancak güvenli otoritelere ihtiyaç devam etmektedir. HO mantığını kullanan bir e-ödeme protokolü [8]'de önerilmiştir.

HO kullanan örnek bir sayısal imzalama ve doğrulama sisteminin aşamaları Şekil 1'de gösterilmiştir. 1. aşamada kullanıcı açık anahtarını HO'ya kaydettirir. Bu bir defaya mahsus bir işlemdir; her imza için bu işlemin tekrarı gerekmez. Kullanıcı, doğrulayıcıya imzalı bir mesaj gönderdiğinde (2.

aşama), doğrulayıcı bu mesajı imzası ile beraber hesap otoritesine yönlendirir (3. aşama). Hesap otoritesi ise imzayı kendi hesap bilgileri içinde sakladığı kullanıcıya ait açık anahtarı kullanarak doğrular ve gerekli teyidi 4. ve son aşamada doğrulayıcıya gönderir.



Şekil 1. Hesap otoritesi kullanarak sayısal imzalı doküman doğrulama örneği

HO kavramı hepsini olmasa da sertifika tabanlı sistemlerin çoğu sorununu çözmektedir. Ortada bir sertifika olmadığı için 3.1.1'de bahsedilen mahremiyet sorunları yaşanmamaktadır. Olmayan sertifikaların iptali de gerekmemektir. HO hizmetleri ücretli olabilir ama uygun uygulamalarda sistemin zaten içinde olan HO'lar bu hizmeti ücretsiz sağlayabilirler. Örneğin e-ödeme sistemlerinde bankalar HO olabilir ve bu hizmet için ücret talep etmeyebilirler. Yine sistem içinden HO'lar seçerek sorunları en aza indirgenebilir.

4. Başarı Hikayeleri

Eleştirilse de açık anahtar tabanlı şifreleme kullanan başarılı uygulamalar yok değildir. PGP (Pretty Good Privacy) [www.pgp.com], SSL (Secure Socket Layer) [9] ve çoğunlukla sertifika gerektirmeden kullanılabilse de SSH (Secure SHell) [www.ssh.com] başarılı uygulamalar olarak göze çarpmaktadır.

4.1. PGP

PGP güvenli bir e-posta yazılımıdır. PGP'nin başarısının ardında iki önemli etken vardır. Birincisi ücretsiz bir yazılım olması, ikincisi ise güçlü şifreleme algoritmaları içermesidir. PGP bu özelliklerini kullanarak hatırı sayılır bir kullanıcı kitlesine ulaşmıştır. Aslında kullanımı kolay bir yazılım değildir. Kendine has ve oldukça karmaşık bir güven ve sertifika modeli içerir. Sistemin yararı, isteyenin kendi güven sistemini istediği şekilde oluşturmasındadır. Ancak bunu yapabilmek için kullanıcının şifreleme ve güvenlik konularında az da olsa bilgi sahibi olması gerekir. Karmaşıklığına rağmen PGP, ücretsiz ve açık kaynak kodlu olma özelliklerini çok iyi kullanarak güvenlik konusunda hassas bir kullanıcı kitlesini kendine bağımlı kılmayı bilmiştir.

4.2. SSL

SSL, PGP'nin aksine, kişisel değil organize bir çaba sonucu gelişmiş bir endüstri standardıdır. Güvenli HTTP bağlantısı sağladığı ve özellikle Internet tarayıcı programlar tarafından desteklendiği (ve böylelikle kullanıcıya ek program kurma yükü getirmediği) için büyük bir kullanıcı kitlesine ulaşmıştır. Güvenli bir HTTP bağlantısı kurulumu sırasında yaşanan mesaj alışverişi, son kullanıcıya saydam bir şekilde gelişmekte, kullanıcı olan biteni farkına bile varmamaktadır. Kök SO sertifikaları da yazılımla beraber geldiğinden kullanıcının herhangi bir anahtar girme zorunluluğu yoktur. Bu kolaylıklar, bazı güvenlik detaylarının gözden kaçmasına[1] sebep olmakla beraber, son derece önemli kullanım kolaylıkları sağlamaktadır.

4.3. SSH

Daha çok telnet ve ftp gibi uzaktan erişim protokolleri yerine kullanılan ve sunucu ile istemci arasındaki iletişimi (özellikle password'u) şifrelemeye yarayan SSH protokolü aslında her zaman sertifika gerektirmemektedir. İstemci, sunucuya ilk bağlantı sırasında sunucunun gönderdiği açık anahtarı çevrim dışı yollarla (örneğin anahtarın özütünü sistem yöneticisini telefonla arayıp kontrol ederek) doğrulayıp listesine ekleyebilir. Böylelikle sertifika gerektirmeden sunucunun açık anahtarı istemci tarafından öğrenilmiş olur. Bu işlem bir seferlik bir işlemdir. Kaldı ki SSH sisteminin kullanım amacı sunucuda hesabı bulunan kısıtlı sayıdaki kullanıcıya hizmet vermektir; her Internet kullanıcısı sunucunun SSH açık anahtarına ihtiyaç duymaz. O yüzdendir ki PKI benzeri bir yapı çok da gerekli değildir. Ancak SSH, sertifika ve PKI sistemlerini de desteklemektedir (ücretli versiyonlarında). SSL örneğinde olduğu gibi kök sertifikalar yazılımla beraber gelmektedir. Ayrıca çevrim içi sertifika dizinlerine erişim desteği de içermektedir.

SSH, ücretsiz dağıtıldığı ve güvenli uzaktan erişim alanında çok önemli bir eksikliği doldurduğu için önemli bir kullanıcı kitlesi kazanmıştır.

5. Nasıl bir Sistem?

Açık anahtar tabanlı şifreleme kullanan uygulamalar hız ve anahtar dağıtım sorunları yüzünden eleştirilse de özellikle kişisel kullanım alanında önemli bir kullanıcı kitlesi kazanmaktadır. Son kullanıcıyı hedefleyen böylesine bir sistemin başarı kazanıp geniş bir kitle tarafından kullanılması için:

1. uygulamanın önemli bir güvenlik açığını adreslemesi,
2. kullanışlı bir versiyonun ücretsiz dağıtılması veya Internet tarayıcı programlarla beraber gelmesi,
3. güvenlik konularında bilgi sahibi olmayan kullanıcıların dahi kolay kullanabilmesi,
4. işlemlerin mümkün olduğunca kullanıcıya saydam gelişmesi,
5. sistemin kullanıcıya anlayamayacağı detay sorular sormaması,
6. kişisel sertifika edinmenin şart koşulmaması,
7. ama güvenlik konusunda bilgili ve araştırmayı seven bir kullanıcı kitlesine de gerekli opsiyonları sunması gereklidir.

Kapalı ve genel amaçlı PKI sistemlerinin altyapı amaçlı kurulumunda ve kullanılmasında başarı sağlanması ise daha değişik şartlara bağlıdır. Bunun için:

[1] Örneğin, bağlanılan sunucunun gerçek kimliği otomatik olarak doğrulanmamaktadır. Bunun için son kullanıcının sertifika detaylarını incelemesi gerekir ki bu da çoğu kullanıcı tarafından yapılmaz.

1. kuruluşun, PKI kurmakla yeni bir altyapı kurduğunun bilincinde olması,
2. bu altyapının üstüne yeni bir yapılanmanın kaçınılmaz olduğunu anlaşılması,
3. yatırımın geri dönüşünün uzun süreceğinin göz önüne alınması şarttır.

6. Sabancı Üniversitesi'nde Yapılan Çalışmalar

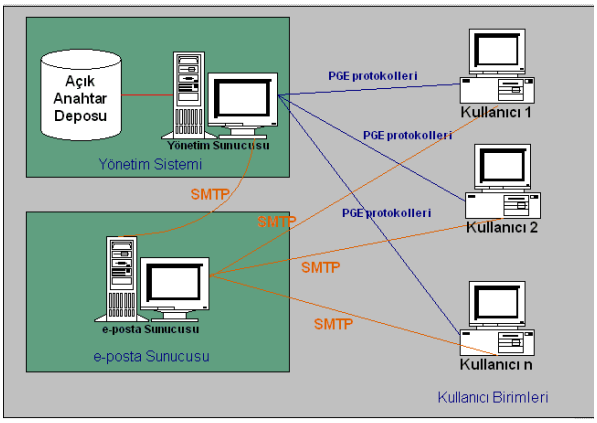
Sabancı Üniversitesi'nde açık anahtar tabanlı pratik sistemler geliştirmek üzere bir dizi çalışma başlatılmıştır. Geniş bir uygulama yelpazesinde devam edecek bu çalışmaların ilk ayağını pratik ve güvenli bir e-posta sistemi oluşturmaktadır. Kısaca PGE olarak adlandırılan bu sistem 2.2'de anlatılan hesap otoritesi (HO) kavramından esinlenilerek tasarlanmıştır. Sertifika tabanlı olmayan bu sistemde açık anahtarlar merkezi ve güvenilir bir sunucu tarafından istem bazında son kullanıcıya saydam bir şekilde dağıtılmaktadır.

PGE kendi içerisinde değişik gizlilik seviyelerinde haberleşmek isteyen şirket,üniversite gibi özelleşmiş birimlerin veya arkadaş gruplarının ihtiyacını karşılamak üzere tasarlanmıştır. Potansiyel kullanıcılarının merkezi bir sunucuda kayıtlarının bulunması sistemin önemli varsayımlarındandır. Şimdilik tasarımın bir parçası olmamakla beraber ileride değişik PGE sunucularının kendi aralarında haberleşerek ayrı PGE alanlarındaki kullanıcıların birbirleri ile güvenli haberleşmesini sağlayacak bir yapı değişikliğine gidilmesi düşünülmektedir.

PGE sisteminin mimarisi tasarlanmış olup gerçekleme aşamasına geçilmiştir. Tamamı JAVA teknolojileri kullanılarak gerçekleştirilecek PGE'nin Türkçe ve İngilizce arayüzleri olacaktır. PGE ücretsiz dağıtılacaktır.

6.1. PGE Mimarisi

PGE sadece bir sunucu yazılımı ve istemci yazılımlarından oluştuğu için ağ trafik yoğunluğuna aşırı bir yüklenme getirmez ve var olan ağ tasarımlarında bir değişiklik istemez. PGE Yönetim Sistemi (YS), Kullanıcı Sistemi (KS) ve bu sistemlerin birbirleriyle güvenli haberleşmesini sağlayan protokollerden oluşur. Sistemin mimarisi Şekil 2'de gösterilmiştir.



Şekil 2. PGE sistem mimarisi

1. Yönetim Sistemi (YS): YS temel olarak kullanıcıların açık anahtarlarının depolanması ve yönetiminden sorumludur. Kullanıcılar açık anahtarlarını YS'ye gönderip depolanmasını sağlarlar veya e-posta atmak istediği diğer kullanıcıların açık anahtarını sorgulayıp YS depolarından doğruluğundan emin olarak alırlar. YS kullanıcılarının açık anahtarlarını da sadece kendisinin yetkisinde bulunan açık anahtar depolarında (veritabanı) saklar.

2. Kullanıcı Sistemi (KS): KS aslında bir e-posta istemci programıdır. KS yazılımı ilk ayarları yapılırken anahtar çiftlerini üretip YS'ye açık anahtarını gönderir ve açık anahtar deposunda depolanmasını sağlar. Sayısal imzalı ve/veya şifreli e-postalar almak veya göndermek için gerekli tüm açık anahtarları kullanıcıya tamamen saydam bir şekilde YS'den alır. Bunun için sadece ilk kurulum aşamasında yüklediği YS'nin açık anahtarını kullanır.

3. Protokoller: YS ve KS arasındaki haberleşmeler için özel protokoller tanımlanmıştır. PGE protokollerinin en önemli özelliği çevrimdışı herhangi bir işleme ihtiyacın olmamasıdır. Bu yönüyle PGE kullanıcılarına büyük bir rahatlık ve kolaylık getirmiştir. Bu protokoller gerekli olduğu yerlerde kullanıcı ve sunucu doğrulama yapmakta ve değişilen mesajların bütünlüğünden emin olmaktadır. Önemli PGE protokolleri arasında "İlklendirme Protokolü", "Açık Anahtar İsteme Protokolü" ve "Açık Anahtar İptal Protokolü" sayılabilir. İlklendirme protokolleri haricinde diğer tüm protokollerde gerekli her yerde sayısal imzalı ve şifreli mesajlaşmalar yapılmıştır.

- İlklendirme protokolü, KS'nin ilk kurulumu sırasında veya daha sonra bir defaya mahsus olarak yapılan ve kullanıcının açık anahtarının YS sunucusuna kaydedildiği protokoldür. Kayıt sırasındaki kimlik doğrulama işlemi pratik bir şekilde e-posta aracılığı ile gönderilen şifreler ile yapılır. Ek önlem olarak kullanıcılardan bu şifre ile beraber iki tarafın da bildiği bazı yarı-gizli bilgiler (annesinin kızlık soyadı gibi) istenmektedir.

- Açık Anahtar isteme protokolü, herhangi bir KS'nin başka bir kullanıcının açık anahtarını öğrenmek için YS'yi sorguladığı protokoldür. Bu protokol KS kullanıcısına saydam gelişir ve sadece YS'nin cevabının imzalı olması yeterlidir.

- Açık Anahtar İptal protokolü, KS'nin sisteme kayıtlı açık anahtarını sistemden silmek için kullandığı protokoldür. İptal isteği iki şekilde doğrulanabilmektedir. Eğer KS'nin açık anahtarı kayıp değilse, KS imzalı bir iptal istek mesajı gönderir. Anahtar kayıp veya ulaşılamaz ise, ilklendirme protokolünde olduğu gibi, iptal isteği e-posta yolu ile gönderilecek şifrenin ve yarı-gizli bilgilerin değiş tokuşu ile doğrulanabilir.

6.2. PGE'nin avantajları

- SO'dan sertifika almayı gerektirmez. Böylelikle kullanıcılarına ek maliyet de getirmez.
- Sisteme kayıtlı kullanıcılar otomatik olarak imzalı ve şifreli e-posta almaya hemen başlarlar.

- Başkalarının açık anahtarını öğrenmek için manuel sorgulamalar yapmaya gerek yoktur. Açık anahtar öğrenme işlemleri kullanıcıya saydam gelişir.
- Var olan sistemin yükünü arttırmaz.
- Ağ yapılarında değişikliğe ihtiyaç duymaz.
- Ağ trafiğine diğer altyapılarıyla aynı yükü getirir.
- PGE, kullanıcılarına kimlik doğrulama, mesaj bütünlüğü, inkar edememezlik ve gizlilik gibi temel güvenlik prensiplerini sunar.
- KS, bilinen diğer e-posta programlarından farklı olmadığı için KS'ye geçmek kolaydır.
- PGE, tüm işlemlerini çevrimiçi yapar.
- PGE kullanıcılarının yanlarında taşımak zorunda oldukları fiziksel kartlara (smart-card, e-token vs.) ihtiyacı yoktur.

6.3. PGE ve diğer güvenli e-posta sistemleri arasındaki temel farklar

Bu kısımda PGE ile PGP [www.pgp.com] ve S/MIME (Secure / Multipurpose Internet Mail Extensions – Güvenli / Çok amaçlı Internet Posta Uzantıları) [10] sistemleri arasındaki farklar ve neden PGE'ye ihtiyaç duyulduğundan bahsedilecektir.

PGP güvenlik açısından çoğu kişi tarafından mükemmel olarak nitelendirilse de içerdiği açık anahtar dağıtım mekanizması ancak tecrübeli kişiler tarafından güvenli bir şekilde kullanılabilmektedir. Ortalama kullanıcı kitlesi, PGP açık anahtar sunucularından[2] indirdiği açık anahtarları sahiplerinin kimliklerinden pek de fazla emin olmadan kullanmak zorunda kalmaktadır. Bunun sebebi, PGP'nin "güven ağı" olarak nitelendirilen açık anahtar dağıtım ve sertifikasyon mekanizmasının karmaşıklığıdır. Ortalama kullanıcılar, bu mekanizmayı tam anlamadıkları için farkında olmadan tanımadıkları PGP kullanıcılarına güvendiklerini ima ederek sistemi kullanabilmektedirler. Bu da değişik aldatmacalara neden olabilmektedir.

Ayrıca PGP İngilizce arayüzlü bir programdır. İngilizce bilmeyen kullanıcılar rahat kullanamamaktadır.

PGE'yi PGP'den ayıran en önemli iki özellikten biri arayüzünün Türkçe destek vermesi, diğeri ise oldukça basit bir açık anahtar indirme ve kullanma protokolü içermesidir. Başkalarına ait açık anahtarlar güvenli, otomatik ve kullanıcıya saydam bir şekilde son kullanıcıya iletilmektedir. Bu açılardan bakıldığında PGE'nin, geniş bir ortalama tecrübeli kullanıcı kitlesi tarafından tercih edileceğine inanılmaktadır.

S/MIME ise sertifika tabanlı bir sistemdir. Kullanıcıların bu sistemi kullanmak için öncelikle ücretli sertifikalar alması gerekir. Bu da S/MIME'nin yaygınlaşmasını engellemektedir. PGE'de ise sertifika gerekmemektedir. Böylelikle yaygınlaşma daha çabuk olabilecektir.

[2] PGP açık anahtar sunucuları, yazmaya ve okumaya açık bir veritabanı mantığı ile çalışan ve hiçbir şekilde içerdiği anahtarlar ile sahipleri arasındaki ilişkiye garanti vermeyen sunucudardır. Kullanıcılar anahtarlar üzerindeki sertifikaları kendileri doğrulamak zorundadırlar.

7. Sonuçlar

25 senelik bir mazisi olan açık anahtar tabanlı şifreleme sistemleri ancak son 10 yılda son kullanıcılara erişen uygulamaların parçası olmuştur. Bu 10 yıl içinde de açık anahtar tabanlı şifreleme sistemlerinin hız ve açık anahtar dağıtım sorunları sistemlerin hızlı gelişmesinin önünde engel teşkil etmiştir. Güvenli bir şekilde açık anahtar dağıtım sorunu, sertifika otoriteliği (bazı kaynaklarda onay kurumu olarak da geçer) denilen yeni bir iş kolunun doğmasına aracılık etmiştir. Ancak yine de yaygınlaşma beklenen boyutlarda olamamıştır. Bu bildiride bu sorun incelenmiş ve açık anahtar tabanlı şifreleme kullanan ürünlerin tasarım kriterleri belirlenmiştir. Bu kriterler ışığında açık anahtar tabanlı olup da sertifika sistemlerini ve SO'ları kullanmayan bir uygulamanın mümkün olabileceği sonucuna varılmıştır. Bir pilot uygulama olarak da açık anahtarların merkezi bir şekilde saklanıp kullanıcıya saydam bir şekilde dağıtıldığı güvenli bir e-posta sistemi tasarlanmış ve gerçekleme aşamasına geçilmiştir.

Kaynakça

1. Diffie W., ve M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644-654, Kasım 1976.

2. Rivest, R., A. Shamir ve L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Şubat 1978.

3. Menezes, A., *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

4. Levi, A. ve M. U. Çağlayan, "Türkiye için bir Açık Anahtar Altyapısı Modeli," *Bilişim 98 - TBD 15. Bilişim Kurultayı Bildiriler Kitabı*, Istanbul, pp. 354-361, Eylül 1998.

5. Ellison C., ve B. Schneier, "Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure," *Computer Security Journal*, vol. 16, no. 1, pp. 1-7, 2000.

6. Levi A., ve C. K. Koc, "Risks in email security," *Communications of the ACM*, vol. 44 no. 8, pp.112, Ağustos 2001.

7. Wheeler A., ve L. Wheeler, *Payment, Security & Internet References*, http://www.garlic.com/~lynn/

8. Levi A., ve C. K. Koc, "CONSEPP: Convenient and secure electronic payment protocol based on X9.59," *Proceedings, The 17th Annual Computer Security Applications Conference*, pp. New Orleans, Louisiana, IEEE Computer Society Press, Los Alamitos, California, Aralık 10-14, 2001.

9. Freier A. O., P. Karlton, ve P. C. Kocher, *The SSL Protocol Version 3*, Netscape Communications Corp., 1996, http://home.netscape.com/eng/ssl3

10. Ramsdell, B., S/MIME Version 3 Certificate Handling, RFC 2632, Haziran 1999.