YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

IPV6 TABANLI AÇIK ANAHTAR KRİPTO SİSTEMİNDE SERTİFİKA YAŞAM DÖNGÜSÜ MEKANİZMALARI ANALİZİ VE TASARIMI

Elektronik ve Haberleşme Müh. Burak ÇALIŞKAN

FBE Elektronik ve Haberleşme Anabilim Dalı Haberleşme Programında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Metin YÜCEL

İSTANBUL , 2007

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

IPV6 TABANLI AÇIK ANAHTAR KRİPTO SİSTEMİNDE SERTİFİKA YAŞAM DÖNGÜSÜ MEKANİZMALARI ANALİZİ VE TASARIMI

Elektronik ve Haberleşme Müh. Burak ÇALIŞKAN

FBE Elektronik ve Haberleşme Anabilim Dalı Haberleşme Programında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Metin YÜCEL

İSTANBUL , 2007

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

OU	: Organizasyonel Birim
O	: Organizasyon
C	: Şehir
SM_{MA}	: Sertifika Makamı No : A
C	: Şehir

KISALTMA LİSTESİ

AA : Açık Anahtar
AAA : Açık Anahtar Altyapısı
API : Application Programming Interface
CRL : Certificate Revocation List
CMC : Certificate Management Protocol Using CMS
CMP : Certificate Management Protocol
CMS : Cryptographic Message Syntax
COH : Certificates of Health
CRMF : Certificate Request Message Format
D-H : Diffie Hellman
DN : Distinguished Name
DNS : Domain Name System
DPV : Delegated Path Validation
DPD : Delegated Path Discovery
EURO6IX : European IPv6 Internet Exchanges Backbone
FQDN : Fully Qualified Domain Name
FER : Full Enrollment Request
HTTPS : Hypertext Transfer Protocol over Secure Socket Layer
IETF : Internet Engineering Task Force
IPSEC : Internet Protocol Security
ISP : Internet Service Provider
IX : Internet Exchange Point
KEYGEN : Key Generator
LDAP : Lightweight Directory Access Protocol
OCSP : Online Certificate Status Protocol
OID : Object Identifier
OU : Organizasyonel Birim
P2P : Peer to Peer
PKCS : Public Key Cryptography Standards
PKIX : Public-Key Infrastructure X.509
POP : Post Office Protocol
SCP : Secure Copy
SCEP : Simple Certificate Enrollment Protocol
SCVP : Simple Certificate Validation Protocol
SMMA : Sertifika Makamı No A
SQN : Sequence
SER : Simple Enrollment Request
S/MIME : Secure/Multipurpose Internet Mail Extensions
SSH : Secure Shell
SSL : Secure Sockets Layer
TLS : Transport Layer Security
TSP : Time Stamp Protocol
VPN : Virtual Private Network

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

ÖNSÖZ

Yüksek lisans eğitimim süresince, engin bilgi ve tecrübelerini esirgemeyen çok değerli hocam Prof. Metin Yücel'e teşekkür ederim. Tezimin hazırlanma aşamasında büyük destek ve sabır gösteren sevgili eşim Nesibe'ye, başta babam Alparslan Çalışkan olmak üzere tüm aile fertlerime, maddi - manevi desteklerinden dolayı Kayalar Şirketler Topluluğu'na, çalışmakta olduğum Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü yönetimine ve birlikte çalışma imkanı bulduğum tüm arkadaşlarıma da ayrıca teşekkür ederim.

ÖZET

Çağımızda, bilişim teknolojisi endüstrisinin odak noktasını, güvenlik unsuru oluşturmaktadır. İnternet tabanlı teknolojilerin gelişiminin devasa boyutlara ulaşması ve bu teknolojinin etken olduğu yeni iş geliştirme yöntemlerinin piyasalara girmesiyle organizasyonlar, rekabetçi ortamın baskısı altında paydaşlarına web tabanlı hizmetler sunmanın getireceği risklere yönelik çözümler araştırmaya yoğunlaşmıştır. Bu amaçla günümüzde, güvenli veri iletişimine yönelik geliştirilen şifreleme mekanizmaları ya da yetkilendirme amaçlı kullanıcı paylaşımları gibi birçok yöntem, yaygın bir şekilde uygulanmaktadır. Ancak geliştirilen bu sistemlerin yetersizlikleri ortaya çıktıkça, güvenliğin daha yüksek seviyeli bir perspektif öngörülerek tasarlanmasının gerekliliği açığa çıkmaktadır. Kavramsal olarak güvenlik unsurları incelendiğinde, erişim kontrolü, veri gizliliği, veri bütünlüğü, anahtar yönetimi, yetkilendirme, inkar edememe ve kullanılabilirlik hususlarını, bir bütün olarak ele alıp, sistemsel olarak uygulamaya geçiren çözümlerin başında, açık anahtar altyapıları gelmektedir. Açık anahtar altyapısı, içerdiği protokoller, servisler ve standartlarla, belirtilen güvenlik unsurlarını içeren yöntemleri, açık anahtar kodlama ve sayısal imza mekanizmalarıyla sunarak, üzerinde diğer uygulamalarının ve ağ güvenlik bileşenlerinin kurulabileceği güçlü ve güvenilir bir yapı oluşturur.

İspanya'nın Murcia Üniversitesi'nde geliştirilmesi süren ve UMU-PKI olarak adlandırılan Java tabanlı açık anahtar altyapısı, IPv6 destekli X.509 sertifikasyon servislerinin, açık anahtar altyapısına bağlı kalarak uygulanabilmesini amaçlayan bir sistem modeli oluşturmaktadır. Yeni nesil internet protokolü olarak da adlandırılan ve güncel internet protokol sürümünün yetersizliklerini ve problemlerini gidermeyi hedefleyen IPv6 protokolü temelli açık anahtar altyapısı çalışmaları, günümüzün ve özellikle geleceğin internetinin, güçlü, güvenilir ve geliştirilebilir bir yapıya sahip olabilmesi açısından kritik öneme sahiptir. Bu çalışma içerisinde, geliştirilmekte olan UMU-PKI altyapısı detaylandırılarak incelenmiş, sertifika yaşam döngüsü içerisinde yer alan yönetim ve doğrulama işlevlerinin sağlanmasına yönelik güncel mekanizmalardan CMC ve SCVP protokollerinin sisteme entegrasyonu önerilerek, sistem modelleri tasarlanmış ve gerçekleme yapılmıştır. Önerilen çözüm ile IPv6 için tasarlanan alt yapı, daha güvenli hale getirilmiştir.

Anahtar kelimeler: IPv6, PKI, Açık Anahtarlamalı Kriptosistemler, UMU-PKI, Sertifika Yönetimi, Sertifika Doğrulama, CMC, SCVP

ABSTRACT

Security, has become a primary focus, througout the entire information technology industry. With the massive advent of the internet enabled businesses, it has become critical for todays companies to ensure security concerns, along with the market pressure, to be present on the World Wide Web. For this purpose, several methods are widely deployed today such as the data encryption or password based authentication for conforming secure data transmissions. Nevertheless, accompanied by the emerging weaknesses of these types of developments, the necessity of designing security in a higher level prospect has come out. In essence, security related access control, data confidentiality, data integrity, key management, authentication, non-repudation or availability issues are supported principally by public key infrastructure in a conceptual manner. In conjunction with the protocols, services and standards, public key infrastructure adresses these topics by methods of public key cryptography and digital signing instruments, allowing solid and secure framework for assembling other various applications or network security mechanisms.

JAVA based PKI with IPv6 support called UMU-PKI is principally aiming to build a system model for integrating IPv6 based X.509 certification services within the public key infrastructure. The efforts on the public key infrastructure based on the new generation internet protocol called IPv6, which contains solutions for the basic drawbacks and problems of current IPv4 protocol, will play a critical role in constructing the solid and reliable future of internet. In this study, UMU-PKI v6 system infrastructure and architecture concerns are examined in detail, certificate lifecycle management and certificate validation processes which should take a vital role in the core part of the whole system processes are deeply studied. CMC and SCVP protocols which are still in progress, are proposed to integrate to the system in order to provide concrete and secure mechanisms on both certificate lifecycle management and validation procedures. Along with this study, operational system integration models are designed with the application based integration schemes including the deep analysis of the two newly configured certificate based protocols. The security structure of the system has been improved via proposed solution.

Keywords: IPv6, PKI, Public Key, Cryptography, UMU-PKI, Certificate Management, Certificate Validation, CMC, SCVP

1. GİRİŞ

İnternet erişim protokollerinin geçerli sürümü olan IPv4, yayınlanmış olduğu 1981 yılından günümüze kadar önemli ölçüde değişikliğe uğramamıştır. Protokolün kolay uygulanabilirliği, ortak çalışmayı destekleyebilmesi ve güçlü yapısı gibi özellikleriyle, günümüz interneti çapında, ağlar arası bir ağa hizmet etme sınavından başarıyla geçmiştir. Ancak protokolün bu ilk sürümü, tasarım bazında bazı konularda öngörü eksikliklerini de bünyesinde barındırmaktadır. Özellikle son yıllarda internet ağı boyutlarının katlanarak genişlemesi, adres alanının neredeyse yetersiz hale gelmesi, ağın omurgasını oluşturan kısımdaki yönlendiricilerin, büyük yönlendirme tablolarını saklamaları, IP seviyesinde güvenlik mekanizmalarına gereksinimin artması ve daha sade bir yapılandırmaya duyulan ihtiyaç gibi birçok nedenle, protokolün yenilenmesi ve geliştirilmesine yönelik çalışmalar hız kazanmıştır. [Huitema, C. 1998] Bu tip sorunlara çözüm bulmak amacıyla IETF grubu, sürüm 6 olarak nitelendirilen yeni bir iletişim kuralları ve standartları takımı geliştirmiştir. Yeni sürüm internet erişim protokolü, IPv6 olarak adlandırılmış olup, getirmiş olduğu yeniliklerle, günümüz ve geleceğin internetinin daha güçlü ve güvenilir bir yapıya sahip olmasını hedeflemektedir. IPv6 ile birlikte gelen yeniliklerin başında, kapsamış olduğu geniş adres alanı sayesinde internet omurga tasarımcılarının esnek hiyerarşik yapıları kurabilmelerine olanak sağlaması, hiyerarşik ve verimli adres konfigurasyonlarının gerçekleştirilebilmesi, daha yüksek servis kalitesi ve güvenlik mekanizmaları, basit ve genişleyebilir paket formatı, entegre mobil IP protokolü ve multicast desteği gelmektedir. [Deering, S. and R. Hinden,1998]

Günümüz internet tabanlı teknolojilerde şirketler ve bireyler, hemen her gün farklı ölçeklerde binlerce işlem gerçekleştirmektedir. Personeller, dosyalarını ve gizlilik içeren bilgilerini, e-posta yoluyla ya da kullandıkları ağ üzerinden paylaşmakta, banka müşterileri, evlerindeki bilgisayarlardan, hesaplarıyla ilgili işlemleri gerçekleştirmekte ve türlü ürünler, online olarak doldurulan formlar aracılığıyla sipariş edilebilmektedir. İnternetin, günlük yaşantı içerisinde bir iş aracı olarak telefon ve fax makinelerinin yerine geçmesi yönünde çok güçlü bir potansiyelin olduğu aşikardır. Aynı zamanda internet, yanlış ellerin kullanımında, hassas ve kritik öneme sahip bilgilerin çalınması, kişilere ya da kurumlara özel bilgilerin ve mesajların ele geçmesi gibi tehdit mekanizmalarına da müsait bir ortam sağlamaktadır. Karşılıklı görüşmelerde, taraflar arasında sıkı bir güven ilişkisi hakim olabiliyorken, bu durum interneti kullanan organizasyonların sayılarının her geçen gün devasa miktarlara ulaşmasıyla, yerini farklı türden güven mekanizmalarının inşaa edilmesi gerekliliğine bırakmıştır.

İnternet ve benzeri açık ağlarda, bilgi güvenliğinin sağlanabilmesi için, geleneksel ticaret yöntemlerinin vazgeçilmez bir parçası olan güven unsurunun, ağ iletişiminde de sağlanmasına yönelik olarak sayısal şifreleme ve imza teknolojilerinden yararlanılmaktadır. Bu teknolojilerde gönderici ve alıcının kimliklerinin doğrulanması, gönderilen iletinin üçüncü taraflarca edinilememesi ve içeriğinin değiştirilememesi sağlanmaktadır. Bu sayede uzun yıllar boyunca askeri ve diplomatik gereksinimlerle geliştirilen kriptoloji yöntemleri, sivil iletişimde de uygulanmaya başlanmıştır. Son yıllarda bu yöntemlerden öne çıkan ve dağıtık platformlarda sıklıkla tercih edilen kripto mekanizması, açık anahtar altyapısı temelli teknolojilerdir. Bu noktada geleceğin internetinin sağlam bir yapıya sahip olmasında, yeni nesil yönlendirme protokolü olan IPv6 ile birlikte gelen teknolojileri destekleyecek, bu protokole ait mekanizmaları güvenli ve güvenilir halde sunacak açık anahtar altyapılarının geliştirilmesi çalışmaları, oldukça büyük öneme sahiptir.

Bir açık anahtar altyapısı, organizasyonların kullanmakta oldukları internet ya da intranet tabanlı ağ sistemleri içerisinde, uygulamak istedikleri güvenlik politikaları gereğince, güven altyapılarını esnek ve geliştirilebilir yöntemlerle oluşturabilmelerine imkan sağlar. Açık anahtar altyapılarının temel amacı, orta veya büyük ölçekli organizasyonel sistemler içerisinde, açık anahtar kodlama yöntemlerinin, ilgili mekanizmalar ve işlevsel birimler aracılığıyla, efektif olarak kullanımını sağlamaktır. Bu yapı içerisindeki temel birimler, Sertifika Makamı, Kayıt Makamı ve Dizin Sunumcu olup, ek olarak uygulanacak servislere göre değişiklik gösteren, akıllı kartlar, zamanlayıcı sunumcular ve OCSP sunumcular sistem içerisinde kendilerine yer bulabilir.

1.1 IPv6 Tabanlı Açık Anahtar Altyapısı

Açık anahtar altyapısı, sertifika talep, yenileme, indirme ve iptal işlemleri gibi farklı servisleri, organizasyon içerisindeki kullanıcılarına web arayüzleri aracılığıyla erişilebilecek şekilde sunar. IPv6 tabanlı açık anahtar altyapısı ise, temelde aynı servisleri, bir IPv6 ağı üzerinden kullanıcılarına sağlamayı hedefler. Sistem içerisindeki kullanıcılar, mekanizmalar ve cihazlar, bu servisler aracılığıyla, sayısal olarak imzalanmış bilgileri kodlayabilirler. Bir IPv6 ağı içerisinde, açık anahtar sertifikalarını kullanarak, sistemdeki bilgileri korumak veya sistem kullanıcılarını yetkilendirmek amacıyla, HTTPS benzeri web sunumcuları içeren servisler, VPN cihazları, yetkilendirme, otorizasyon ve hesaplama servisleri yer alabilir. Ayrıca, bir HTTPS web sunumcu yetkilendirmesinin sertifikalanması gibi farklı senaryolar içerisinde, IPv6 adreslerini veya DNS adlarını, alternatif özellik bilgileri olarak da

kullanabilirler. Bu servis, temelde IPv6 tarafından sunulmakta olup, sayısal sertifikaların IPv6 adreslerini içerebilmelerini sağlamaktadır. Bu ve benzeri işlevler, IPv6 ile birlikte gelen yeni teknolojilerle birlikte, açık anahtar altyapısı içerisinde düşünülmesi gereken öğelerdir. IPv6 tabanlı geliştirilen yeni teknolojiler, bu tip bir altyapıyı kullanacak sistemler açısından, geniş ölçeklerde beraberinde türlü güvenlik sıkıntılarını ve gereksinimlerini de ortaya çıkaracaktır. Sistemlerin bu altyapı dahilinde yaygın bir şekilde uygulanabilmesi için öncelikli olarak, bu güvenlik problemlerinin ve ihtiyaçlarının belirlenerek çözümlenmesi ya da aynı altyapıyı sağlayacak farklı güvenlik mekanizmalarının sistem içerisine dahil edilmesi gerekmektedir. Burada öne çıkan başlıklar, temelde günümüz organizasyonlarında oldukça yoğun bir şekilde uygulanmakta olan, açık anahtar temelli güvenlik ve gizlilik mekanizmalarının sistemlere entegrasyonu altında yer almaktadır. Bu nedenle IPv6 tabanlı bir açık anahtar altyapısı üzerinde çalışmaların yoğunlaşması, gelecek internet temelli yapıların güvenliğinin sağlanmasında ve benzer şekilde IPv6 temelli altyapıların yaygınlaştırılmasında büyük öneme sahiptir.

1.2 UMU-PKI v6

UMU-PKI v6 ya da diğer bir tanımıyla, Java IPv6 PKI altyapısı, IPv6 destekli X.509 sertifikasyon servislerinin tasarımını ve geliştirilmesini amaçlar. Bu altyapı, herhangi bir organizasyon tarafından, kullanıcılara ait haberleşme işlevlerini güvenli hale getirmek için, farklı seviyelerde açık anahtar mekanizmaları sunar. Bu güvenlik entegrasyonunu kullanan istemciler, sertifika talebi, sertifikaların yenilenmesi veya iptal edilmesi gibi temel sertifika işlevlerini, kullanmakta oldukları web arayüzleri aracılığıyla gerçekleştirebilirler. Aynı zamanda kullanıcılar, kendilerine opsiyonel olarak sağlanabilecek akıllı kartları kullanarak, kriptografik bilgilerinin saklanmasında, mobilitelerini ve sistem güvenilirliğini arttırabilirler. UMU-PKI, dağıtık organizasyonlara ait uygulamaların ve kişisel gizlilik seviyesine sahip bilgilerin korunması ve saklanmasında, açık anahtar altyapısını, IPv6 tabanlı yeni teknolojileri de destekleyecek ve gelişimlerini kolaylaştıracak biçimde sağlayarak, sistem bünyesindeki farklı işlevsel mekanizmalara sahip birçok fonksiyonel birim ve uygulamalarla, ciddi bir güvenlik ve kullanıcı gizliliği platformu ortaya koymaktadır. Euro6IX projesi, IPv6 ağının entegrasyonuna yönelik, günümüzdeki en kapsamlı çalışmadır. Bu ağ aynı zamanda, UMU-PKI v6 modelinin testleri ve hali hazırdaki servislerin, IPv4'ten IPv6'ya aktarımının modellenebileceği en uygun ortamı oluşturmaktadır. Proje kapsamında, kullanıcılarla ya da ISP ve IX'lerle, gerçek IPv6 haberleşmesi mümkün olabilmektedir. UMU-PKI sistemine yönelik geliştirilmekte olan uygulamaların testleri de bu ağ içerisinden sağlanmaktadır.

1.3 Sertifika Yönetimi ve Yaşam Döngüsü

Açık anahtar altyapısı sistemlerinde, kullanıcılar, sertifika makamları tarafından kendi kimlikleriyle ilişkilendirilmiş olan sertifikalara sahiplerdir. Kullanıcıya ait açık anahtar bilgisi, kullanıcı isim bilgileri ve e-posta adresi gibi bilgiler, sertifika içerisinde yer almaktadır. Sertifikalar, sistemde imzalanarak oluşturulurlar ve erişime açık bir sertifika deposunda saklanırlar. UMU-PKI sistemi için de kavramlar benzerdir. Sistem içerisindeki makamlar tarafından imzalanarak oluşturulan sertifikalar, sistem kullanıcılarının açık anahtar altyapısı ile ilgili hemen her işleminde, yetkilendirme ya da farklı bir amaçla kullanılabilirler. Sertifikaların sistem içerisindeki işlevleri, sertifika yaşam döngüsü adı verilen bir mekanizmalar topluluğu olarak tanımlanır. Sertifika yaşam döngüsü, sistemde sertifikalarla ilişkili olarak gerçekleşen tüm fonksiyonları içerir. Sertifika yaşam döngüsü içerisinde yer alan, sertifikaların üretilmesi, sistem içerisinde dağıtımı, sistemde kullanım sürelerinin sonlandırılması ya da geçersiz hale getirilmeleri, sistem içerisinde yetkilendirilmesi, askıdaki sertifika taleplerinin durumlarının sorgulanması ve geçerlilik sürelerinin güncellenerek sistem içerisinde yenilenmesi işlevlerini içeren yönetim mekanizmalarının tümü, sertifika yönetimi kavramı içerisinde işletilirler. [Adams, C. ve Farrell, S. 2000]

Şekil 1.1 Sertifika yaşam döngüsü

Sertifika yaşam döngüsü içerisinde yer alan işlevler, Şekil 1.1'de tasarlanmış gösterime paralel olarak üç ana aşama altında özetlenebilir. Bu aşamalar, sertifikaların talep edilmeleri gibi, sertifikalarla ilgili işlemlerin, sistem içerisindeki başlangıç gereklerini içeren Başlangıç Aşaması, sertifikaların yayınlanmasının ardından oluşabilecek süreçleri içeren Yayın Aşaması ve son olarak da sertifikaların sistemden kaldırılması ile sonlanan süreci içeren Son Aşama olarak belirtilmiştir. Sertifika yönetimi, bu bağlamda, bir sertifika yaşam döngüsü içerisindeki

tüm mekanizmaları kapsayan, öngören, yöneten ve işleten bir sistem modeline sahip olmalıdır. Bu sistem modeli, sertifikaların istemciler tarafından talep edilmesi sürecinden başlayarak, sistemde geçersiz hale getirilmelerine kadar geçen süreçler içerisinde, fonksiyonel amaçları doğrultusunda, sertifika kullanıcılarına hizmet veren bir yapıyı kapsar. Bu yapı içerisinde, sertifika talep, yenileme, edinme, oluşturma, durum sorgulama, iptal gibi farklı işletim süreçlerine ait uygulamaların, sistem içerisinde gerçekleştirilebilmesi gerekir. Sistem içerisinde yapılandırılacak olan sertifika yaşam döngüsü mekanizmaları, açık anahtar altyapısının, üstlenmiş olduğu işlevleri doğru bir biçimde gerçekleştirebilmesine olanak sağlayan kritik yapılardan birisidir.

1.4 Sertifika Doğrulama

Sistemdeki sertifikaların yaşam döngüsünde yer alan işlevlerin dışında, gerek istemciler, gerekse diğer sertifika birimleri tarafından, sertifikaların bir güven ortamı içerisinde işletilmesi zorunluluğu bulunmaktadır. Buradaki güven ibaresi, sertifikaların oluşturularak imzalandığı üst makamın güvenilir olmasından çok, sistemde oluşturulan sertifikaların, doğru sertifika yolu yapılandırmalarıyla oluşturulup oluşturulmadığına yönelik sağlanan bilgiyi ifade eder. Sistemdeki bir birimle ilişkili bir sertifika yayınlandığında, bu sertifikanın sistemde işlevsel olarak kullanılabilmesi için, sertifika sahibi birimin, sertifika yayınlayan makama güven duyması gerekir. Bu güven, ilgili birimin, sertifika taleplerini değerlendirirken, sertifika makamının tutarlı bir politikaya bağlı olduğuna ve bu ilkelere uymayan birimlere sistem içerisinde sertifika vermeyeceğine, emin olduğunu doğrular. Buna ek olarak sertifika dağıtıcısının, geçerliliğini yitiren sertifikaları iptal ederek sistemde yayınlayacağına güven duyduğuna dair bir bilgiyi de içermektedir. Aynı zamanda sertifikayla ilgili Kök Sertifika Makamı'nın, güvenilir Sertifika Makamları listesinde yer aldığı ve oluşturulmuş olan sertifika yolundaki makamların sertifikalarının da geçerli olduğu anlamına gelmektedir. Sertifika yolu, organizasyon yapısı ya da açık anahtar altyapısı hiyerarşisi içerisinde, alt sırada bulunan bir güvenilir Sertifika Makamı'ndan, Kök Sertifika Makamı'na kadar tüm sertifika sıra düzenindeki her bir makama ait sertifikaları içerir. [Dzambasow, Y. et. al, 2005] Örneğin Kök Sertifika Makamı için sertifika yolu, kendi tarafından imzalanarak oluşturulmuş tek bir sertifika olarak oluşurken, hiyerarşik yapı içerisinde Kök Sertifika Makamı'nın hemen altında yer alan bir Sertifika Makamı'na ait sertifika yolu ise, kendi sertifikası ve Kök Sertifika Makamı'na ait sertifika olmak üzere iki adet sertifikayı içerecektir.

Sertifika yolunun bu tanımı altında sertifika doğrulama işlevi, bir sertifika yolu üzerinde bulunan tüm sertifikaların geçerliliğinin kontrolü olarak ifade edilebilir. Bir başka ifadeyle sertifika doğrulama, ilgili sertifikadan Kök Sertifika Makamı'na giden yol üzerindeki sertifikaların geçerliliğinin doğrulanması işlemidir. [Housley, R. et. al, 2002] RFC 3280 dokümanında da belirtilmiş olan bu tanım çerçevesinde, sertifika yönetim mekanizmaları göz önünde bulundurulduğunda, sertifika yönetimi ve doğrulama işlevlerinin, sistem içerisinde bir bütün halinde değerlendilerek oluşturulması ve doğru bir tasarımla yapılandırılması, açık anahtar altyapılarına ait süreçler açısından kritik değer taşımaktadır. Bu mekanizmalar birbirlerine bağımlı olup, gerek sertifika yönetiminde, gerekse doğrulama mekanizmalarında oluşabilecek muhtemel bir sorun ya da eksiklik, açık anahtar altyapısının ve üzerindeki sistemlerin güvenilirliği ile ilgili sistem bünyesinde soru işaretlerine sebep olabilir.

1.5 UMU-PKI Sertifika Yaşam Döngüsü Mekanizmaları

Bir açık anahtar altyapısı içerisindeki sertifikaların yönetilmesi unsuru, sertifika yaşam döngüsü mekanizmalarının, ilgili sisteme entegrasyonu ile sağlanmaktadır. Benzer şekilde sistem güvenlik altyapısı içerisindeki bir diğer önemli unsur olan, sertifika doğrulama işlevleri ise, sistemde uygulanacak sertifika yapılarının, doğru ve geçerli sertifika yolları ile oluşturulduğunu garanti eden mekanizmalar bütünü olarak değerlendirilebilir. IPv6 temelli X.509 sertifika işlevleri desteğinin sağlanmasında, kritik öneme sahip mekanizmaların başında, sistemde yer alacak sertifika tabanlı servislerin yönetimi ve sertifika yapılarının doğru bir şekilde oluşturulması gelmektedir. Sistemde sağlanacak sertifika yönetim ve doğrulama işlevlerinin, genel kabul görmüş metodolojilere sahip olması gerekliliği, sistemin dağıtık platformlara entegrasyonunda oldukça önemlidir. Bu çalışma içerisinde, UMU-PKI altyapısına ait sistem modelinde yer alması gereken sertifika yönetim ve doğrulama mekanizmalarının, sistemsel süreçler ve teknik gereksinimler göz önünde bulundurularak tasarımları, süreç bazlı akış mimarilerinin ve uygulama tabanlı entegrasyon modellerinin gerçekleştirilerek, bu tasarımlar sonucunda, sistemsel olarak edinilecek kazançların ortaya konması hedeflenmiştir.

IETF PKIX çalışma grubu, açık anahtar altyapılarıyla ilişkili olarak, sertifika yönetim protokolleri tanımlamıştır. Bu protokoller genelde X.509 v3 açık anahtar sertifikalarına yönelik olup, bu protokollerden öne çıkanlar, PKIX CMP ve CMC protokolleridir. Bu iki protokol de, amaçları doğrulusunda paralellikler içerirken, CMC protokolünün sahip olduğu bazı yapısal özellikleri ve eklenen güncel servisleri itibariyle, CMP protokolüne kıyasla bir takım avantajlara sahiptir. Bu avantajlar, protokolün henüz geliştirilmekte olan eklentileriyle, günümüz teknolojilerindeki güncel yapıları kapsayabilmesi, UMU-PKI sisteminin sertifika yaşam döngüsündeki tüm süreçleri içerecek metodlara sahip olması, CMP protokolüne kıyasla daha hızlı entegre edilebilir, uygulama kodunun daha etkin bir şekilde geliştirilebilir olması ve bu sayede yaklaşık %10 oranında daha düşük kod yükü altında entegrasyonun sağlanabilmesi olarak özetlenebilir. [SSH Communications, 2002] CMS temelli sertifika yönetim mekanizmaları sunan CMC protokolü, bir açık anahtar sisteminde ihtiyaç duyulan açık anahtar servislerine yönelik arayüz ihtiyacını gidermek amacıyla tasarlanmıştır. [Myers, M ve Liu, X. et. al, 2000]

Sertifika yolunun oluşturularak, sertifikaların doğrulanmasına yönelik günümüz gelişen teknolojilerinin başında SCVP protokolü temelli çalışmalar öne çıkmaktadır. SCVP, sertifika yolu yapılandırma veya doğrulama işlevlerini bir sunumcuya devrederek, organizasyon içerisindeki doğrulama politikalarını, merkezi olarak yönetebilme ve bu sayede açık anahtar tabanlı uygulamaların entegrasyonunu kolaylaştırma imkanını sağlar. [Malpani, A. ve Turner, S. 2000] Sertifika yolu doğrulama, örneğin sertifika yolundaki sertifikaların iptal edilmiş olup olmadığı bilgisinin alınması, bir ya da birden fazla güven bağlantı noktasına sahip bir doğrulama politikasına uygun şekilde gerçekleştirilir.

UMU-PKI sisteminin devam eden çalışmaları göz önünde bulundurulduğunda, sertifika yönetimi ve sertifika yolu doğrulama süreçlerinde ihtiyaç duyulan mekanizmaların, sistemin tüm birimlerince uygulanabilir ve kolaylıkla entegre edilebilir bir yapıya sahip olma zorunluluğu bulunmaktadır. Bu iki temel unsurun sisteme sorunsuz bir şekilde eklenebilmesinde ve sistemin ilişkili olduğu birimlerce desteklenebilir yapıda olmasında, uluslararası platformlarda kabul gören servislerin sistem içerisine entegrasyonu üzerinde çalışmaların gerçekleştirilmesi gerekmektedir. Bu noktada önerilen yöntem, UMU-PKI sisteminin, sonraki bölümlerde detaylı olarak incelenecek olan CMC sertifika yönetim ve SCVP sertifika doğrulama protokollerini kapsayacak şekilde ya da sistemin genelinde, bu iki protokole ait servislerin dahil edilmesini sağlayacak bir entegrasyon projesinin tasarımı ve uygulama temelli modellenmesi olacaktır. Bu çalışma içerisinde yer verilecek olan konular,

bu tasarım temelli geliştirilmiş olup, bu iki protokole ait servislerin ve mekanizmaların, UMU-PKI sistemi içerisinde tasarım ve uygulama tabanlı modellenmesinin gerçekleştirilmesiyle sonlanmıştır.

Çalışma içerisinde öncelikli olarak açık anahtar altyapıları detaylandırılarak, günümüz teknolojilerindeki uygulamalardaki güvenlik mimarileri içerisinde, sahip oldukları bileşenlerin ve sundukları mekanizmaların analizi gerçekleştirilmiştir. Açık anahtar altyapılarının, özellikle çalışmanın sonraki aşamalarında sıklıkla yer alacak öğeleri ve servisleri, detaylı olarak incelenerek, bu servislerin altyapıya katkıları ve uygulama yöntemleri üzerinde durulmuştur. Açık anahtar altyapıları ile ilgili bölümün ardından, UMU-PKI v6 sistemine geçiş yapılarak, UMU-PKI modelinin sistem mimarisi, bileşenleri, sunmayı amaçladığı servisleri ve çalışma içerisinde tasarlanan mekanizmaları, bu bölümde detaylı olarak incelenmiş, henüz üzerinde, gereken analizler gerçekleştirilmemiş olan, geleceğin IPv6 tabanlı ilk açık anahtar altyapısı, çalışmanın ikinci bölümü içerisinde kapsamlı olarak ele alınmıştır. Üçüncü bölümde, henüz geliştirme çalışmaları devam etmekte olan CMC ve SCVP protokollerinin detaylı analizleri gerçekleştirilmiştir. Bu analiz çalışması içerisinde, öncelikli olarak, protokollerin tanımlarının refere edildiği dokümanlardan ve ilgili diğer referanslardan edinilirek analiz içerisinde geliştirilen, protokollere ait temel yapılar ve protokol bileşenleri üzerinde durulmuştur. Protokollerin tanımlarının ve amaçlarının incelenmelerinin sonrasında, protokolü oluşturan bileşenler, önem ve anlam sıralamasına uygun şekilde hiyerarşik bir düzende aktarılmıştır. Protokol bileşenlerinden, sonraki bölümlerde gerçekleştirilecek entegrasyon tasarımlarında öne çıkacak başlıklar, detaylı olarak, nesne yapıları ve bu yapıların uygulama tabanlı kullanım yöntemleriyle birlikte ilgili bölümler içerisinde aktarılmıştır.

Protokollerin detaylandırılarak, teorik altyapıları ve yapısal bileşenleriyle birlikte, uygulamaya yönelik nesne tabanlı niteliklerinin incelenmesinin ardından, UMU-PKI sistemine ait sertifika yönetim ve doğrulama mekanizmaları altında yer alan süreçlerin tasarımı ve modellenmesi aşamasına geçilmiştir. Bu aşamada UMU-PKI sistemine ait servisler öngörülerek, öncelikle sertifika yönetimi ve sonrasında sertifika doğrulama mekanizmalarına ait süreçler, ilgili protokollerinin gerekleri dikkate alınarak, operasyonel olarak tasarlanmıştır. Gerçekleştirilmiş olan süreç tasarımlarında, günümüz tasarım modellemelerinde sıkça kullanılmakta olan, akış diyagram modellerinden yararlanılarak gösterimler gerçekleştirilmiş olup, süreçteki her bir akış birimi için gereken ek açıklamalar da model üzerinde sunulmuştur. Doğrulama mekanizmalarının entegrasyonunda, bu aşamalara ek olarak, UMU-PKI sistemi içerisindeki sertifika doğrulama işlevlerini sağlayacak

doğrulama servisinin, süreç ve akış bazlı tasarımları gerçekleştirilmiş ve servisin çalışma mantığı, akış modeli ve uygulama yönlü tasarımı, bu aşamadaki çalışmalar içerisinde sunulmuştur. Süreç tasarımları, sonraki aşamalarda gerçekleştirilecek olan, protokollerin uygulama tabanlı entegrasyon modellerinin tasarımlarına bir altyapı oluşturmuş olup, bu tasarımlar baz alınarak, uygulamaya dönük modellemeler gerçekleştirilmiştir. Uygulama temelli entegrasyon tasarımları, önceki aşamada hazırlanan operasyonel süreç modelini temel alarak, günümüz uygulama tabanlı teknolojilerde, özellikle JAVA tabanlı sınıflar ve sertifika mekanizmalarının, ilgili süreçte yer alan işlevi sağlayabilecek şartlarda tasarlanmalarını hedef alarak oluşturulmuştur. Sunulan tasarımlarda, operasyonel süreç tasarımlarına benzer şekilde, akış diyagram modellerinden faydalanılmış, ancak bu tasarımlarda, süreçler yerine nesne ve metodların ilişkisel olarak uygulamaya yansıtılmasına yönelik modeller yer almıştır.

Çalışmanın sonunda, UMU-PKI sisteminin günümüz açık anahtar tabanlı güvenlik mimarileri içerisinde, IPv6 desteğini içermesinin yanı sıra sağladığı avantajlar, UMU-PKI sistemi mekanizmaları içerisinde sertifika yaşam döngüsü ve doğrulama işlevlerinin sağlanmasına yönelik çalışma içerisinde tasarlanan CMC ve SCVP protokolleri bazlı süreç ve uygulama entegrasyon mimarilerinin sisteme katkıları, neden bu protokollerin çalışma içerisinde tercih edildikleri ve UMU-PKI sisteminin bu entegrasyonlar sonucunda kazanacağı avantajlar özetlenerek, bu sistemle ilgili gerçekleştirilebilecek geliştirme çalışmaları hakkında öneriler ve gelecek çalışma konuları sunularak çalışma sonlandırılmıştır.

1.6 Gerekli IETF Protokolleri

Günümüzde birçoğu yaygın bir şekilde sistemlerde uygulanmakta olan, sistemle ilişkili IETF protokolleriyle ilgili olarak protokollerin öngördüğü mekanizmalar ve uygulama amaçları hatırlatılmıştır.

- SSL : Bu güvenlik protokolü, internet üzerindeki haberleşmelerde gizliliği sağlar. Protokol aracılığıyla, istemci-sunumcu etkileşimli uygulamalar, farklı güvenlik tehditlerine karşı güvenli bir şekilde haberleşme imkanına sahip olurlar.
- SSH : Ağ içerisindeki bir sunumcuya, uzakta bulunan bir makineden bağlantıyı mümkün kılar. Güvensiz makineler arasındaki iletişim güçlü bir kriptografi ile şifrelenir.
- LDAP : Online dizin servislerine erişimde kullanılan bir protokoldür. TCP'nin üzerinde çalışmakta olup, LDAP ya da X.500 temelli bir dizin servisine erişimi sağlar. [Boeyen, S. et.al, 1998]

- SCEP : Ağ cihazlarının sertifikalanmasının, güvenli bir şekilde sağlanmasında görev alır. Sertifika Makamı veya Kayıt Makamı tarafından yürütülen açık anahtarların dağıtımı, sertifika iptal ve durum sorgulama operasyonlarını destekler.
- OCSP : Sistemdeki birimlerin, sertifikaların güncel durumlarını tespit edebilmelerini sağlar. Burada belirtilen durum bilgisi, genelde, sertifikaların iptal olup olmadığı, yani geçerliliğine ait bilgidir. [Myers, M. et.al, 2000]
- TSP : Bu protokol, bir sertifika işleminin gerçekleştirilmesinde, belirli bir zaman aşımı süresinin, bir dış uygulama tarafından tespit edilebilmesini sağlar.
- S/MIME : Güvenli bir şekilde MIME bilgilerinin gönderilmesi ve alınmasını sağlar. MIME standardını temel alan S/MIME, elektronik mesajlaşma uygulamalarındaki yetkilendirme, mesaj bütünlüğü, kaynağın geri çevrilememesi, gizlilik ve bilgi güvenliği gibi servislerin, kriptografik güvenlik servisleri aracılığıyla gerçekleştirilebilmesini sağlar. [Ramsdell, B. 1999]
- IPSEC : IP katmanında gizlilik ve yetkilendirme servislerini sağlayan protokoldür. TCP/IP iletişiminde verilerin şifrelenerek iletimini öngörerek güçlü bir şifreleme sağlarken aynı zamanda iletilen verinin bütünlüğünü de garanti eder.

2. Açık Anahtar Altyapısı ve UMU-PKI Mimarisi

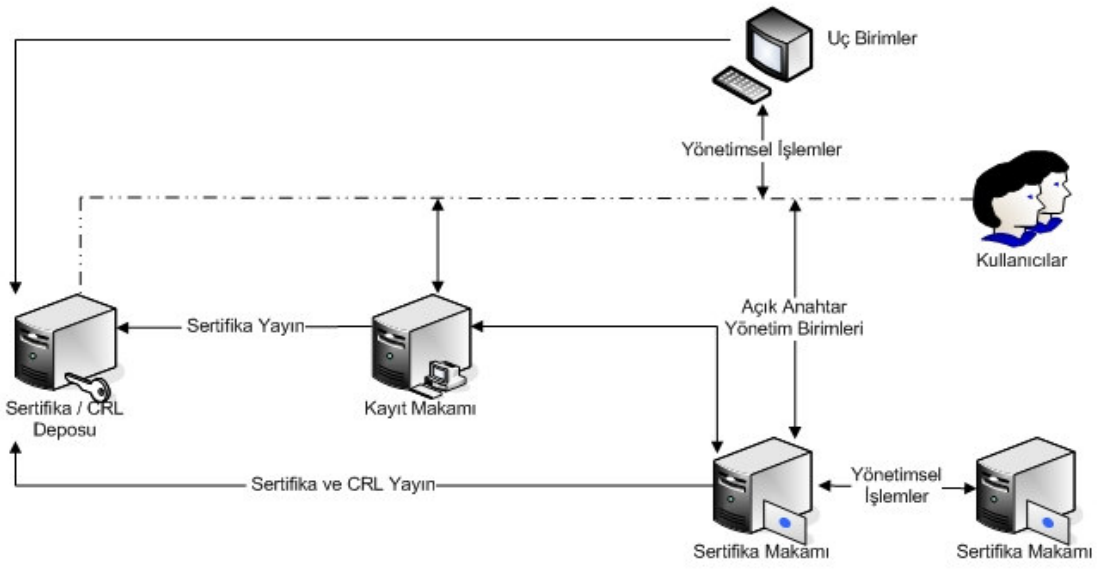
Çalışmanın bu bölümünde, açık anahtar altyapısı çatısı altında UMU-PKI sistemi tanıtılmıştır. Açık anahtar altyapıları, günümüz güvenlik mimarilerinin, özellikle dağıtık organizasyonlarda temel yapı taşlarını oluşturmaktadır. Gerek tanımı itibariyle, üzerinde çalışacak sistemlere yönelik bir altyapı mimarisi olarak değerlendirilmesi, gerekse yapısı itibariyle destekleyebildiği farklı sistematik konfigurasyonel tasarımlarla açık anahtar altyapıları, bu bölüm içerisinde hem genel ortak bileşen ve nitelikleriyle, hem de çalışmanın sonraki aşamalarda gereksinim duyulacak konuların detaylandırılmalarıyla, güçlü bir analize tabi tutulmuştur. Açık anahtar altyapılarıyla ilgili gerçekleştirilen bu çalışmanın sonrasında, UMU-PKI sistemine ait yapılar ve servislere geçiş yapılmıştır. Temelde bir açık anahtar altyapısı mimarisi öngören bu sistemle ilgili olarak bu bölümde sistem mekanizmalarına geniş olarak yer verilmiştir.

Bölüm içerisinde ilk olarak açık anahtar altyapısı kavramı tanıtılarak, teknolojilerin gelişim tarihçesi içerisinde, bu tür yapıların geliştirilmesine ait kısa ve özet bir tarihçe sunulmuştur. Açık anahtar altyapılarının vazgeçilmez mekanizmalarından, asimetrik kodlama yöntemi hakkında bilgiler verildikten sonra bu yapıların temel taşlarından olan sertifikalar ve sertifika formatları üzerinde incelemeler sunulmuştur. Açık anahtar altyapısı ile ilgili temel bileşenlerin tanıtılmasının ardından, sertifika tabanlı açık anahtar işlevleri, önem sırasına göre incelenmiştir. Bu işlevlerden, çalışmanın sonraki bölümlerinde öne çıkacak olan servisler, detaylı bir analiz içerisinde çalışmada yer almıştır. Bu servislerin sonrasında UMU-PKI mimarisine geçiş yapılarak, temel sistem yapısı tanıtılmış ve sistem bileşenleri, gerek sahip oldukları mekanizmalar, gerekse sistem içerisindeki kullanım yöntemleri ele alınarak detaylı olarak aktarılmıştır. UMU-PKI sistemine ait temel servisler ve sistemin sahip olduğu ek servislerin sunulmasının ardından bu bölüm içerisindeki çalışma sonlandırılmıştır.

2.1 Açık Anahtar Altyapısı Kavramı

Bir açık anahtarlama altyapısı, genel olarak Internet ya da genel erişime açık mimariler içerisinde, açık anahtar sertifikalarının bir grup güvenlik protokolüne uygun biçimde yönetilmesini sağlar. Bu protokoller, IPSEC, SSL, TLS veya S/MIME açık anahtar kriptografisi ile kendilerine bağlı olan servislerde gizlilik, veri bütünlüğü, veri kaynağı otorizasyonu ve geri çevrilememe özelliklerini kazandırırlar. Bir açık anahtar altyapısı, organizasyon içerisinde tanımlı bir fonksiyonu sunmaktan öte, organizasyona ait diğer güvenlik servisleri için bir temel teşkil eder. Açık anahtar altyapısının temel işlevi, açık

anahtarların ve sayısal sertifikaların güvenlik ve bütünlük çatısı altında dağıtım ve kullanımının sağlanmasıdır. Açık anahtar altyapısının anlaşılabilmesi için öncelikle kodlama yönteminin anlaşılması gerekir. Açık anahtar kriptografisi ya da asimetrik kodlama, temelde bir anahtar çifti kullanır. Bunlardan ilki, anahtarın taşıyıcısına ait olan gizli anahtar olup ikincisiyse sistem dahilinde açık olarak uygulanabilen açık anahtardır. [Adams, C. ve Farrell, S. 2000] Açık anahtar sistemlerinin kullanıcıları, açık anahtar sertifikalarına güvenmek durumundadır. Sertifika, bir açık anahtarı, ilgili kullanıcıyla ilişkilendiren bir veri yapısını içermektedir. Bu bağlantı, kişinin kimliğini doğrulayarak sertifikayı imzalayacak, güvenilir bir Sertifika Makamı ile mümkün olmaktadır.



Şekil 2.1 Açık anahtar altyapısı genel işleyiş

Bir açık anahtar altyapısının temel bileşenleri :

- Sistem içerisinde açık anahtar sertifikalarını yayınlayacak, geçerliliklerini yenileyecek ve iptal edecek bir Sertifika Makamı.
- Bağlantısız kullanıcıları yetkilendirecek ve sertifikaların özelliklerini tanımlayacak bir Kayıt Makamı.
- Sayısal dokümanları kodlayacak ve imzalayacak açık anahtar kullanıcıları.
- Güvenilir bir Sertifika Makamı tarafından sayısal imzaları yayınlayacak, açık anahtar kullanıcıları.

- Kullanılabilir açık anahtar sertifikalarını ve sertifika iptal listelerini depolayacak, genel erişime açık sunumcular.
- Açık anahtar sertifikaları (Bkz. Ek 1)
- Sertifika uzantıları (Bkz. Ek 2)
- Organizasyon içerisindeki güvenlik ve sertifika mekanizmalarını tanımlayan ve kontrol mekanizmaları sunan sertifika politikası

2.2 Açık Anahtar Altyapısı Fikri

Açık anahtar altyapılarının, geçmiş 400 yıllık kriptoloji tarihi içerisinde geliştirilmiş olan en ciddi geliştirme olduğuna dair akademik dünyada güçlü bir kanı mevcuttur. Bir bilginin, bir açık anahtarla bağlanması fikri yaklaşık 30 yıllık bir geçmişe sahiptir. Ancak geçmiş yıllar içerisinde, bu fikrin geliştirilmesi ve yaygın bir şekilde kullanımı, tam anlamıyla ancak son dönemlerde sağlanabilmiştir. Bu bölümde açık anahtar fikriyle ilgili gelişimler, tarihsel olarak kısa ve özet bilgilerle sunulmuştur.

Açık anahtar kriptografisinin geçmişi incelendiğinde, ilk yıllarda insanlar, bir telefon rehberi kullanımına benzer şekilde, içerisinde birçok açık anahtar bilgisini içeren veritabanlarından faydalanmaktalardı. Örneğin Bob, bu rehber içerisinden Alice için tanımlı kayda bakarak, bu isimle ilişkili açık anahtara erişim yapabiliyordu. [Schneider, B. 2004] Bu yöntem, içeriği itibariyle problemlere neden olmaktaydı. Bu problemlerin başındaysa, veritabanının bütünlüğünün korunmasındaki güçlükler gelmekteydi. Diğer bir sıkıntıysa, veritabanındaki isim ve açık anahtar eşleştirilmesine karşı duyulan güvende ortaya çıkmaktaydı. Bu tür sıkıntılara karşılık geliştirilen çözüm, açık anahtar altyapısı fikri olmuştur. Açık anahtar altyapısı, çoğunlukla bir standart olarak algılansa da, gerçekte bir grup politikalar, ürünler ve kurallar bütünü olarak ele alınmalıdır. Politikalar, kuralları tanımlayarak hangi kriptografik sistemlerce çalıştırılabilirliğini belirlerler. [Pfleeger, C. ve Pfleeger, S. 2003] Bu gruplar, mimariyi oluşturan yapıya ait temel kolonlar olarak değerlendirilebilir. Farklı mimarilerin, birbirlerine kıyasla teknik olarak özelliklerinin karşılaştırılması için, Ekler bölümünde sunulan tablo incelenebilir. (Bkz. Ek 4)

2.2.1 Whitfield Diffie ve Martin Hellman - 1976

Stanford Üniversitesi profesörlerinden Martin Hellman ve öğrencisi Whitfield Diffie, 1976 yılında yayınladıkları makaleyle, açık anahtar kriptografisi ile ilgili bilgileri topluma ilk kez sunmuşlardır. Bu zamana kadar kullanılan şifreleme yöntemleri, temelde simetrik anahtarların kullanımı ve dağıtımına dayanmaktaydı. Yayınlanan bu makalede, açık dizin olarak ifade edilen, genel erişime açık bir paylaşımlı ortamın, anahtar dağıtımındaki sorunu gidermesine yönelik faydaları ve simetrik anahtarlamayla ilgili sıkıntılara karşı oluşturduğu çözüm niteliğindeki özellikleri aktarılmıştır. Makalede belirtilen ana fikirde, bir kişi, ikinci bir kişi için, özel bir mesajı şifreliyorsa, bu kişi, sadece ikinci kişiyle paylaşacağı tek bir anahtarı kullanmalıdır. Benzer şekilde eğer bir kişi, ikinci bir kişiden bir mesaj almışsa, bu mesaj, bu kişinin güvendiği bir birim tarafından geçerlenerek doğrulanmış olmalıdır. [Ellison, C. et.al, 1999]

2.2.2 Rivest Shamir Adleman - 1977

Diffie ve Hellman tarafından ortaya atılan açık anahtar fikri, teorik olarak 1976 yılında çıkmış, ancak bir yıl sonrasında, MIT üniversitesinin bir diğer matematikçiler topluluğunun oluşturduğu ekipte yer alan, Ronald L. Rivest, Adi Shamir, ve Leonard M. Adleman, bu teoriyi gerçek hayatta uygulayabilecek bir yöntemi sunmuşlardır. Soyadlarının ilk harflerini alarak, RSA olarak adlandırdıkları bu yöntemle, açık anahtar kriptografisine yönelik pratik bir kodlama metodu ortaya konmuştur. RSA metodu aracılığıyla kullanıcılar, sahip oldukları anahtarlardan birisini dağıtırken, diğerini gizli tutabilir. İletişimde bulunmak isteyen bir kişi, açık anahtarını kullanarak mesajı şifreleyebilir ve mesajı alıcısına iletebilir. Gizli anahtarını kullanarak alıcı, mesajı deşifre ederek okuyabilir.

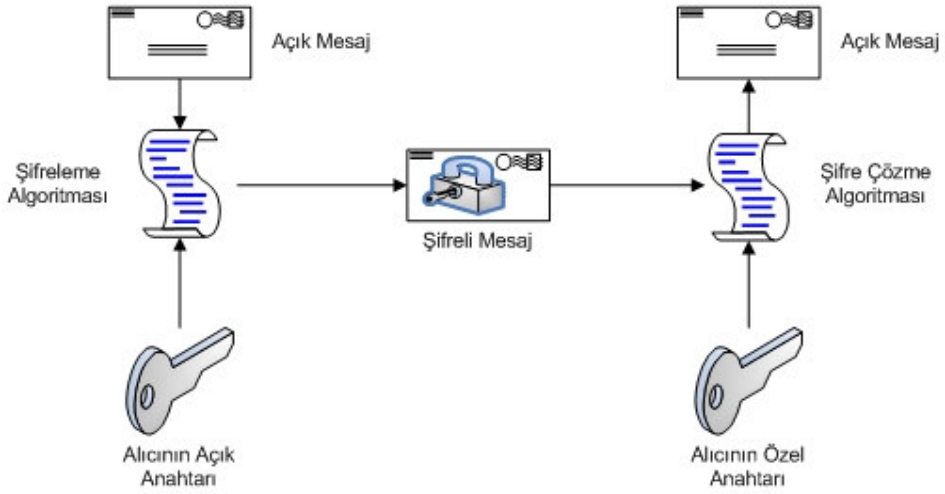
2.2.3 Loren Kohnfelder - 1978

Diffie ve Hellman tarafından geliştirilen ortak paylaşımlı dizin mantığı, açık anahtar kriptografisinin gücünü duyurmak için etkili bir yöntem olarak gözükse de, yöntemin bazı sıkıntıları mevcuttu. Loren Kohnfelder, 1978 yılında MIT üniversitesinde geliştirdiği yüksek lisans bitirme tezinde, bu problemlere yönelik çözümler geliştirmiştir. [Ellison, C. et.al, 1999] Kohnfelder tezi, literatürde açık anahtar altyapısında çok önemli olan, sertifika ve sertifika iptal listeleri gibi kavramları ortaya atmasıyla, açık anahtar tarihçesinde önemli bir yere sahiptir.

2.3 Açık Anahtarlı Şifreleme

Bir diğer adıyla asimetrik şifreleme yöntemi, simetrik şifreleme yönteminin en önemli dezavantajı olan, özel anahtarın karşı tarafa iletilme zorluğunu gidermek amacıyla geliştirilmiş bir yöntemdir. Asimetrik şifreleme mantığında, açık ve gizli olarak adlandırılan iki adet anahtar oluşturulur. İletilecek veri, açık anahtarla şifrelenir ve ancak gizli anahtarla deşifre edilebilir. Açık anahtar, umuma açık olup, sistem birimlerince bilinmesinde herhangi bir sakınca yoktur. Ancak gizli anahtarın sistem içerisinde kesinlikle gizli olarak saklanması gerekir. Bunun sebebi, açık anahtarla sadece veri şifrelenirken, bu anahtarla şifrelenen veriler ancak ve ancak ilgili gizli anahtarla açılabilir.

Yöntemin çalışma mantığı kısaca belirtilecek olursa, alıcı, kullandığı cihaz üzerinden bir açık ve gizli anahtar çifti oluşturur. Oluşturulan açık anahtar, alıcıdan göndericiye, sistem aracılığıyla iletilir. Gönderici, aldığı açık anahtarla veriyi şifreler ve şifreli bir şekilde veriyi alıcıya iletir. Alıcı da, kendisinde bulunan gizli anahtarı kullanarak veriyi deşifre eder. Asimetrik kriptosistemlerde, açık anahtarın ele geçmesi durumunda gizli anahtarın öğrenilebilmesi neredeyse imkansızdır. Bu sistemlerin dezavantajı olarak, simetrik kodlama yöntemiyle kıyaslandığında performans açısından yavaş olmalarıdır. Önceki bölümde aktarılan RSA ve eliptik eğri kriptosistemleri olarak bilinen DSA, bilinen ve yaygın olarak uygulanan iki farklı asimetrik kripto algoritması uygulamasıdır. Hız ve bilgisayar gücü gereksinimleri doğrultusunda, gizli anahtar sistemlerinin mesajı kodlaması ve açık anahtar sistemlerinin gizli anahtarı kodlaması çalışmaları gerçekleştirilmektedir. Tipik bir asimetrik şifreleme işleyişi modeli aşağıdaki şekilde sunulmuştur. (Bkz. Şekil 2.2)



Şekil 2.2 Asimetrik şifreleme

Açık anahtar altyapısı, gizlilik, veri bütünlüğü, kimlik belirleme ve inkar edememe fonksiyonlarını, sayısal sertifika kullanımı ile sağlar. Sertifika, kişiye özel sayısal bir kimlik olduğu gibi aynı zamanda, sahibine ait bilgiler ile gerekli algoritma anahtarlarını üzerinde bulunduran bir yapıya sahiptir. Sertifikaların saklanmasında ortaya çıkan güvenlik problemleri, depolama aracı olarak akıllı kartların kullanımıyla ortadan kalkar. Akıllı kartların kullanımı ve taşınması, yapısı nedeniyle oldukça kolaydır. Akıllı kartlar, sahip oldukları güvenlik mekanizmaları sayesinde fiziksel, elektriksel ve kimyasal etkilere karşı korunurlar.

2.4 Açık Anahtar Sertifikaları

Önceki bölümde belirtildiği gibi bir Sertifika Makamı'nın temel görevleri, açık anahtar sertifikalarının oluşturulması, yönetimi, saklanması ve iptal edilmesi işlemleridir. Bir açık anahtar sertifikası, sistemdeki bir birime ait kimlik bilgisini ve açık anahtarı, birbirleriyle ilişkilendiren bir mekanizma sağlar. Bu şekilde sistemdeki bir başka birim, bu mekanizma sayesinde, sertifikanın sahibi olan birimin kimliğini doğrulayabilir. Sertifika yapısı en genel haliyle, ilgili birimin adını, sertifika geçerlilik süresini ve birime ait açık anahtarı içerir. Sayısal sertifikalarda en yaygın olarak kullanılan format, IETF X.509 standartlarını içeren yapılardır. X.509 tabanlı açık anahtar sertifikalarının detaylı olarak profilleri, IETF RFC 2459'da belirtilmektedir. Aynı zamanda sayısal sertifika formatı için tek bir yapı olmayıp, üretici veya uygulayıcıya bağlı olarak farklı standartlar da geliştirilmektedir. (Bkz. Ek 1)

2.5 Sertifika Uzantıları

Sertifika uzantıları, bir sertifika içerisinde yer alan verilere ek olarak, organizasyonel gereksinimlerin karşılanabilmesi amacıyla, farklı türdeki verilerin sertifikada yer alabilmesini sağlayan yapılardır. Burada önemli bir husus, sistemde sertifikaları değerlendiren mekanizmaların, uzantıların yapılarıyla ilgili bilgi sahibi olmalarıdır. Aksi halde sistemdeki işlevsel fonksiyonlarda hatalar oluşabilir. Bir sertifika uzantısının içereceği bilgi türleri arasında, organizasyonun sahip olduğu açık anahtar altyapısına özel olarak politika bilgileri, sertifika kullanımıyla ilgili bilgiler ya da sertifika iptal bilgileri yer alabilir. (Bkz. Ek 2)

2.6 Açık Anahtar Altyapısı İşlevleri

Geniş ölçekli bir açık anahtar altyapısında sağlanması gereken en genel fonksiyonlar, bu bölümde genel olarak açıklanmıştır. Bu fonksiyonlar, farklı tipteki organizasyonel altyapılarda ortak kullanılmasına karşın, uygulanacak ek servislere yönelik altyapı işlevlerinde bazı eklentiler ve farklı servisler uygulanabilmektedir.

2.6.1 Kayıt

Açık anahtar altyapısı içerisindeki kullanıcılar, sahip oldukları açık anahtarlarla ilişkilendirilebilmek için, güvenilir bir servis aracılığıyla sistemde kayıtlı olmalıdırlar. Bu servis talepleri alarak kayıt ve kullanıcı kimlik bilgilerini doğrular. Bu işlem, sistemdeki Sertifika Makamı'nın, ilgili birim için bir sertifika yayınlamadan önce, bu birimin, kendisini sistemdeki Sertifika Makamı ya da Kayıt Makamı'na tanıtması olarak düşünülebilir. İşlem sonunda Sertifika Makamı, birimin açık anahtarına ait sertifikayı sistemde yayınlayarak birime iletir veya sertifika deposunda saklar. [Adams, C. ve Farrell, S. 1999]

2.6.2 Sayısal İmzalar

Sistemde iletilen mesajlara ait, bütünlük, inkar edilemezlik ve doğruluk ilkelerini sağlayan temel yapılardır. Mesajların özet bilgisinin alınması ve bu bilgilerin kodlanması mekanizmalarına sahiptir. Sayısal imzalar, internet haberleşmesinin, şirketler ve bireyler açısından açık anahtar kriptografisinin kullanımıyla daha güvenilir hale getirilmesinde uygulanan öncül yöntemlerden birisidir. Örneğin, bir e-posta için sayısal imza oluşturulmak istendiğinde, gizli anahtar kullanılarak e-postanın bir kopyası şifrelenir. Şifrelenmiş bu veri, sayısal imza olarak adlandırılır. Oluşturulan sayısal imza, e-posta ve göndericiye ait sertifikayla birlikte sistemden alıcıya iletilir. Sayısal imza, alıcı tarafında ancak ve ancak, göndericiye ait sertifika içerisinde alıcıya iletilen açık anahtar kullanılarak deşifre edilebilir ve doğrulanabilir. Bir kişi, birim ya da bir elektronik cihazın kimliğinin saptanmasında sayısal imza, sayısal sertifikalarla birlikte kullanılabilir. Aynı zamanda sayısal imza, ilgili mesaj ya da dosyanın içeriğinin bozulmadığının doğrulanmasında da uygulanabilir.

2.6.3 Şifreleme

Mesajların gizlilik içerisinde taşınabilmesini sağlayan ve bünyesindeki kripto mekanizmaları ile mesaj içeriklerinin anlaşılmasını ve edinilebilmesini engelleyen servislerdir. Açık anahtar altyapısı içerisinde şifreleme, açık ve gizli anahtarların karşılıklı olarak kullanılmasına dayanır. Bu bölümdeki bilgiler, çalışma içerisinde yer alan Asimetrik Şifreleme kısmında detaylı olarak belirtilmiştir. (Bkz. Bölüm 1.3)

2.6.4 Zaman Pulu

Sistemde yapılan işlemlerde, işlem güvenliği kadar işlemin yapılma zamanı da önem taşımaktadır. Bu nedenle yetkili bir makam veya zaman sunumcu tarafından, sayısal imzalara bir servis aracılığıyla zaman pulları atanarak bu kontrol sistem içerisinde sağlanır. [Adams, C. et.al, 2000]

2.6.5 İnkar Edilemezlik

Bağımsız dış makamlar tarafından gerçekleştirilen, sistemdeki bilgilerin doğruluğu ve gerçekliği bazında oluşturulması, yeniden elde edilmesi, işlenmesini, onaylanması ve iptal edilmesi gibi fonksiyonları içeren servistir. İletilen mesajın göndericisi tarafından iletildiğinin kanıtlanmasına yönelik mekanizmalar sağlar. Sistemde bu tip bir işlemi yapan, işlemi yapmadığını ya da işlemi yapanın kendisi olmadığını, bu servis sayesinde iddia edemez.

2.6.6 Anahtar Yönetimi

Sistemdeki anahtarların güvenli bir şekilde saklanması ve yönetilmesinden sorumlu temel açık anahtar altyapısı servislerinden birisidir. Anahtarların oluşturulması, sahiplendirilmesi, dağıtımı, yedeklemesi gibi anahtar yaşam döngüsünde yer alan mekanizmaları yöneten ve işleten servistir.

2.6.7 Anahtar Çifti Güncelleme

Sistemdeki her bir anahtar çiftinin, belirli periyotlarla sistem içerisinde yeni bir anahtar çiftiyle değiştirilerek güncellenmesi ve bu anahtarlarla ilgili sistemde yeni bir sertifikanın yayınlaması gerekmektedir. Bir sertifika yenileme işleminde, bir sertifika düzeyi, uç birim tarafından kullanılmakta olan anahtarın güncellenmesini talep edebilir. Bu talep, sistemde bir anahtar güncelleme talebi mesajı olarak yer alır. Şifrelemede kullanılan anahtarlar sistemde güncellendiğinde, önceki şifre çözme anahtarlarına ait geçmiş kayıtların saklanması gerekir. Bu arşiv niteliğindeki kayıtların saklanmasıyla, sistem kullanıcıları, geçmiş dönemlerde eriştikleri ve şifrelerini çözdükleri mesajlara erişim haklarını devam ettirirler. Sertifika Makamı'na ait anahtarların da, sistemdeki diğer tüm anahtarlar gibi belirli bir geçerlilik süresi bulunmakta ve belirli zaman aralıklarında güncellenmeleri gerekmektedir.

İmzalama anahtar çiftinin sistemde güncellenmesi durumunda, önceki imzalama anahtarı güvenli bir şekilde imha edilir. Bu imha işlemi, yetkisiz bir kişinin imzalama anahtarına erişmesini engeller. Nitekim, geçmiş imzalama anahtarlarının sistemde saklanmasına bir gerek bulunmamaktadır.

2.6.8 Dizin Servisleri

Dizinler, belirli bir kullanıcı grubuna, bir takım bilgilerin sunulmasını sağlayan araçlardır. Günümüzde birçok uygulama ve araç, türlü dizin servisi mekanizmalarını kullanmaktadır. Dizinler üzerindeki bu tip uygulamaların artmasıyla, farklı tipteki dizinlerin yönetimi ve

desteklenmesinde sıkıntılar meydana gelmiştir. Bu sıkıntılara yönelik geliştirilen çözümlerin başında, standart bir erişim protokolü üzerinden tek bir dizin içerisinde tüm bilgilerin kullanıcılara sunulması gelmektedir. Bu tür dizin mekanizmalarına, dizin servisleri adı verilir.

Bir açık anahtar altyapısı göz önüne alındığında, dizin servislerinin kullanım amaçları, e-posta ya da benzeri uygulamalarda mesajın kodlanarak gönderilmesinin öncesinde, kullanıcıya ait sertifika bilgisinin alınması gerektiğinde veya sertifika iptal listesi gibi sertifika durumlarını içeren bilgilerin sistemden edinilmesinde bu servislerden yararlanılmaktadır. Bütün açık anahtar yapılarında dizinlere gerek duyulmayabilir. Açık anahtar tabanlı uygulamalarda, en yoğun kullanıma sahip dizin servisi erişimleri, LDAP protokolü ile sağlanır. Ancak diğer dizin tabanlı uygulamalar da benzer şekillerde sistem konfigurasyonu ve doğru entegrasyonlarla, açık anahtar sistemleri içerisinde rahatlıkla uygulanabilir.

2.6.9 Yetkilendirme

Güvenilir üçüncü şahıslar tarafından, sistemdeki kaynaklar ve bu kaynaklar üzerinde hakları bulunan birimlerle ilgili olarak, hakların tanımlanması, grupların atanması, doğrulama, kullanıcıların gruba eklenmesi, yönetim yetkilerinin belirlenmesi gibi fonksiyonları içeren işlevler bütünüdür.

2.6.10 Güven Noktası

Sistemdeki her bir açık anahtar kullanıcısı, güvendiği bir makam tarafından yayınlanan bir açık anahtara sahip olmalıdır. Organizasyonlar, kendi içinde tek bir güven yönetimi etki alanında, ilgili Sertifika Makamları tarafından oluşturulan ve belirli periyotlarla güncellenen bir güven noktası oluştururlar. [Dzambasow, Y. et. al, 2005] Farklı etki alanlarında, organizasyonlar arasındaki güven ilişkileri, çapraz sertifikaların uygulanmasıyla sağlanır. Güvenilir bir üçüncü partiyle olan ilişkilerdeki güven mekanizmasının temel özellikleri, geçişkenlik ve seçicilik ilkeleriyle açıklanır. Bir A birimi, B birimine güven sağlamışsa ve B birimi, C birimine karşı bir güven ilişkisi oluşturmuşsa, bu durumda geçişkenlik ilkesi gereğince A birimi ile C birimi arasında bir güven mekanizması oluşur. Seçicilik ilkesinin temel mantığı ise, güven makamı olarak işlev gören bir birimin, güven ilişkisinde bulunduğu birimlere karşı aktivitelerinde farklı ve birbirleriyle bağımsız mekanizmalara sahip olabilmesi olarak açıklanabilir. Hesaplama tabanlı, bilgi tabanlı ya da geçişkenlik tabanlı gibi birçok güven mekanizması türü bulunmaktadır. Benzer şekilde farklı güven modelleri de, farklı sistemler ve altyapılara yönelik olarak yoğun bir şekilde uygulanabilmektedir. Bu güven modellerinden bazıları, Kök Sertifika Makamı Modeli, Köprü Modeli ve Son Kullanıcı Güven

Modeli olarak örneklenebilir. Bu modeller genellikle, farklı yapılara sahip ortamlarda ya da organizasyonel etki alanlarında, açık anahtar altyapılarında gereken güven mimarilerinin oluşturulmasında, tasarımcılara farklı alternatifler sunmaları açısından önem taşımaktadır.

2.6.11 Politika Tabanlı Sertifika Yolu Doğrulama

Bir sertifikanın yayınlanması, genelde bir Sertifika Makamları zinciri içerisindeki doğrulama mekanizmalarının gerçekleştirilmesi sonucunda oluşur. Zincirde yer alan her bir Sertifika Makamı, doğrulama ile ilgili tüm politika gereklerini sağlayabilmelidir. Sertifikanın doğrulanması prosedürü, belirtilen sertifika yolundaki tüm makamlar tarafından, kontrol mekanizmaları gerçekleştirildikten sonra işlevsel hale gelir.

2.7 UMU-PKI Tasarımı

IPv6 destekli açık anahtar altyapısı tasarımı, sistem kullanıcılarının haberleşmelerinde, çeşitli güvenlik mekanizmalarını entegre etmek isteyen bir organizasyonun, geçerli ve sistematik sertifikalama servislerini uygulayabilmesini amaçlar. Bu servisler aracılığıyla bir kullanıcı, kullandığı arayüz üzerinden, sertifika talebi, yenilenmesi, iptali ya da haberleşmek istediği başka bir kullanıcının sertifika bilgilerinin sorgulanması işlevlerini, sistem bünyesinde kolaylıkla gerçekleştirebilir. Aynı zamanda bu kullanıcılar, içinde bulundukları organizasyonca dağıtılan akıllı kartları kullanarak, kendilerine ait kriptografik bilgileri saklayabilir ve bu sayede mobilitelerini de arttırabilirler.

UMU-PKI, tamamen JAVA teknolojisi kullanılarak geliştirilmiş bir altyapıya sahip bir sistem modeli olup, bu özelliği sayesinde, farklı platformlar içerisinde kolaylıkla entegre edilebilir bir yapı öngörür. Sistem dahilinde akıllı kartların uygulanması, şu an itibariyle ancak Windows tabanlı sistemlerde mümkün olup, Unix platformunda işletilen sistemlerde de benzer şekilde kullanılabilmesine yönelik çalışmalar devam etmektedir.

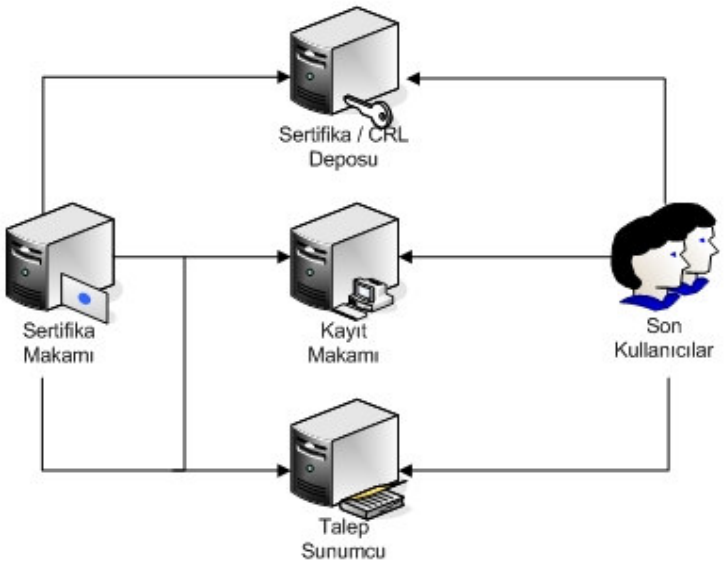
UMU-PKI v6'ya ait temel karakteristik fonksiyonaliteler :

- LDAP v6 dizin servisleri, kullanıcıları, sertifika iptal listelerini ya da Sertifika Makamı'na bağlı sertifikaları saklamayı destekler.
- Kullanıcılar, web arayüzleri üzerinden sertifika işlemlerini gerçekleştirebilirler.
- Kullanıcılar, akıllı kartlarla, kendilerine ait gizli anahtarlarlar, sertifikalar veya Sertifika Makamı'na ait sertifikalar gibi kriptografik bilgileri saklayabilirler. Bu sayede kullanıcılar, farklı bir web arayüzü içerisinden işlemlerini gerçekleştirerek mobilitelerini arttırabilirler.
- HTTP üzerinden temel açık anahtar konfigürasyonu yapılabilir.

- Aynı organizasyon içerisinde sistem yöneticisince belirlenerek uygulanan güvenlik politikaları, sistem dahilindeki tüm bileşenlerce entegre edilebilir.
- JAVA tabanlı geliştirilmiş bir sistem olup, bu özelliği sayesinde, farklı platformlara uygulanabilirliği sağlanmıştır.
- PKIX grubunca belirlenmiş IETF standartlarına uygundur.
- VPN cihazlarının sertifikalanmasına yönelik SCEP protokolünü destekler.
- 6WIND VPN yönlendiricileri destekler.
- Çapraz sertifikaları destekler.
- Sistemdeki birimler arası haberleşmede IPv6 veya IPv4 kullanılabilir.

2.7.1 Sistem Bileşenleri

UMU-PKI v6 sistemine ait temel sistem bileşenleri, açık anahtar altyapılarının genelinde karşılaşılan, uygulanan ve çalışma amaçları bilinen, temel yapıları içermektedir. Bu birimlerden işlevleri ve sistem içerisindeki sorumlulukları açısından öne çıkanlar, Sertifika Makamı, Kayıt Makamı, Talep Sunumcu ve Sertifika Deposu'dur. [Skarmeta, F. et.al, 2002] İşlevsel olarak sistem içerisinde görev yapan bu bileşenlerin temel sorumlulukları, günümüzde uygulanmakta olan diğer açık anahtar altyapılarındaki işlevleriyle büyük ölçüde benzerdir. Bu bileşenlerin sistem içerisindeki ilişkileri aşağıdaki şekilde gösterilmiştir. (Bkz. Şekil 2.3)



Şekil 2.3 UMU-PKI v6 sistemi temel bileşenleri

2.7.1.1 Kayıt Makamı

Sertifika Makamı tarafından işlenmesi için gönderilen taleplerin, iletiminden sorumludur. Sistem konfigürasyonuna bağlı olarak, birden fazla sayıda Kayıt Makamı, sistemde yer alabilir. Sistem yöneticisi, sertifikalama, yenileme ya da iptal taleplerini, ilgili politika gereğince kontrol edebilir. Bir Kayıt Makamı, istemciler tarafından iletilen taleplerin geçerliliğini kontrol ederek, CPS'e ve sistem politikasına uygun olup olmadığına karar verir. Bu kontroller sonucunda talep geçerlenerek Sertifika Makamı'na iletilir. Kayıt Makamı, Windows tabanlı ortamlarda, akıllı kart kullanımını tam olarak destekleyebilmektedir.

2.7.1.2 Talep Sunumcu

Sistem içerisindeki tüm birimlerden yapılan talepleri toplayan ve saklayan birimdir. Sistemde yer alan Kayıt Makamı, son kullanıcılar ve sistem yöneticisinden gelen farklı tipteki talepleri, kendi bünyesindeki veritabanında, Sertifika Makamı tarafından işlenmek üzere saklar.

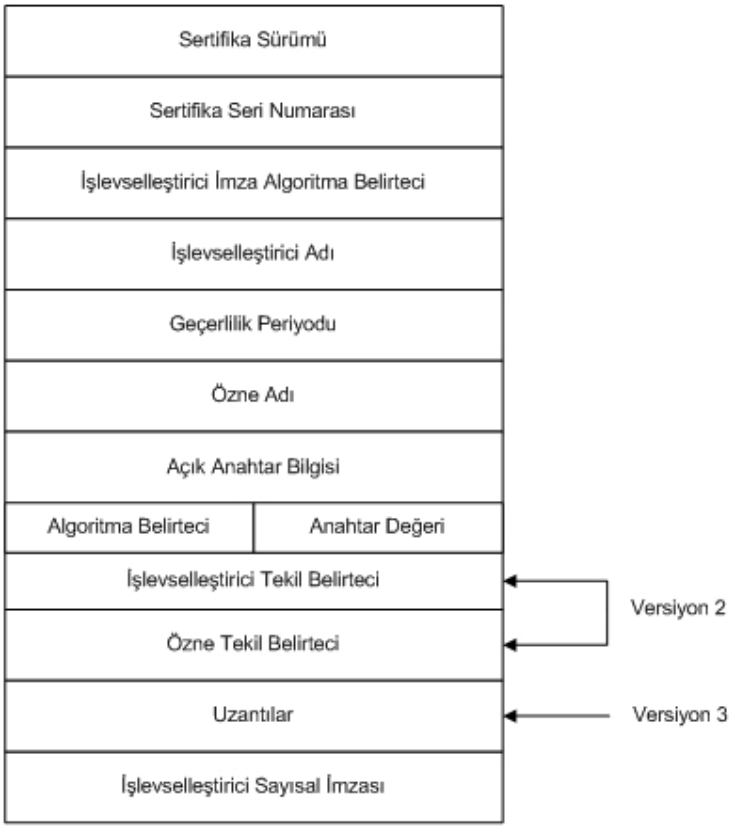
2.7.1.3 Sertifika Makamı

Sertifika Makamı, önceki kısımda belirtilmiş olan Talep Sunumcu'ya ait veritabanında yer alan talepleri gerçekleştirmekten sorumludur. Eğer işlenecek talep, bir sertifika talebiyse ve sertifika, sistem politikalarına uygun olarak gerçeklenebilir durumdaysa, Sertifika Makamı tarafından oluşturulan sertifika, veritabanında kaydedilerek yayınlanır. [Kaliski, B. 1998] İlgili kullanıcı, imzalı bir e-posta ile bilgilendirilir. Eğer talep, bir sertifika yenileme talebiyse, sistemde geçerli olan sertifika bilgisi, Sertifika Makamı tarafından veritabanında ve LDAP sunumcuda güncellenir. Eğer sertifika iptal talebiyse, veritabanındaki sertifika bilgisi, iptal edilmek üzere Sertifika Makamı tarafından işaretlenerek, sertifika iptal listesinde yayınlanmak üzere kaydedilir. Buradaki kritik husus, Talep Sunumcu ile Sertifika Makamı arasındaki bağlantının tek yönlü olmasıdır. Bu sayede sistemdeki herhangi bir birim Sertifika Makamı'na direk bağlantı kuramaz.

2.7.1.4 X.509 Sertifikaları

X.509 formatı, açık anahtar altyapılarında en yoğun uygulamaya sahip sertifika formatı olup, aynı zamanda SSL, IPSec, S/MIME ya da PEM gibi PKI tabanlı protokollerde ve uygulamalarda desteklenen sertifika formatıdır. Sertifika verisi X.208 ve X.209 ITU-T standartlarında tanımlı ASN.1 notasyonu ile yazılmıştır. X.509'un ilk sürümü ITU-T tarafından 1988 yılında tanımlanmış olup, şu anki sürümü olan X.509 v3 ise 1996 yılından bu yana geçerli durumdadır. [Housley, R. et. al, 2002]

X.509 sertifikalarına ait format yapısı Şekil 2.4'te gösterilmektedir. (Bkz. Şekil 2.4)



Şekil 2.4 X.509 sertifikası

X.509 sertifika formatı ve sertifika uzantılarıyla ilgili detaylı bilgiler EK-1 ve EK-2 bölümlerinde sunulmuştur. (Bkz. Ek 1 , EK-2)

2.7.1.5 Son Kullanıcılar

Sistemde yer alan son kullanıcılar, web arayüzleri aracılığıyla, Kayıt Makamı üzerinden sertifika işlemlerini gerçekleştirebilirler. Eğer bu işlem, bir sertifika talebi işlemiyse, talep, Talep Sunumcu'da kaydedildikten sonra Kayıt Makamı tarafından geçerlenir ya da iptal edilir. Eğer bir yenileme ya da iptal talebiyse, bu durumda talep, direk olarak Sertifika Makamı tarafından işleme alınır. Web arayüzleri üzerinden bu işlemleri gerçekleştirmek isteyen kullanıcılar, akıllı kartları ile kriptografik bilgilerini saklayarak da bu işlevleri sağlayabilirler.

2.7.1.6 Politika

Politikanın tanımlanması ve uygulanması, Java tabanlı PKI v6 sisteminin ana karakteristiklerinden biridir. Sistemdeki birimler tarafından gerçekleştirilebilecek işlevler, bu politika ile uyumlu olmalı, aksi taktirde sistem içerisinde bu işlevlerin yerine getirilmemesi sağlanmalıdır. Politikalar, organizasyondaki kullanıcıların, sistemde gerçekleştirecekleri işlemlerde, hangi sınırlamaların olması gerektiğini tanımlayan temel yapılardır. Örneğin bir organizasyon, kendi politikası gereğince tüm işlevsel sertifikaların, 2048 bit uzunlukta RSA anahtarlarını kullanmalarını ya da sadece belirli bir zaman aralığı içerisinde sertifika yenilemelerinin oluşturulmasını kullanıcılarına zorlayabilir.

2.7.1.7 Sertifika Deposu

UMU-PKI v6 sistemi, opsiyonel olarak sistem kullanıcılarına, Sertifika Deposu kullanımı imkanını sağlamaktadır. Bir LDAP v6 dizini içerisinde tüm kullanıcı sertifikaları, Sertifika Makamı'na ait sertifika ve iptal listeleri, kullanıcıların işlemlerine açık halde saklanabilir. Sertifika Makamı tarafından yayınlanan sertifikalar, sistem içerisinden Sertifika Deposu'na, LDAP tabanlı bir istemci uygulamasıyla aktarılabilirken, benzer şekilde sistemdeki istemciler, aktarılan sertifikaları Sertifika Deposu'ndan edinebilirler. Burada özellikle LDAP tabanlı dizin ve uygulamaların kullanılmasının nedeni, sistem içerisindeki X.509 tabanlı sertifikaların en uygun şekilde X.500 dizinlerinde saklanabilmesidir. Bu tür dizin erişim işlemlerinde de en yaygın olarak tercih edilen protokol, LDAP protokolüdür.

2.7.1.8 Akıllı Kartlar

UMU-PKI v6 sistemi, web ortamlarında ya da Kayıt Makamı veya Sertifika Makamı birimlerince akıllı kartların kullanılmasına olanak sağlar. Kullanıcılar, kendilerine ait kriptografik bilgileri saklamak amacıyla akıllı kartlara sahip olabilirler. Her akıllı kart, kullanıcısına ait gizli anahtarı, ilgili sertifikayı ve Sertifika Makamı'na ait sertifikayı içerebilir.

2.7.2 Temel Servisler

Çoğunlukla PKI sistemlerince uygulanabilen, sertifikaların talep edilmeleri, geçerlenmeleri, yayınlanmaları, iptal edilmeleri ve yenilenmelerini içeren sertifika yaşam döngüsü bileşenlerini sağlayan yapılardır.

2.7.2.1 Sertifika Talepleri

UMU-PKI v6 sistemi içerisinde sertifika talepleri, iki farklı format üzerinden yapılandırılabilmektedir. Bunlar, Microsoft Internet Explorer için geliştirilmiş olan PKCS#10 ve Netscape için geliştirilmiş KEYGEN format tipleridir. Talep yapısı her iki format tipi için aynıdır. Sertifika talepleri sistem içerisinde birkaç farklı şekilde gerçekleştirilebilir. [Skarmeta, F. et.al, 2002] Bunlar :

- Kullanıcılar, akıllı kartlarını kullanarak ya da kimliklerini ispatlayacak bir dokümanla, Kayıt Makamı'na erişim yaparak taleplerini gerçekleştirebilirler. Kayıt Makamı, PKCS#10 nesnesini kullanarak yeni bir talep oluştur. Talep, SSL bağlantısıyla Talep Sunumcu'ya iletilerek, ilgili gizli anahtar, kullanıcıya ait akıllı kart ya da bir dosya içerisinde saklanır.
- Sistem servisleri, bir başka yazılım tarafından oluşturulan talep için daha önce yaratılmış bir PKCS#10 nesnesiyle birlikte Kayıt Makamı'na iletilir. Bu talebin doğruluğu, sistem yöneticisi tarafından sistem politikasına uygun şekilde geçerlenir ve işleme alınmak üzere sistemden Talep Sunumcu'ya iletilir. Bunun yanı sıra, SCEP protokolü vasıtasıyla uygulamalar ve servisler sertifikalanabilir, bu sayede sistemde yer alan cihazlar da sertifika işlemlerinde yer alabilir. VPN cihazları genelde bu servisi kullanırlar.
- Kullanıcılar, kendi web arayüzleri vasıtasıyla ve eğer varsa akıllı kartlarını kullanarak, yeni bir sertifika talebi üretebilirler. Bu yöntemle kullanıcılar, taleplerini direk olarak Talep Sunumcu'ya iletir. Kayıt Makamı bu talepleri alır ve geçerler. Sonrasında talepler, sistemde yeniden Talep Sunumcu'ya iletilerek Sertifika Makamı tarafından işlenmek üzere saklanır.

2.7.2.2 Sayısal Sertifikaların Edinilmesi

Sertifika Makamı tarafından bir sertifika geçerli hale getirildiğinde, ilgili kullanıcının bu sertifikayı sistemden alabilmesi gerekir. Bu edinme işlemi, bir web arayüzü veya LDAP deposu üzerinden gerçekleştirilebilir. Kullanıcıya ait sertifika, iptal listesi ya da Sertifika Makamı sertifikası, web üzerinden kullanıcının arayüzüne indirilir. Eğer kullanıcı bir akıllı karta sahipse, açık ve gizli anahtarlar bu kart üzerinde kaydedilebilir. VPN cihazları gibi kriptografik cihazlar, SCEP protokolü aracılığıyla, Talep Sunumcu üzerinden direk olarak sertifika edinebilirler. Bunun yanı sıra sistemdeki geçerli sertifikalar, herkese açık bir depoda saklanır ve herhangi bir LDAP istemcisi üzerinden kullanıcılar tarafından edinilebilir.

2.7.2.3 Sertifika Yenileme Talepleri

Sistemde yer alan son kullancılar, kendi sertifikalarına ait geçerlilik sürelerini, sistemde yer alan politika uyarınca yenileyebilir ve güncelleyebilirler. Bu tip talepler, Kayıt Makamı veya web arayüzü üzerinden SSL bağlantıları ile sistemde sağlanabilir.

2.7.2.4 Sertifika İptal Talepleri

Kaybedilmiş ya da içeriği bozulmuş akıllı kartlar gibi, sistemdeki istisnai durumlarda, sayısal sertifikaların geçerliliği, geçerlilik süreleri dolmadan önce sistemde kaldırılabilir. Kullanıcılar, sertifikalarını iptal etmek için iki farklı yöntem kullanabilir. Bunlar, Kayıt Makamı üzerinden yeni bir iptal isteğinde bulunma ya da bu işlemi web üzerinden gerçekleştirmedir. İkinci yöntem ancak kullanıcının sertifikasının gizli anahtarına sahip olduğu durumlarda geçerlidir.

2.7.2.5 Politika Tanımlama

Sistemde tanımlanmış bir politika, bir seri numarası, geçerlenme tarihi, bir sonraki geçerlenme tarihi ve bir grup politika biriminin oluşturduğu sayısal bir dokümandır. Buradaki birimler, politika sınırlamalarının, sistemdeki hangi birimler üzerinde uygulanacağını tanımlar. Örneğin bir politika tanımlaması,"*ou*=Kripto, *o*=UEKAE, *c*=Istanbul" grubuna bağlı tüm birimler üzerinde geçerli olmak kaydıyla, bir RSA anahtarı uzunluğu koşulu içerebilir. Politika tanımlandıktan sonra, Sertifika Makamı tarafından imzalanır ve sistemde entegre edilebilir hale gelir.

2.7.3 Ek Servisler

UMU-PKI sistemi, kullanıcılarına fayda sağlayacak bir takım ek servislerle, sistem yapısının kullanılabilirliğini geliştirmektedir. Bu servislere örnek olarak SCEP protokolü ve IPv6 tabanlı 6WIND tipi yönlendiriciler için geliştirilen sertifikalama servisleri gösterilebilir.

2.7.3.1 SCEP

Bu servis, sayısal sertifika işlemlerine ihtiyaç duyan VPN cihazlarına yönelik, sertifikalama desteği sağlamaktadır. CISCO tarafından geliştirilen SCEP, VPN ortamlarındaki yönlendirici ve cihazların taşıdığı bilgilerin güvenliğini sağlar. [Liu, X. et. al, 2000] SCEP, sistemdeki sertifikaların, PKCS#10 gibi standart formatlarda, ağdaki cihazlara atanabilmesini sağlar. Bu protokolle Sertifika Makamı ve Kayıt Makamı, bu cihazlara yönelik açık anahtar dağıtım, sertifika kayıt, iptal, sorgulama gibi işlemleri gerçekleştirilebilirler. Yönlendiriciler, sunumcular ve ağdaki benzeri cihazlar, bu protokolü kullanarak, güvenli bir biçimde VPN

kurabilmek için gereken açık ve gizli anahtarları edinebilirler. Bu protokolün servis olarak uygulaması, sertifika taleplerini dinleyen ve VPN cihazlarla etkileşimde bulunan JAVA servletlerinin oluşturduğu bir SCEP sunumcusu tarafından sağlanır. UMU-PKI SCEP entegrasyonu, temelde tekil anahtarların kullanımına dayalı olup, CISCO yönlendiriciler üzerinde başarı ile test edilmiştir.

2.7.3.2 6WIND Yönlendiriciler

Gelişmiş IPv6 tabanlı yönlendiricilerden 6WIND yönlendiriciler, kriptografik bilgilerin alınmasında, kullanıcılarına farklı ve özel yöntemler sağlarlar. [Thompson, J. 2002] Sertifikaların yayınlanması ve alınmasında SSH protokolünden faydalanılır. UMU-PKI v6, kullanıcılarının sertifika edinme ve talep işlemlerinde SCP [SSH] kullanabilmeleri için, son sürümünde bir güncellemeye tabi tutulmuştur. SCP, bir SSH servisi olup amacı, lokal ve uzak sistemler arasında güvenli dosya kopyalama işlemini gerçekleştirmektir. Bu uygulama, sadece 6WIND tipi yetkili cihazların veri alışverişinde bulunduğu özel bir dizin içerisinden ortak çalışma imkanını sağlamaktadır. Yetkili istemci birimler sertifika talebinde bulunduklarında talep, Kayıt Makamı yöneticisince değerlendirilerek geçerlenir ya da iptal edilir. Eğer sertifika geçerlenirse, istemci, aynı dizin içerisinden geçerlenme bilgisine erişebilir. Aynı zamanda yetkili istemci, Sertifika Makamı'na ait sertifikaya ya da iptal listesi sertifikalarına, aynı yöntemle sistem içerisinden edinebilir.

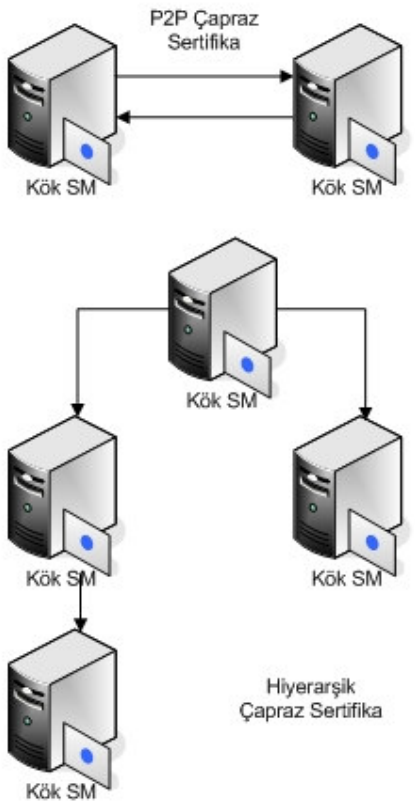
2.7.3.3 Çapraz Sertifika

Çapraz sertifikalama işlemi, farklı ortamlardaki Sertifika Makamlarının, kendi aralarında güven ilişkisi kurabilmelerine olanak sağlar. İki Sertifika Makamı arasında bir güven ilişkisi kurulduğu durumda, makamlardan birine ait sayısal sertifika ve anahtarlar, diğer makam tarafından, aynı organizasyon içerisindeymişcesine kullanılabilir. Bunun için Sertifika Makamları, birbirlerine ait çapraz sertifika bilgilerini edinirler. Çapraz sertifikalama, bir yönetim birimi içerisinde bulunan istemci ve uç birimlerin, farklı bir yönetimsel organizasyon içerisindeki istemcilerle güvenli bir biçimde haberleşmelerini sağlar.

Farklı şekillerde çapraz sertifikalama yapılabilir. Karşılıklı (P2P) Çapraz Sertifika olarak adlandırılan ilk yöntemde, iki Kök Sertifika Makamı arasında bir güven ilişkisinin kurulabilmesi için, ilişki kurmak isteyen makamlardan farklı bir Sertifika Makamı tarafından imzalanarak oluşturulan ve Çapraz Sertifika olarak adlandırılan yeni bir sertifikadan faydalanılır. Makamlar arasında güven ilişkisinin oluşturulmasında en fazla kullanılan Hiyerarşik Çapraz Sertifika yöntemindeyse, bir Kök Sertifika Makamı, hiyerarşik yapı

içerisinde altında yer alan Sertifika Makamı ile güven ilişkisi kurar. Bu yapı, makam hiyerarşisi içerisinde dallanarak devam eder. Son kullanıcıya ait sertifikanın doğrulanmasında, sistem içerisindeki zincirde yer alan tüm makamların sertifikaları doğrulanmalıdır. [Dzambasow, Y. et. al, 2005]

Şekil 2.5'de, belirtilen iki çapraz sertifikalama yönteminin işlevsel yapısı gösterilmektedir.



Şekil 2.5 Çapraz sertifikalama yöntemleri

UMU-PKI, Sertifika Makamları arasında P2P ve hiyerarşik yöntemle, güven ilişkilerinin kurulabilmesine olanak sağlar. P2P çapraz sertifikalamada UMU-PKI v6, sistem içerisindeki makamların güven ilişkisi kurabilmelerine yönelik, bir başka Sertifika Makamı'na ait sertifikayı imzalayabilmelerini sağlar. Bu yöntemde sadece Kök Sertifika Makamları güven ilişkisi kurabilir. İlişkiler tek yönlü olup, her yön için tek bir P2P çapraz sertifika üretilir. Sistemde yer alan hiyerarşik çapraz sertifikalama yönteminde ise sistem yöneticisi, kullanacağı bir Sertifika Makamı uygulaması yardımıyla hiyerarşik yapı içerisinde altta yer alan makamlar için yeni açık anahtar sertifikaları geçerler. Sistem içerisinde yeni bir alt Sertifika Makamı yaratıldığında bu makam, bir önceki çapraz sertifikaya gereksinim duyar.

Sistemdeki bir istemci, açık anahtar sertifikası talebinde bulunduğunda, tüm ara Sertifika Makamları ve Kök Sertifika Makamı zinciri, bu talebe müdahil olur. Bu nedenle sistemdeki bir son kullanıcının sertifikası doğrulanmak ya da geçerlenmek istendiğinde, zincirde yer alan tüm sertifikaların uygulama tarafından doğrulanması gerekir.

2.7.3.4 OCSP

Bir IETF standardı olan OCSP, herhangi bir X.509 yetkilendirme sistemi içerisinde yer alan sertifikaların geçerliliğini kontrol etmek amacıyla geliştirilmiştir. [Myers, M. et.al, 2000] OCSP altında yayınlanan sertifikalar, bir OCSP yetki belirteci alanına sahiplerdir. Sertifika uygulamaları, OCSP talepçisini kullanmak suretiyle OCSP yanıtlayıcısı üzerinden sertifika durumlarını öğrenebilirler. Burada temel amaç, sertifika iptal listeleri üzerinden ağa dağıtım yapmak suretiyle, periyodik olarak ağ üzerinden sertifika durumlarının elde edilmesi yöntemi yerine, daha etkin bir sertifika durumu sorgulama mekanizması geliştirmektir. Bir web uygulamasının SSL aracılığıyla bağlantı kurmasına benzer şekilde, bir güvenlik uygulaması içerisinde OCSP modelini içeren bir sertifika işlevsel hale getirildiğinde, uygulama, ilk olarak ilgili prosesin gerçekleşmesinden önce, bu sertifikanın geçerli olup olmadığını kontrol etmek ister. Sertifika geçerlilik süresini belirten bir başlangıç bitiş süresi ve opsiyonel olarak bir iptal listesi yayınlayıcısı ya da bir OCSP yanıtlayıcısına erişimde kullandığı adres bilgilerini içerir. Web gezginci ya da diğer bir sertifika işleyici uygulama, bu OCSP yanıtlayıcısı bilgisini kullanarak durum bilgisini elde etmeye çalışır.

Sertifika durumunun kontrolü işlemi, sertifika altyapılarının konfigürasonunda her zaman önemli bir yere sahiptir. Tarihsel olarak bakıldığında, iptal listelerinin kullanım zorluğu nedeniyle bu yöndeki çalışmalar, bu tip sistemlerin oluşumlarının başlangıcından bu yana kadar, üzerinde düşünülen konulardan birisi olmuştur. Günümüzde OCSP'nin bir standart haline gelmesi ve ayrıca CoreStreet, RSA, Verisign gibi şirketlerin desteklemesiyle sertifika sistemleri altyapılarının bu modeli içerecek şekilde güncellenmesi zorunlu bir hale gelmiştir.

3. Sertifika Yönetim ve Doğrulama Mekanizmaları

Önceki bölümlerde, açık anahtar altyapılarına ait bileşenler ve sistemsel mekanizmalar incelenerek, UMU-PKI sistemine ait mimaride yer alan birimlerin ve servislerin analizi gerçekleştirilmişti. Analizler içerisinde, açık anahtar tabanlı sistemde yer alacak sertifika mekanizmalarının önemi belirtilmiş, bu mekanizmaların gerçeklenmesinde, sertifika yaşam döngüsü ve doğrulama işlevlerinin temel alındığı yöntemlerin, sistemde sağlanmasının önemi vurgulanmıştır. Kritik değere sahip bu mekanizmaların, önceki bölümde detaylı bir şekilde ele alınan UMU-PKI sistemi bünyesinde, servislerinin sağlanmasına yönelik önerilen CMC ve SCVP protokollerinin, tercih edilme gerekçeleri ve diğer protokollerle eşleştirildiğinde, sahip oldukları avantajlarla ilgili bilgiler, çalışmanın ilk bölümü içerisindeki konular arasında yer almıştır. Bu aşamada, protokollerin detaylı analizleri gerçekleştirilmiş olup, protokollerin UMU-PKI sistemi içerisindeki süreçler öngörülerek modellenmesi, gerek süreç bazlı gerekse uygulama temelli entegrasyon tasarımlarının gerçekleştirilmesi ve bu protokollerin sistemde yer almasıyla edinilecek kazançlar üzerine detaylı incelemeler sunulmuştur.

Protokollerin UMU-PKI sistemi içerisine entegrasyonu çalışması içerisinde öncelikli olarak, sistemdeki sertifika yaşam döngüsü süreçlerini kapsayacak ve bu süreçleri yönetecek CMC protokolü ele alınarak, protokolün içeriği, gerekleri, temel yapıları ve uygulama yöntemleri incelenmiş, sonrasında SCVP protokolüyle ilgili çalışmalar gerçekleştirilmiştir. Protokollerin detaylandırılarak incelenmesinin ardından, bu bölüm içerisinde, protokollerin entegrasyonuna yönelik çalışmaların yer alacağı, süreç bazlı ve uygulama teknolojileri temelli tasarım ve modelleme aşamalarına geçilmiştir. Sertifika yaşam döngüsü içerisindeki tüm ana süreçler, birbirlerinden bağımsız olarak modellenmiş ve sonrasında tüm süreçlerin ortak çalışabilmesini öngörecek bir yönetim modeli tasarımı gerçekleştirilmiştir. SCVP protokolünde de benzer şekilde, süreçlerin ayrık olarak modellenmesinin ardından, sistemde yer alacak doğrulama servisinin süreç bazlı akış tasarımı gerçekleştirilmiş, bu servis mantığı içerisindeki mekanizmaların, süreçlerle ortak çalışabilirliği ile ilgili entegrasyon çalışması sunulmuştur.

3.1 CMC Protokolü

CMC protokolü, PKIX çalışma grubunun kontrolünde geliştirilen bir standart olup, PKCS#10 ve CRMF tabanlı açık anahtar sertifika ürün ve servislerine yönelik arayüz oluşturma gereksinimini sağlamaktadır. CMS ise, sayısal imzalama, yetkilendirme veya herhangi bir veri için anahtar yönetimli ya da yönetimsiz kodlama gibi farklı kripto mekanizmalarını içeren PKCS#7 nesnesinin IETF tarafından standartlaştırılmış varisidir. [Prafullchandra, H. et. al,

1997] Ek olarak, içerilen temel sertifika servislerini destekleyecek bir takım tamamlayıcı servisler de protokolde sunulmaktadır.

3.1.1 Protokol Gerekleri

Protokolün uygulanmasındaki temel gereksinimler şu şekilde sıralanabilir : [Myers, M ve Liu, X. et. al, 2000]

- PKCS#10 ve CRMF spesifikasyonlarını olabildiğince temel alır.
- Protokolün alt bileşeni olarak günümüz teknolojilerinde uygulanmakta olan PKCS#10 talep ve PKCS#7 yanıt nesneleri işlevselliklerini destekleyecektir.
- S/MIME ve diğer gruplarca gereksinim duyulan, çoklu anahtar kayıt protokollerini kolaylıkla destekleyebilecektir.
- Protokol bileşenlerindeki tüm operasyonların tek bir döngü içerisinde tamamlanması öngörülmüştür. Bunun mümkün olamadığı durumlarda, döngü sayısı asgari tutulacaktır.
- Protokol tasarımı, anahtar oluşum sürecinin tümünün istemcide olmasını sağlayacaktır.
- Protokoldeki algoritmalar S/MIME bağlantılı algoritmaları içerecektir.
- Protokol POP metodları içerecektir.
- Gecikmiş ya da askıdaki sertifika talepleri gibi dış kaynaklı prosedürlere gerek olan talepleri destekleyecektir.
- İstemciler ve sertifika yayınlayıcıları arasında yer alan Yerel Sertifika Makamları ilişkisel bağlantılarının sistem içerisinde oluşabilmesi desteklenecektir.

3.1.2 Genel Bakış

CMC protokolü gereğince bir sertifika kayıt işlemi, sistemde mümkün oldukça tek bir mesaj döngüsü içerisinde oluşturulmalıdır. Sistemde oluşturulacak bir kayıt işlemine ait talep, en basit haliyle, istemcide oluşturularak sunumcuya iletilir. Gönderilen talebe karşılık yanıt, sunumcuda oluşturularak istemciye iletilir. Protokol tanımı incelendiğinde, iki farklı tipte talep ve yanıt mesajları formatları tanımlanmaktadır. Sertifika talepleri, PKCS#10 ya da CRMF tabanlı oluşabilmeyi destekleyebilmelidir. İki farklı mesaj tipinden ilki, “basit” talep olarak adlandırılan ve başka bir servise ihtiyaç duyulmayan ortamlarda, sadece PKCS#10 tabanlı mesajları içerirken, ikinci tür format, “tüm” talep olarak adlandırılır ve bir CMS zarfı içerisinde paketlenen PKCS#10 veya CRMF tabanlı mesajları içerir. Açık anahtar sertifika yanıtları, CMS’te tanımlanmış olan signedData nesnesi tabanlıdır. Herhangi başka bir servise ihtiyaç duyulmadığında yanıt, bozulmuş bir signedData nesnesi ya da bir signedData nesnesi içerisinde zarflanmış ResponseBody nesnesi olarak oluşabilir.

İstemcilerin sistemden gerçekleştirecekleri sertifika yenileme (aynı anahtarla yeni bir sertifikanın oluşturulması) ya da yeniden anahtarlama (yeni anahtarlarla yeni bir sertifikanın oluşturulması) işlemleri için herhangi bir özel servis sağlanmamıştır. Bunun yerine, bir yenileme / yeniden anahtarlama mesajı, kimlik doğrulaması var olan Sertifika Makamı'na ait sertifikalardan karşılanan, herhangi bir sertifika kayıt mesajına benzer şekilde sistem bünyesinde uygulanır. Kullanıcılarla Sertifika Makamları arasında görev yapan Yerel Kayıt Makamları, istemciden gelen kayıt mesajlarını alıp, kendilerine özel bilgileri de ekleyerek, ikinci seviyede mesajı yeniden zarflamak suretiyle, oluşan talebi Sertifika Makamı'na iletmekle sorumludur. Sistem içerisindeki konfigurasyona bağlı olarak birden fazla sayıda Yerel Kayıt Makamı bulunabilir. Yerel Kayıt Makamları ile ilgili olarak sonraki bölümlerde daha detaylı bilgiler sunulmuştur. (Bkz. Bölüm 3.1.3.19) Ek olarak işlem yönetimi, yeniden denetim, ertelenmiş sertifika yayınlama, sertifika iptal ve iptal listeleri yeniden edinme servisleri de sistem içerisinde sunulmuştur. CMC protokolüne ait talep ve yanıt formatları, daha önce belirtildiği üzere sistemde iki farklı format türünde oluşturulabilir. Basit talep yanıt çifti ve tüm talep yanıt çifti olarak adlandırılan bu formatlar ve içeriklerindeki nesne modellerine ilişkin yapısal gösterim Çizelge 3.1 'de sunulmuştur. (Bkz. Çizelge 3.1)

Basit PKI Talebi	Basit PKI Yanıtı	Tüm PKI Talebi	Tüm PKI Yanıtı
PKCS #10	CMS Mesajı	SignedData	SignedData
Sertifika Talebi Konu Adı Açık Anahtar Özellikler	SignedData SignerInfo Yok ContentInfo Boş	PKIData Kayıt Özellikleri Sertifika Talebi CMS Nesneleri Diğer Mesajlar	ResponseBody Kayıt Özellikleri CMS Nesneleri Diğer Mesajlar Diğer Sertifikalar
İmzalanır	İmzalanmaz	Talepteki anahtar çiftiyle imzalanır	SM tarafından imzalanır

Çizelge 3.1 CMC talep ve yanıt formatları

3.1.3 Protokol Bileşenleri

Bu bölümde sertifika kayıt talepleri ve yanıt mesajlarını oluşturmada kullanılacak bileşenler incelenmiştir. Burada tanımlanan nesne tipleri ve içerikleri, sonraki bölümlerde yer alan entegrasyon modellerinde ve sistem mimarisi tasarımlarında kullanılmıştır. Bileşenler, bu bölümde genel olarak tanıtılmış olup, tasarımlarda ve modellemelerde gereksinim duyulmayan detaylardan bu bölümdeki tanımlarda bahsedilmemiştir. Bu bileşenlerle ilgili detaylı ve kapsamlı bilgiler, tanımlarda refere edilen kaynaklardan edinilebilir. Tanımlanacak bileşenlerin uygulamada kullanım amaçları belirtilerek, bileşenleri oluşturan temel nesne ve metodlarla ilgili açıklamalara yer verilmiştir. Bu bölümdeki bileşenlerden öne çıkanlar, CMC protokolünün temel talep nesnesi olan PKIData ve yanıt nesnesi olarak tasarlanan ResponseBody nesneleridir. Protokolde yer alacak talep ve yanıt formatları, belirli bir içerik türüne sahip olup, bu içerik türü ContentInfo nesnesine ait tanımlamalarla belirlenir. İçerik nesnesi türleri, CMS protokolünde tanımlanmış olan signedData, envelopedData ve Data türleridir. EnvelopedData, bilginin paketlenmesini sağlarken, Data, genel olarak yapılandırılmamış bilgilerin taşınmasında kullanılır. Bu spesifikasyonda kullanılan son içerikse, CMS'te var olan, yetkilendirmede kullanılmasının yanında taleplerin ve yanıtların genel taşınma paketi olarak da kullanılabilen, signedData nesnesidir. [Myers, M ve Liu, X. et. al, 2000] Bu içerik türlerinden, sonraki aşamalarda gerçekleştirilmiş olan süreç tabanlı ve uygulama modelli tasarımlar içerisinde sıklıkla uygulanacak tür, signedData olacaktır.

Sonraki bölümlerde sunulan örnekler üzerinden detaylandırılmış olan kontrol özellikleri ise, bu yapılar içerisindeki mekanizmalarda, belirli özellik bilgilerinin (örneğin durum bilgisi) sistematik yönetim mesajları içerisinde tanımlanmalarını ve bu özelliklerin protokolde tanımlı değerlerini içerecek denetim işlevlerinin sistem içerisinde sağlanabilmesine yönelik mekanizmaları sağlar. CMC protokolünde tanımlanmış farklı talep ve yanıt formatlarından olan Basit Talep, Basit Yanıt, Tüm Talep ve Tüm Yanıt yapıları sonraki bölümde tanıtılmış, hangi durumda hangi tip format çiftinin sistem içerisinde ne tip bir konfigurasyonla uygulanacağı analiz edilmiştir.

3.1.3.1 PKIData Nesnesi

Bu protokol için tasarlanmış bir içerik nesnesi olup, talep mesajının gövdesini oluşturur. Bu gövde şu şekilde tanımlanır :

PKIData

{

controlSequence (TaggedAttribute sınıfı özellik nesnesi)

reqSequence (TaggedRequest sınıfı talep nesnesi)

cmsSequence (TaggedContentInfo sınıfı ContentInfo nesnesi)

otherMsgSequence (Mesaj nesnesi)

}

3.1.3.2 ResponseBody Nesnesi

PKIData nesnesine benzer şekilde bu protokol için tasarlanmış bir başka içerik nesnesi olup, yanıt mesajının gövdesini oluşturur. Bu gövde şu şekilde tanımlanır :

ResponseBody

{

controlSequence (TaggedAttribute sınıfı özellik nesnesi)

cmsSequence (TaggedContentInfo sınıfı ContentInfo nesnesi)

otherMsgSequence (Mesaj nesnesi)

}

3.1.3.3 Sertifika Talepleri

Sistemde gerçekleştirilebilecek sertifika talepleri, PKCS#10 ya da CRMF tabanlı olabilir. PKCS#10 tabanlı taleplerde sistemdeki sunumcular, PKCS#10 talebinin gövdesini işleyebilmelidirler. İstemciler eğer basit PKI talebi mesajı kullanıyorlarsa mutlaka, eğer tüm PKI talebi mesajı kullanıyorlarsa opsiyonel olarak PKCS#10 talep gövdesini oluşturabilmelidirler. Bu gövdenin oluşumunda istemciler, bir konu adı ve açık anahtar belirtirler. Eğer istenirse bu gövde içerisinde bir ya da daha fazla X.509 v3 sertifika uzantısı

bilgisi de eklenebilir. (Bkz. Ek 2) Eğer talep CRMF tabanlıysa, bu durumda sunumcular, CRMF talep gövdesini PKCS#10 nesnesine benzer şekilde işleyebilmelidirler. İstemciler tüm PKI talebi mesajı kullandıklarında, CRMF mesaj gövdesi oluşturmak zorunda değildir, ancak sistemdeki bir CRMF mesajı, tüm PKI talebi durumunda mutlaka bir konu adı ve açık anahtar tanımlamasına sahip olmalıdır. Sertifika talebinin belirli bir bölümünü, sayısal imza bilgisi oluşturur. Buradaki imzalama işleminde, D-H algoritması, bir anahtar uzlaşma algoritması olarak tanımlanmış olup direkt olarak gereken imza nesnesinin oluşturulmasında kullanılmaz.

3.1.3.4 Gövde Kısmı Belirteçleri (BodyPartID)

Bir PKIData veya PKIResponse mesajının her bileşeni, kendileriyle ilişkili olan bir gövde kısmı belirtecine sahiptir. Bu belirteç, ilgili nesne içeriğinde 4 oktet uzunluğunda kodlanmış olarak yer alır. Aynı nesne içerisindeki belirteçlerin tekil olma zorunluluğu bulunur. Örneğin iç içe geçmiş iki CMC mesajında olduğu gibi, farklı seviyelerdeki nesnelerde, aynı belirteçler kullanılabilir. CMC durum bilgisi (CMCStatusInfo) gibi bazı kontrol özellikleri de, benzer şekilde gövde belirteci kullanabilirler. (Bkz. Bölüm 3.1.3.14)

3.1.3.5 Kontrol Özellikleri (Control Attributes)

Bu kısımda, bir mesajın tüm işlevleri içerisinde, kontrol mekanizması olarak kullanılan, kontrol özellikleri incelenmiştir. Protokol bünyesindeki her bir kontrol özelliği, bir nesne belirteci ve bu belirtece ait bir değerden oluşur. Sunumcular, herhangi bir kontrol özelliğinin geçerli olmaması durumunda, tüm PKIData mesajının işlemini durdurmak zorundadır. Bu durumda oluşturulacak yanıt, geçersiz ya da tanımlanamamış olan kontrol özelliğini belirten bir hata mesajı olacaktır. Standart bir kontrol özelliğinin yapısı şu şekildedir :

TaggedAttribute

{

bodyPartID (Tamsayı nesne belirteci)

attrType (Özellik tipini belirten nesne belirteci)

attrValues (Özellik nesnesi içerik değerine sahip nesne)

}

Kontrol özellikleriyle ilgili detaylar ve örnek kontrol özelliklerinin içeriklerindeki nesne yapılarına yönelik bilgiler sonraki bölümlerde sunulmuştur. (Bkz. Bölüm 3.1.3.13)

3.1.3.6 İçerik Bilgi Nesneleri (ContentInfo)

Sistemdeki bir talep ya da yanıt mesajına ait cmsSequence alanı, içerik bilgi nesneleri içerebilir ya da içermeyebilir. ContentInfo nesneleri ve içerdiği tanımlamalar, CMC protokolüne ait mesajlarda, talep ve yanıt iletilerinde oldukça önemlidir. Sonraki bölümlerde yer alacak tasarımlarda da bu nesne yoğun bir şekilde kullanılmıştır. Bu nesnenin genel yapısı şu şekildedir :

TaggedContentInfo

{

bodyPartID (Tamsayı nesne belirteci)

contentInfo (İçerik tipi ile içeriği ilişkilendiren CMS/PKCS#7 tanımlı ContentInfo nesnesi)

}

Bu yapıdaki ContentInfo nesnesi, üç farklı içeriğe sahip olabilir. Bunlar signedData, EnvelopedData ve Data içerikleridir. EnvelopedData, bu protokol içerisinde, hassas nitelikteki kritik bilgilerin gizliliğini sağlamada kullanılan birincil yöntemdir. Bu protokol, EnvelopedData nesnesini kullanarak, talebin tamamının kodlanmasını sağlar. Data ise, genel olarak yapılandırılmamış bilgilerin taşınmasında kullanılır. Son içerikse, CMS'te tanımlı olan, yetkilendirmede kullanılmasının yanında taleplerin ve yanıtların genel taşınma paketi olarak da kullanılabilen signedData nesnesidir. SignedData, kayıt mesajlarının yapılandırılmasında, iki farklı amaçla kullanılabilir. Sertifika kayıt taleplerinin bir parçası olarak, PKIData'nın paketlenmesinde kullanılabileceği gibi, yanıt mesajının dışında da görev alabilir.

3.1.3.7 Diğer Mesajlar

Diğer mesajlar bölümü, gelişi güzel ya da tercihe bağlı veri nesnelerinin, mesajın bir parçası olarak taşınmasına olanak sağlar. Burada belirtilen gelişi güzel veriden kasıt, bir CMS contentInfo nesnesi içeriğinde paketlenmemiş olan bilgilerdir. Buradaki veri, bir kontrol özelliği tarafından referansa tabi tutulmadığı sürece göz ardı edilebilir.

3.1.3.8 Basit Kayıt Talep Mesajları

Önceki bölümde belirtildiği gibi CMC protokolünde gerçekleştirilebilecek olan bir kayıt talebinin en basit hali, bir PKCS#10 mesajıdır. Eğer bu türde bir kayıt talebi, imza oluşturabilecek türden bir özel anahtara yönelik olarak sistemde oluşturulmuş ise, bu durumda ilgili PKCS#10 nesnesi, bu özel anahtarla imzalanmalıdır. Eğer talep, bir D-H anahtarına yönelik olarak sistem içerisinde uygulanmaktaysa, bu durumda D-H mekanizması sistemde uygulanır. Sunumcular, bu tür basit kayıt talebi mesajlarını destekleyebilmelidir. Eğer basit kayıt talep mesajı kullanılıyorsa ve sistemde bu mesaj onaylanmışsa, sunumcuların, istemcilere basit kayıt yanıt mesajı döndürmesi gerekir. Eğer talep başarısız olarak değerledirilmişse, bu durumda sunumcudan hiçbir yanıt dönmez veya yanıt olarak sadece bir tek tüm kayıt yanıtı mesajı iletilir.

3.1.3.9 Tüm PKI Talep Mesajları

Tüm kayıt talepleri, basit kayıt talepleriyle kıyaslandığında, uygulamalarda yüksek seviyede fonksiyonalite ve esneklik sağlarlar. İstemciler kayıt işlemlerinde bu tipte mesajları kullanmak zorundadır. Sunumcular da benzer şekilde bu mesajları destekleyebilmelidir. Bir kayıt yanıtı, bütün tüm kayıt taleplerine karşı sistem içerisinden iletilebilmelidir. Tüm kayıt talep mesajı, temelde içerik olarak bir signedData nesnesi içerisine paketlenmiş haldeki, bir PKIData nesnesinden oluşmaktadır. PKIData nesnesi içeriğinde, tüm kontrol özellikleri, sertifika talepleri, CMS tabanlı nesneler ve diğer nesneler yer alır. Tüm kayıt talebi içerisindeki bileşenler, bir gövde kısmı belirteci ile işaret edilirler. PKIData nesnesini çevreleyen signedData nesnesi, imza sertifikası talebine ait özel anahtarla veya bir önceki sertifikaya ait imza anahtarıyla imzalanabilir. [Myers, M ve Liu, X. et. al, 2000]

Bu çalışma içerisinde gerçekleştirilmiş örnek entegrasyon tasarımlarında Tüm Talep formatlarından faydalanılmıştır. Özellikle üçüncü bölümde yer alan CMC protokolünün UMU-PKI v6 sistemi içerisine entegrasyonuna yönelik tasarım ve uyumlaştırma modelleri içerisindeki süreçlerde yer alan talepler, Tüm Talep formatı öngörülerek tasarlanmıştır.

3.1.3.10 Basit Kayıt Yanıt Mesajları

Sunumcular, olanaklı olan her durumda basit kayıt yanıt mesajlarını kullanmak zorundadır. İstemciler de bu tür yanıtları işleyebilmeli ve değerlendirebilmelidir. Basit kayıt yanıt mesajları, içeriğinde signerInfo nesnesi bulunmayan bir signedData nesnesinden oluşur. Talep edilen sertifikalar, signedData nesnesine ait sertifika çantası içerisinde yer alırlar. Sertifikalar sıralı bir şekilde bulunmaz. Sunumcular bir veya daha fazla kendinden imzalı sertifikaya ait

bir sertifika zinciri oluşturacak şekilde, tüm orta düzey sertifikalara sahip olmalıdır. Sunumcu ek olarak, sertifika iptal listesi çantasında yer alan listeleri de geri iletebilir.

3.1.3.11 Tüm PKI Yanıt Mesajları

Eğer bir tüm PKI talep mesajı, olumsuz olarak sonuçlanmışsa ya da gelen sertifikalardan farklı bir takım ek servisler gerektiğinde, sunumcular, tüm yanıt mesajı döndürmek durumundadır. Sunumcular, başarısızlıkla sonuçlanan basit PKI talepleri karşılığında da, opsiyonel olarak tüm PKI yanıtı döndürebilirler. İstemciler bir tüm PKI yanıt mesajını işleyebilmeli ve değerlendirebilmelidirler. Bir tüm kayıt yanıt mesajı, bir responseBody nesnesini enkapsüle eden signedData nesnesinden oluşur. Burada responseBody nesnesi içerisindeki kontrol özellikleri, tüm CMS nesnelerinin önünde yer almalıdır. Sistemde hak verilen sertifikalar, kayıt yanıt mesajının, ilgili signedData nesnelerinin sertifikalar alanı içerisinde yer alır.

İstemciler, tüm kayıt yanıt mesajlarında ve tüm PKI yanıt mesajları içerisinde, herhangi bir nedenle içerikteki sertifikaların, sıralı bir düzende olduğunu varsaymamalıdır. Sunumcular, sadece yeni yayınlanan sertifikalar yerine, sertifika zinciri içerisinde yer alan tüm orta düzey sertifikaları da dahil etmelidirler. Sunumcular aynı zamanda sertifika iptal listesi çantasındaki ilgili CRL listelerini de iletebilir. Benzer şekilde sunumcular, kendinden imzalı sertifikaları da aynı yöntemle içeriğe dahil edebilirler. Bu çalışma içerisinde gerçekleştirilmiş olan entegrasyon tasarımlarındaki yanıt süreçlerinde, Tüm Yanıt formatlarından faydalanılmıştır. Özellikle üçüncü bölümdeki CMC protokolünün UMU-PKI entegrasyonu tasarımlarındaki süreçlerde yer alan yanıtlar, Tüm Yanıt formatı öngörülerek tasarlanmıştır.

3.1.3.12 Bir PKI Mesajının Kodlanması

Bir talep ya da yanıtın, yetkisiz birimler tarafından içeriğinin elde edilmesinin engellenmesi amacıyla, sistem içerisinde kodlanması gerekmektedir. Burada sağlanan gizlilik, açık anahtar mesajının (signedData) bir CMS EnvelopedData nesnesi içerisine paketlenmesiyle sağlanır. EnvelopedData içerisinde paketlenen bilgi, protokol içerisinde id-signedData olarak belirtilir. Eğer bir PKI mesajına, zarflanmış bilgi katmanı eklenecekse, ikinci imza katmanı, zarflanmış bilgi katmanının dışında yapılandırılmalıdır.

Bu opsiyonların detaylarını içeren çizelge, aşağıdaki gösterimde sunulmuştur. (Bkz. Çizelge 3.2)

Normal	Opsiyon 1	Opsiyon 2
SignedData PKIData	EnvelopedData SignedData PKIData	SignedData EnvelopedData SignedData PKIData

Çizelge 3.2 Mesaj kodlama opsiyonları

Bu çizelgede belirtilen opsiyonlardan, Opsiyon 1 ve Opsiyon 2, hassas ya da kritik öneme sahip verinin kodlanması sonucunda, bu tür bilgilerin dışarıya sızmasını engellemeye yönelik olarak oluşturulmuştur. Yerel Kayıt Makamları, bu tip mesajlardaki zarflanmış veride yer alan zarf mekanizmasını kaldırıp, veri içeriğini işleyebilirler. Açık anahtar mesajları, kodlanmış halde ya da açık olarak iletilebilir. Sunumcular bu şekilde gösterilmiş her üç sürümü de destekleyebilmelidir.

3.1.3.13 Kontrol Özellikleri

Kontrol özellikleri, talep ve yanıt mesajlarının her ikisinde de ortak olarak taşınan bileşenlerdendir. [Myers, M ve Liu, X. et. al, 2000] Her bir kontrol özelliği, bu özelliği tanımlayan bir belirteç (OID) ve bu özelliğin bilgisini içeren veri kısmından oluşur. CMC protokolünde, sertifika taleplerine ait yapıların bir parçası olarak, oldukça fazla sayıda kontrol özelliği tanımlanmıştır. Tanımlanan bu kontrol özellikleri, genelde özellik sertifikaları için uygun olmamakla birlikte, bu tür sertifikaların desteklenebilmesine yönelik protokol bünyesinde eklentiler ve güncelleme çalışmaları geliştirilmektedir.

Bazıları detaylı olarak sonraki bölümlerde incelenmiş olan örnek kontrol özelliklerine ait tanımlamalar, Çizelge 3.3'te sunulmuştur. (Bkz. Çizelge 3.3)

Kontrol Özelliği	OID	Gösterim
cMCStatusInfo	id-cmc 1	CMCStatusInfo
identification	id-cmc 2	UTF8String
identityProof	id-cmc 3	Octet String
dataReturn	id-cmc 4	Octet String
transactionId	id-cmc 5	Integer

Çizelge 3.3 Örnek kontrol özelliği tanımlamaları

3.1.3.14 CMC Durum Kontrol Özelliği (CMCStatusInfo)

CMC durum kontrol özelliği, tüm PKI yanıt mesajlarında, bir istemci talebi karşılığında iletilir. Sunumcular, tek bir mesaj gövdesi içerisinde birden fazla CMC durum kontrol özelliği bilgisini yayabilirler. İstemciler, yanıt mesajı içerisinde birden çok durum kontrol özelliği bilgisini işleyebilmelidir. Kontrol özelliği tasarımı şu şekilde gösterilebilir :

CMCStatusInfo

{

cMCStatus (Durum bilgisini gösteren CMCStatus nesnesi)

bodyList (Durum bilgisinin hangi nesneleri etkileyeceği bilgisini içeren BodyPartID nesne belirteçleri listesi)

statusString (Taleple ilgili ek bilgileri içeren UTF8String formatında nesne)

otherInfo

{

failInfo (Hata detay bilgisini içeren CMCFailInfo nesnesi)

pendInfo (Alt kısımda detaylandırılacak PendInfo nesnesi)

}

}

PendInfo

{

pendToken (String tipinde askıdaki talep durum sorgulamada kullanılan jeton nesnesi)

pendTime (Sunumcunun talep durum sorgulamalarını kabul edeceği zamanı belirten GeneralizedTime tipinde zaman nesnesi)

}

3.1.3.15 Bilgi Dönüş Kontrol Özelliği (DataReturn)

Bu kontrol özelliği, istemcilerin keyfi bir takım bilgileri sunumcuya göndermelerine ve sonrasında kayıt yanıt mesajı içeriğinde almalarına imkan sağlar. Aynı kontrol, hem taleplerde hem de yanıtlarda kullanılabilir. Eğer bu kontrol özelliği bir kayıt mesajı içerisinde iletilmişse, bu durumda sunumcu, kayıt yanıt mesajı içerisinde bu bilgiyi döndürmelidir. Eğer bilgi dönüş kontrol özelliğindeki bilginin gizli olması söz konusu olursa, bu durumda istemcinin kontrol bilgisi üzerinde kodlama uygulaması beklenir.

3.1.3.16 İşlem Yönetim Kontrol Özelliği (TransactionID)

Sistem içerisinde gerçekleştirilen her işlem, bir işlem belirteci ile diğer işlemlerden ayırt edilir. Bu kontrol özelliğinin kullanılması durumunda istemciler, işlem belirteçleri üreterek sunumcudan bu işlemlere ait tamamlanma mesajı gelene kadar, bu işleme ait değeri saklarlar. Sunumcular, kendilerine gelen işlem belirteçlerini, gönderecekleri yanıt içerisinde iletirler. Burada transactionID kontrol özelliği, bir operasyonu işaret etmektedir. Operasyonun durumuyla ilgili olarak istemci ve sunumcu arasında kullanılır. İstemciler gerektiğinde talep mesajları içerisine bu kontrol özelliğini ekleyebilirler. Eğer talepte bu kontrol mevcutsa, bu taleple bağlantılı olan tüm talep ve yanıt mesajları, bu kontrolü içermek durumundadır.

3.1.3.17 Sertifika Kabul Doğrulama Kontrol Özelliği

Bazı Sertifika Makamları, istemcilere atanan sertifikaların, istemcilerce uygulanabilir olup olmadığının doğrulanması için, istemcilerden olumlu bir konfirmasyon bilgisi beklerler. Sertifika kabul doğrulama kontrolü, bu amaçla kullanılmaktadır. Eğer bir sertifika talebinin CMCStatusInfo alanı, confirmRequired olarak tanımlanmışsa, istemci, konfirmasyon bilgisini göndermek durumundadır. İstemciler gönderdikleri konfirmasyon bilgisinin, sunumcular tarafından alındığına dair doğrulama mesajını da beklerler. Sunumcular bir tüm kayıt yanıt mesajı ile bu beklentiye cevap verir.

3.1.3.18 Uzantı Ekleme Kontrol Özelliği

Bu kontrol özelliği, temelde Yerel Kayıt Makamları tarafından, sertifikalar içerisine ek uzantı bilgilerinin eklenmesinde kullanılırlar. Bu özelliğe ait sözdizimi ve detaylar şu şekildedir :

AddExtensions

{

pkiDataReference (İlgili PKIData talebine ait referans belirteç nesnesi)

certReferences (PKIData içerisindeki sertifikalar dizisi nesnesinde yer alan sertifikalara ait referans belirteçleri dizisi)

extensions (Refere edilen sertifika taleplerine uygulanacak uzantıları içeren dizi)

}

3.1.3.19 Yerel Kayıt Makamları

Yerel Kayıt Makamları, açık anahtar altyapılarında, sertifika tabanlı uygulamaları değerlendiren, onaylayan ya da reddeden birimlerdir. Aynı zamanda bu makamlar, sertifikaların iptal edilmelerine yönelik onay mekanizmasında da görev alabilirler. Yerel Kayıt Makamları, üst düzey bir Sertifika Makamı yerine ve bu makama ait yetkilendirmenin altında görev alırlar. Yerel Kayıt Makamları, son kullanıcılar ya da istemcilerle, Sertifika Makamları arasında bulunur. Son kullanıcı tarafından bakıldığında bu makamlar, Sertifika Makamı olarak görülürken, sunumcu tarafında bu makamlar birer istemcidir. Kayıt mesajlarını alarak, işleme tabi tutarlar ve Sertifika Makamları'na iletirler. Bu işlemler, birden çok kayıt mesajını gruplama, tüm taleplere özel ya da standart sertifika uzantılarının eklenmesi, özel anahtarların arşivlenmesi ya da taleplerin farklı Sertifika Makamları'na

yönlendirilmesi olarak sıralanabilir. Yerel Kayıt Makamları, bir kayıt mesajını aldıklarında üç seçeneği değerlendirirler. Bunlar, mesaj üzerinde herhangi bir değişiklik yapmadan iletme, mesaja yeni bir katman ekleyerek yeniden paketleme ve son olarak da bir ya da daha çok katmanı mesajdan çıkartıp yeni bir katman ekleyerek mesajı yeniden paketlemedir. Eğer bir Yerel Kayıt Makamı, aldığı mesaja yeni bir katman eklemek suretiyle yeniden paketlerse, yeni bir PKIData nesnesi oluşturmuş olur. Farklı mesajları bir arada gruplarsa, her kayıt mesajı Yerel Kayıt Makamı'na ait PKIData nesnesinde ilgili tüm kontrol özellikleriyle beraber uygun bir bölgeye yerleştirilir. Yerel Kayıt Makamı, bir kayıt mesajını işleme aldığında, sertifika gövde kısmında hiçbir değişiklik yapmamalıdır. Bunun nedeni, sertifika gövdesinde yapılacak herhangi bir değişikliğin, talepte yer alan imzanın geçersiz olmasına yol açmasıdır. Bazı durumlarda Yerel Kayıt Makamı'ndan, gelen mesajdaki katmanları çıkartması beklenir. Bu işleme örnek olarak gelen kodlanmış mesajdaki kodlamayı kaldırarak veriyi elde etme söylenebilir.

3.2 Sertifika Doğrulamada SCVP Protokolü

SCVP protokolü, bir istemcinin, sertifika yolu yapılandırma ve doğrulama işlevlerini, bir sunumcuya devredebilmesine imkan sağlamaktadır. Sertifika yolu yapılandırma ya da doğrulama (örneğin sertifika yolunda bulunan sertifikaların iptal edilmiş olup olmadığı bilgisinin alınması), bir ya da birden fazla güven noktasına (Bkz. Bölüm 1.6.11) sahip bir doğrulama politikası gereğince gerçekleştirilir.

Sertifika doğrulama karmaşık bir operasyondur. Eğer sertifika kullanımı geniş kapsamlı ortamlarda uygulanmak üzere yapılandırılacaksa, uygulamanın sertifikaları kabulü esnasında gerçekleştireceği işlevlerin sayısı asgari tutulmalıdır. Günümüzde açık anahtar altyapısını destekleyen birçok uygulama bulunmasına karşın, sertifikaların yapılandırılması ve doğrulanması konusundaki yükü hafifletecek çözümler, yeterince geliştirilememiştir. SCVP protokolü, iki farklı uygulama grubu için bu yükü azaltabilecek çözümler sunar. İlk grup uygulamaların iki amacı vardır. Sertifikanın içerisinde bulunan kimlik tanımlamasının ilgili açık anahtara ait olup olmadığının doğrulanması ve bu anahtarın uygulamada planlanan amaca uygun olarak kullanılıp kullanılamayacağıdır. Bu gruptaki istemciler, sertifika yolu yapılandırma ve doğrulama işlemlerini bir SCVP sunumcusuna devrederler. Bu işleme, delege edilmiş yol doğrulama adı verilir. İkinci grup uygulamalarsa, sertifika doğrulama yapabilen ancak güvenilir ve etkin bir sertifika yolu yapılandırma metoduna sahip olmayan uygulamalardır. Bu gruptaki istemciler, sertifika yolu yapılandırma görevini bir SCVP

sunumcuya devrederken, sertifika yolunun doğrulanması işlevini devretmezler. Bu işlem, delege edilmiş yol yapılandırma olarak adlandırılır.

SCVP protokolünün başlıca amacı, sertifika yolu yapılandırma veya doğrulama işlevlerini bir sunumcuya delege ederek, organizasyon içerisindeki doğrulama politikalarını merkezi olarak yönetebilme ve bu sayede PKI tabanlı uygulamaların entegrasyonunu kolaylaştırmadır. [Malpani, A. ve Turner, S. 2000] SCVP, sertifika işlemlerinin büyük bir kısmını daha çok kendisi gerçekleştiren istemciler tarafından, bu işlemleri başka bir birime delege etmek üzere kullanılır. Ancak sunumcuya tam olarak güven sağlandığı durumlarda, sertifika yolu yapılandırma ve doğrulama işinin delege edilmesinde ve organizasyon içerisindeki politikaların doğru şekilde uygulandığının garantiye alınmasında da fayda sağlar.

İstemciler tarafından tam olarak güven sağlanamayan SCVP sunumcular da, istemcilere sertifika yolu sağlayabilirler. Aynı zamanda bu sunumcular, başka bir SCVP sunumcudan gelen sertifika yolunun doğrulanması işleminde, istemcilere sertifika iptal listesi bilgilerinin sağlanmasında da görev alabilirler. Bu tip servisler, ara sertifikaları bulup doğrulayacak, sertifika iptal listelerini ve OCSP yanıtlarını alacak protokollere sahip olmayan istemciler için kritik değer taşımaktadır. Güven sağlamış SCVP sunumcular, sertifika yolu yapılandırma ve doğrulama işlemlerini istemciler adına gerçekleştirirler. Bu servisleri kullanan bir istemci için, SCVP sunumcuya sağladığı güven ilişkisi tam olmalıdır. Bir istemcinin, bir SCVP sunumcuya güvenmesi için iki ana gerekçe söylenebilir. Bunlardan birincisinde istemci, kendine ait sertifika yolu doğrulama uygulamasını, aldığı her sertifika için çalıştırma yükünü fazla görebilir. Diğer bir gerekçeyse, organizasyonun, doğrulama politikalarını merkezi olarak yönetme gereksinimidir.

3.2.1 Genel Bakış ve Protokol Gerekleri

Delege edilmiş yol doğrulama ve delege edilmiş yol yapılandırma işlevleri için tanımlanmış tüm gerekler, SCVP protokolü için de geçerlidir. [Malpani, A. ve Turner, S. 2000] SCVP protokolünün delege edilmiş yol yapılandırma desteğinin olması durumunda, sertifika yolu yapılandırıldığında, delege edilmiş yol doğrulama yanıtının aynısı edinilirken, iptal bilgilerinin bir kısmı veya tümü ele geçmeyecektir.

SCVP protokolü basit bir talep-yanıt modelini içerir. SCVP istemcisi, bir talepte bulunarak sunumcuya iletir ve sonrasında sunumcu yanıt oluşturarak istemciye gönderir. Tipik bir SCVP uygulaması, HTTP protokolü üzerinde gerçekleştirilebilirken, benzer şekilde e-posta mekanizmalarında da uygulanabilir. SCVP, iki tip talep-yanıt çiftini içermektedir. Bunlardan

ilki ve en sık kullanılanı, sertifika doğrulamaya yönelik olarak uygulanan talep-yanıt çiftidir. İkinci talep-yanıt çiftiyse, doğrulama politikalarına ait listelere karar vermede ya da belirli bir SCVP sunumcunun destekleyebildiği varsayılan parametre bilgilerinin belirlenmesinde kullanılır. [Malpani, A. ve Turner, S. 2000]

3.2.2 Doğrulama Politikaları

RFC 3379'da tanımlanmış olan bir doğrulama politikası, bir sertifikanın doğrulanmasında, SCVP sunumcu tarafından kullanılacak kural ve parametreleri belirtir. SCVP protokolünde sunumcu tarafından uygulanacak doğrulama politikası, istemci tarafından gönderilen talep içerisinde, ek parametrelere gerek duyulmadan tam olarak belirtilebileceği gibi, istemci tarafından bazı ek parametrelerle de aynı talep içerisinde tanımlanabilir. Politikaların tanımlanması genel olarak oldukça uzun ve karmaşık bir süreç alabilmesine karşın, bazı politikalar daha basit halde tanımlanabilir. Eğer nesne belirteci, hem kullanılacak olan algoritma tanımını, hem de doğrulama politikasına ait ek parametre bilgilerini içerebilirse, bu durumda talep oldukça basitleşebilir. Eğer doğrulama politikası, gerekli olan tüm parametreleri kendi içerisinde tanımlamışsa, bir SCVP talebi, sadece doğrulanması istenen sertifika ve ilgili doğrulama politikası bilgilerini içerecektir. Bir sunumcu, destekleyebildiği doğrulama politikalarına ait referans bilgilerini yayınlar. Bu politikaların üstüne yazılabilecek parametrelere sahip olması durumunda sunumcu, bu parametrelere ait varsayılan değerleri de sistemde duyurur. Eğer sunumcu tarafından duyurulan parametre değeri, istemci için uygun bir değerse, istemci, talepten bu parametre değerini çıkartarak talebi basitleştirebilir. Temel sertifika yolu prosesi algoritması, bir veya birden fazla güven bağlantı noktasını bir giriş bilgisi olarak ele alabilir. İstemci, sadece belirli bir Sertifika Makamı'na ait bir sertifika talebinde bulunursa, tek bir güven bağlantı noktası belirtilir. Birden fazla Sertifika Makamı'na yönlü yolları talep eden istemci, bir grup güven bağlantı noktası tanımlamak durumundadır.

3.2.3 Doğrulama Algoritması

Doğrulama algoritması, istemci ve sunumcu tarafından üzerinde anlaşma sağlanan ve bir nesne belirteci ile tanımlanan bilgidir. Algoritma, sertifikanın geçerli olup olmadığının anlaşılması için sunumcu tarafından gerçekleştirilmesi gereken kontrolleri tanımlar. Bir doğrulama algoritması, doğrulama politikasına ait parametrelerden biridir. SCVP protokolü, PKIX-1 de tanımlı olan ve istemcinin doğrulanacak sertifikalarla ilgili ek bilgileri talep edebilmesini sağlayacak, temel bir doğrulama algoritması tanımlar. Gerektiğinde ek kontrollerin de tanımlanabileceği yeni doğrulama algoritmaları da, bu yapı içerisinde

oluşturulabilir. Bu tip yeni algoritmalar, yeni bir takım parametrelerin de tanımlanabilmesine olanak sağlar. Bu parametrelere ait değerler, algoritmayı kullanan bir doğrulama politikası tarafından tanımlanabilir ya da direk olarak istemci tarafından iletilen talep bilgisi içerisinde belirtilebilir. Uygulama bağımlı doğrulama algoritmaları, ek olarak temel doğrulama algoritmasınca kapsanmayan bir takım özel tanımlamaları da içerebilir.

3.2.4 Doğrulama Gerekleri

Sertifika yolunun belirli bir politika gereğince geçerli ve doğru olarak değerlendirilebilmesi için PKIX-1'de tanımlı geçerlilik kriterlerini sağlaması ve tüm doğrulama politikası yasaklarına uygun olması gerekmektedir. Sertifika iptal kontrolü, PKIX-1 gereklerinde opsiyonel olarak belirtilmiştir. İstemciler, taleplerinde iptal kontrolünün olup olmayacağını ve iptal bilgisinin yanıt içerisinde yer alıp almayacağını bildirirler. Sunumcular kendi işleyebilecekleri iptal bilgisi kaynaklarını (iptal listeleri) belirlemek zorundadır. İstemci, bir talep oluşturarak sunumcuya iletir ve sonrasında sunumcu bu talebe karşılık yanıt oluşturarak istemciye gönderir. Tipik SCVP uygulaması HTTP, e-posta ya da sayısal imza nesnelerini taşıyabilecek farklı bir uygulamayla da sağlanabilir. SCVP iki adet talep yanıt çifti sunar. İlk talep yanıt çifti doğrulamada kullanılırken, ikincisi, doğrulama politikalarına ait listeye karar vermek ve sunumcu tarafından desteklenen parametrelerde kullanılır.

3.2.4.1 Doğrulama Talebi

Bir SCVP istemcisinden sunumcuya gönderilen talep, tek bir CVRequest nesnesi olmak zorundadır. SCVP talepleri korumasız ve korumalı olarak adlandırılan iki ayrı forma sahiptir. Korumalı talepler, istemcinin sunumcuya karşı gerçeklenmesinde ya da talep-yanıt çifti üzerinde anonim istemci entegrasyonunda kullanılır. Burada belirtilen koruma, bir sayısal imza tarafından ya da mesaj geçerleme kodu tarafından sağlanır. Eğer istemciye ait açık anahtar, sertifika içerisindeyse, bu durumda istemcinin gerçeklenmesinde kullanılabilir. Bir sunumcu, kendisine gelen tüm taleplerin korumalı olmasını zorunlu tutup, korumasız gelen talepleri dikkate almayacağını belirtebilir. Alternatif olarak bir sunumcu, korumasız talepleri işleyebileceğini de belirtebilir. Korumasız bir talep, bir CMS ContentInfo nesnesi içerisine enkapsüle edilmiş CVRequest nesnesinden oluşur.

SCVP protokolü gereğince sistem içerisinde oluşturulacak taleplere yönelik genel yapı aşağıdaki nesne hiyerarşisinde belirtilmiştir. Bu yapı içerisinde öncelikli olarak ContentInfo nesnesi tasarlanmış, sonrasında bu nesnenin içeriğinde yer alan ve önceki bölümde üzerinde durulan CVRequest nesnesinin yapısı incelenmiştir.

Talebin en temel nesnesi olan CVRequest nesnesi içerisindeki en önemli bileşen olarak nitelendirilen Query nesnesi de benzer şekilde yapısal olarak sunulmuştur.

ContentInfo

{

contentType (Talep içerik tipini gösteren nesne belirteci)

contentCVRequest (Alt kısımda detaylandırılacak CVRequest nesnesi)

}

CVRequest

{

cvRequestVersion (Talepte kullanılan CVRequest nesnesi sürümünü içeren tamsayı nesne)

query (Alt kısımda detaylandırılacak Query nesnesi)

requestorRef (Opsiyonel olarak talep sahibi istemciyi belirten referans string tipinde nesne)

requestNonce (Opsiyonel olarak talebi belirten referans string tipinde belirteç nesnesi)

requestExtensions (Opsiyonel olarak talepteki uzantı bilgilerini içeren string belirteç nesnesi)

dhPublicKey (Opsiyonel olarak talepte yer alabilen dhPublicKey tipinde D-H açık anahtarı)

}

Query

{

queriedCerts (CertReferance tipinde talepte yer alan sertifikayı işaret eden sertifika belirteci)

checks (Sunumcudan gerçekleştirilmesi talep edilen işlem türlerini içeren nesne belirteçleri)

wantBack (queriedCerts alanındaki sertifikalarla ilgili sunumcudan beklenen bilgileri içeren nesne belirteçleri dizisi)

validationPolicy (Doğrulama işleminde sunumcunun uygulayacağı politikayı belirten nesne belirteci)

responseFlags (Opsiyonel olarak sunumcunun yanıtında istenen ek bilgileri içeren nesne)

serverContextInfo (Opsiyonel olarak daha önce aynı sunumcudan bir doğrulama işlemi gerçekleştirilmiş ise bu işleme yönelik belirteç nesnesi)

validationTime (Opsiyonel olarak istemci tarafından sunumcunun doğrulamayı gerçekleştirmesi istenen GeneralizedTime zaman nesnesi)

intermediateCerts (Opsiyonel olarak sunumcunun sertifika yolu oluşturma işleminde yardımcı olarak kullanılabilecek ek sertifikaları içeren sertifika belirteçleri nesneleri dizisi)

revInfos (Opsiyonel olarak sunumcunun sertifika doğrulama işleminde yardımcı olarak kullanılabilecek iptal listelerini içeren sertifika iptal listeleri nesneleri dizisi)

producedAt (Opsiyonel olarak istemci tarafından sunumcudan bellekteki bilgileri kullanabilmesine olanak sağlayacak geçmiş doğrulama bilgilerinden faydalanmaya yönelik GeneralizedTime tipinde geçmiş zaman bilgilerini içeren zaman nesnesi)

queryExtensions (Talebe ek olarak farklı bilgilerin yer alabileceği uzantıları içeren geleceğe yönelik SCVP eklentilerinde kullanılmak üzere tasarlanmış uzantı nesnesi)

}

Korumalı talepler, daha sonra bir ContentInfo nesnesi içerisine enkapsüle edilecek olan, bir SignedData ya da AuthenticatedData nesnesi içerisine enkapsüle edilmiş bir CVRequest nesnesinden oluşur. Burada SignedData ya da AuthenticatedData nesnesine ait EncapsulatedContentInfo alanı, id-ct-scvp-certValRequest değerine sahip bir eContentType alanından ve DER kodlamalı CVRequest talebini içeren bir eContent alanından oluşmaktadır. [Malpani, A. ve Turner, S. 2000] Talep, sayısal olarak imzalanacağı zaman, SignedData nesnesi kullanılır. AuthenticatedData nesnesi ise, sistemde, mesaj geçerleme kodunun kullanıldığı durumlarda gereklidir. Tüm SCVP istemcileri, imzalanmış talepler ya da yanıtlar için, SignedData nesnesini kullanmak zorundadır. Aynı zamanda istemciler, mesaj geçerleme kodu kullandıkları durumlarda, AuthenticatedData nesnesini kullanmalıdırlar. Eğer istemci, SignedData nesnesini kullanacaksa, SCVP talebini imzalamaya uygun ve belirli bir özel kimliğe, bir sertifika aracılığıyla bağlı olan bir açık anahtara sahip olmalıdır. İstemci, talebi enkapsüle eden SignedData nesnesi içerisine, kendisine ait sertifikayı işaret edecek olan bir belirteç koymalıdır. İstemciye ait talep, istemciye ait sertifika bilgisini içerebilir ancak talep boyutunu azaltmak için bu sertifikanın istemci tarafından talepten çıkartılması da mümkündür. Sertifikaların doğrulanmasına yardımcı olması için SCVP sunumcu tarafından talep içerisine ek bir takım sertifikalar da eklenebilir. İstemci, talebi enkapsüle eden

AuthenticatedData nesnesi içerisine açık anahtarını ya da kendi açık anahtarına sahip belirli bir referans bilgisini koyabilir. SignedData, AuthenticatedData ve ContentInfo nesnelerine ait sözdizimi ve semantikler CMS içerisinde tanımlanmıştır. CVRequest nesnesi istemciye ait talebi içerir. [Malpani, A. ve Turner, S. 2000] CVRequest nesnesine ait sözdizimi, genel hali ile şu şekilde tanımlanır :

CVRequest

{

cvRequestVersion (Talepte kullanılan CVRequest nesnesi sürümünü içeren tamsayı nesne)

query (Önceki bölümde detaylandırılmış Query nesnesi)

requestorRef (Opsiyonel olarak talep sahibi istemciyi belirten referans string tipinde nesne)

requestNonce (Opsiyonel olarak talebi belirten referans string tipinde belirteç nesnesi)

requestorName (Opsiyonel olarak imzalı taleplerde sunumcu tarafından verilecek yanıtta yer alacak olan istemci kimlik bilgisi nesnesi)

responderName (Opsiyonel olarak talepte istemci tarafından sunumcuyu belirtecek string türünde nesne)

requestExtensions (Önceki bölümde üzerinde durulmuş uzantı nesneleri dizisi)

signatureAlg (Opsiyonel olarak imzalamada kullanılacak algoritmayı belirten algoritma nesne belirteci)

hashAlg (Opsiyonel olarak sunumcunun yanıtta uygulayacağı özet bilgisini hesaplamakta kullanılacak algoritmayı belirten algoritma nesne belirteci)

requestorText (Talep eden istemci tarafından sunumcunun vereceği yanıtta kendisine aynen geri gelmesini beklediği kontrol amaçlı string tipinde nesne)

}

3.2.4.2 Doğrulama Yanıtı

SCVP sunumcunun istemciye yönelik yanıtı, tek bir CVResponse nesnesidir. Bir SCVP yanıtı, birçok farklı forma sahiptir : [Malpani, A. ve Turner, S. 2000]

- TLS veya benzeri bir güvenli taşıma yöntemi kullanılarak yapılmış bir talebe karşılık olumlu yanıt. Bu yanıtlar sunumcu tarafından korunmamalıdır.
- İçeriğinde protectResponse tanımlaması FALSE olarak belirtilen bir talebe yönelik olumlu yanıt. Bu yanıtlar sunumcu tarafından korunmamalıdır.
- Sunumcu diğer tüm olumlu yanıtları korumak durumundadır. Eğer sunumcu politika gereği korumalı olumlu bir yanıt döndüremiyorsa, bir hata yanıtı döndürür.
- TLS veya benzeri bir güvenli taşıma yöntemi kullanılarak yapılan bir talebe yönelik olumsuz hata yanıtı. Bu yanıtlar sunumcu tarafından korunmamalıdır.
- İçeriğinde protectResponse tanımlaması FALSE olarak belirtilen bir talebe yönelik olumsuz hata yanıtı. Bu yanıtlar sunumcu tarafından korunmamalıdır.
- MAC geçerliliği olan durumda, AuthenticatedData talebine yönelik hata yanıtı.
- Geçerlenmiş bir talebe yönelik bir hata yanıtı. Sunumcu bu yanıtı korumalıdır.
- Tüm diğer hata yanıtları sunumcu tarafından korunmamalıdır.

Bir talebe ait tüm gerekleri yerine getiren sunumcu, talebe olumlu yanıt verir. Eğer sunumcu, doğrulama politikasını kullanarak, bir sertifika yolu oluşturabilirse veya talep edilen tüm parametreleri yerine getirebilirse, talebe olumlu yanıt verebilir. Aksi halde hata mesajıyla yanıtlar. Korumalı talepler ve yanıtlarda sunumcular, zorunlu olarak SignedData ve zorunlu olmasalar da AuthenticatedData nesnelerini destekleyebilmelidir. İstemci tarafından yanıtın korumalı olması talep edilirse, SCVP sunumcular bu nesneleri kullanmalıdırlar. Sunumcu, korumalı bir talebe karşılık, korumalı bir yanıt veriyorsa, talepteki koruma mekanizmasının aynısını, vereceği yanıtta kullanmak zorundadır. Korumasız yanıtlar için genel yapı şu şekildedir :

ContentInfo

{

contentType (Doğrulama yanıtı tipini belirten nesne belirteci)

content (Sonraki bölümde detaylandırılacak CVResponse nesnesi)

}

Korumalı yanıtlarda CVResponse bilgisi, daha sonra bir ContentInfo içerisine enkapsüle edilecek olan bir SignedData ya da AuthenticatedData içerisine enkapsüle edilmiş halde bulunur. SCVP sunumcu, kendine ait sertifikaları, SignedData içerisindeki, sertifika alanına dahil etmelidir. Bu alanda başka sertifikalar da yer alabilir. Ayrıca sunumcu, SignedData içerisindeki CRL alanında, iptal listelerini tutabilir. SignedData içerisinde unsignedAttrs tanımı olmamalı ve signerInfos alanında tek bir SignerInfo bilgisi yer almalıdır. SignerInfo içerisindeki signedAttrs bilgisi, CMS'te tanımlı content-type ve message-digest özelliklerini içerirken, ESS'de tanımlanmış signing-certificate özelliğini de içermelidir. Bu özelliğin içinde, sertifika belirteçlerine göre yapılmış olan sıralamada, ilk sıradaki sertifika, SCVP sunumcuya ait sertifika olmalıdır. Diğer sertifikalar tercihe göre sıralanabilir. CVResponse, sunumcuya ait yanıt bilgisini içerir. [Malpani, A. ve Turner, S. 2000]

CVResponse

{

cvResponseVersion (Önceki bölümde aktarılan cvRequestVersion nesnesi ile aynı sürüm değerini içerecek tamsayı tipinde sürüm belirteç nesnesi)

serverConfigurationID (Sunumcu tarafından yanıtın oluşturulması zamanında sunumcunun sahip olduğu konfigurasyonu belirten tamsayı tipinde nesne)

producedAt (Sunumcunun yanıtı oluşturma zamanını belirten GeneralizedTime tipinde zaman nesnesi)

responseStatus (Sunumcu tarafından istemciye iletilecek olan ve istemcinin talebine ait durum bilgisini içeren sayısal durum kodlarına sahip içerik nesnesi)

respValidationPolicy (Sunumcunun talebi gerçeklemesine yardımcı olacak doğrulama politikasına ait belirteci işaret eden nesne)

requestRef (Sunumcudan gönderilen yanıtın hangi talebe ait olduğunu istemciye bildiren referans nesnesi)

requestorRef (Opsiyonel olarak talep sahibi istemciyi belirten referans string tipinde nesne)

requestorName (Opsiyonel olarak imzalı taleplerde sunumcu tarafından verilecek yanıtta yer alacak olan istemci kimlik bilgisi nesnesi)

replyObjects (Yanıtta istemciye iletilen nesnelere ait referans bilgilerini içeren nesne)

respNonce (İletilen yanıtla talebi ilişkilendiren, talepte ve yanıtta aynı değerlere sahip olması gereken string tipinde nesne)

cvResponseExtensions (Yanıta ek olarak farklı bilgilerin yer alabileceği uzantıları içeren geleceğe yönelik SCVP eklentilerinde kullanılmak üzere tasarlanmış uzantı nesnesi)

requestorText (Talep eden istemci tarafından sunumcunun vereceği yanıtta kendisine aynen geri gelmesini beklediği kontrol amaçlı string tipinde nesne)

}

3.2.5 Sunumcudan Politika Talebi

Bir SCVP istemcisi, ValPolRequest nesnesini kullanarak, SCVP sunumcuya ait politikaları, konfigurasyon bilgilerini ve sunumcunun desteklediği doğrulama politikaları listesini talep edebilir. Bu talep, bir ContentInfo alanı içerisinde enkapsüle edilmiş ValPolRequest nesnesinden oluşur. İstemci, talebi imzalamaz. Buna uygun sunumcu implementasyonları, politika talebini işlemek zorundadır.

ContentInfo

{

contentType (Doğrulama yanıtı tipini belirten nesne belirteci)

content (Sonraki bölümde detaylandırılacak ValPolRequest nesnesi)

}

ValPolRequest

{

vpRequestVersion (Önceki bölümde incelenmiş CVRequestVersion ile aynı tanım ve özelliklere sahip tamsayı tipinde sürüm bilgisi nesnesi)

requestNonce (Opsiyonel olarak talebi belirten referans string tipinde belirteç nesnesi)

}

3.2.6 Doğrulama Politikası Yanıtı

ValPolRequest talebine karşılık sunumcu, bir ValPolResponse nesnesi ile yanıt verir. Bu yanıt, tek ve gelen talebe özgü olmayıp, aynı ValPolResponse bilgisi içerisinde birden çok talebe yanıt olarak kullanılabilir. Sunumcu yanıtı kendi sertifikasıyla imzalar. Yanıt, daha sonra bir ContentInfo tarafından enkapsüle edilecek bir SignedData içerisine enkapsüle edilmiş ValPolResponse nesnesinden oluşur.

ValPolResponse nesnesine ait yapı ve detaylar şu şekilde tanımlanabilir : [Malpani, A. ve Turner, S. 2000]

ValPolResponse

{

vpResponseVersion(Önceki bölümde incelenmiş ValPolRequestVersion ile aynı tanım ve özelliklere sahip tamsayı tipinde sürüm bilgisi nesnesi)

maxCVResponseVersion (Sunumcu tarafından doğrulama taleplerinde desteklenebilecek en fazla sürüm numarası bilgisini sağlayan tamsayı tipinde nesne)

maxVPResponseVersion (Sunumcu tarafından doğrulama politikası taleplerinde desteklenebilecek en fazla sürüm numarası bilgisini sağlayan tamsayı tipinde nesne)

serverConfigurationID (Sunumcu tarafından yanıtın oluşturulması zamanında sunumcunun sahip olduğu konfigurasyonu belirten tamsayı tipinde nesne)

thisUpdate (Sunumcu tarafından bu politika yanıtının imzalanma tarih ve zamanını belirten GeneralizedTime tipinde zaman nesnesi)

nextUpdate (Sunumcunun politika yanıtlarını güncelleyeceği zaman bilgilerini içeren GeneralizedTime tipinde zaman nesnesi)

validationPolicies (Sunumcu tarafından desteklenen politikalara ait referans bilgilerini içeren nesne belirteçleri dizisi)

validationAlgs (Sunumcu tarafından desteklenen doğrulama algoritmalarına ait referans bilgilerini içeren nesne belirteçleri dizisi)

authPolicies (Sunumcuya ait geçerlenme işleminde kullanılacak politika bilgilerini içeren nesne)

responseTypes (Sunumcunun desteklediği yanıt türlerine ait referans bilgilerini içeren nesne)

defaultPolicyValues (Sunumcunun uyguladığı varsayılan doğrulama politikası bilgisini belirten nesne)

revocationInfoTypes (Sunumcunun işleyebileceği iptal bilgisi kaynaklarına ait referansları içeren nesne)

signatureGeneration (CVResponse türünde mesajların imzalanmasında sunumcu tarafından uygulanabilecek sayısal imza algoritmaları bilgisini içeren nesne)

signatureVerification (Sunumcu tarafından doğrulanabilecek sayısal imza algoritma bilgilerini içeren nesne)

hashAlgorithms (Sunumcunun sertifika ya da taleplerin özetini çıkarmada kullanabileceği algoritmaları belirten nesne)

serverPublicKeys (Sunumcu tarafından uygulanabilecek KeyAgreePublicKey tipinde açık anahtar bilgilerini içeren nesne)

clockSkew (clockSkew için öngörülen zamanı içeren tamsayı türünde nesne olup varsayılan değeri 10 dakikadır)

requestNonce (Opsiyonel olarak talebi belirten referans string tipinde belirteç nesnesi)

}

ResponseTypes

{

cached-only (Sunumcu sadece bellekteki yanıtları iletebilir)

non-cached-only (Sunumcu sadece bellekte olmayan yanıtları iletebilir)

cached-and-non-cached (Sunumcu tüm yanıtları iletebilir)

}

3.3 Protokollerin Entegrasyon Modelleri

Bu bölümde, önceki bölümlerde detaylı olarak yapısal ve uygulama tabanlı analizleri gerçekleştirilen CMC ve SCVP protokollerinin UMU-PKI içerisinde sertifika yönetim ve sertifika doğrulama mekanizmaları olarak entegrasyonuna yönelik süreç tabanlı modeller ve uygulamaya yönelik tasarımlar sunulmuştur. Öncelikle, sertifika yaşam döngüsü işlevlerinin, sisteme entegrasyonunu sağlayacak CMC protokolünün, içerdiği mekanizmalardan sistemde gereksinim duyulan sertifika işlevleri yönetim fonksiyonlarının modellenmesi ve sisteme dahil edilmesi yönünde çalışmalar aktarılmış, sonraki bölümde, sertifika doğrulama işlevlerinin SCVP protokolünün tanımları doğrultusunda entegrasyon modellemesi gerçekleştirilmiştir.

Protokollere ait mekanizmaların, sistem içerisinde modellenmesine yönelik çalışmaların aktarılma sürecinde, öncelikle ilgili mekanizmanın tanımı ve sistemdeki yeri ile ilgili bilgiler sunularak, bu mekanizmanın sistemdeki operasyonel süreç modellemesi gerçekleştirilmiştir. Süreç modellemesinin tasarlanmasının ardından, bu süreç modeli baz alınarak, protokolün tanım bilgileri doğrultusunda sisteme uygulama tabanlı entegrasyon tasarımı, örnek bir modelleme altında sunulmuştur. CMC ve SCVP protokollerinin sisteme uyarlanmasında her bir mekanizma, belirtilen bu süreç içerisinde çalışmada yer almıştır.

3.3.1 Sertifika Yönetiminde CMC Entegrasyon Modelleri

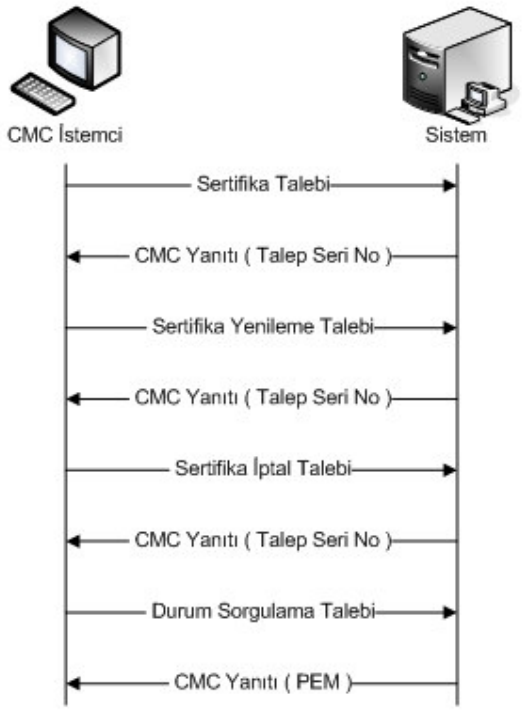
CMC protokolünün entegrasyon modelleri, sertifikaların kayıt, iptal, yenileme, edinme, iptal listesi edinme ve askıdaki sertifika talebinin durumunun sorgulanmasını sağlayan işlemlerin sistem içerisinde sağlanabilmesine yönelik aşağıda sıralanan 6 adet servisi sağlamalıdır :

- Sertifika Talepleri : CMC, aynı mesaj içerisinde farklı türden birden fazla sertifika talebi içerebilmesi ve bu sayede sistemdeki döngü sayısını da azaltarak hem yüksek fonksiyonalite hem de sistemde esneklik sağlar.
- Sertifika Yenileme Talepleri : Bir kullanıcı, kendi kişisel sertifikasının yenilenmesi için sistemde bir talep oluşturabilir.
- Sertifika İptal Talepleri : Bir kullanıcı, gerçekleşmesi beklenen sertifika iptal işleminin yerine getirilmesi için bir neden belirtmek suretiyle, PKI sisteminden kendi kişisel sertifikasının iptalini talep edebilir.
- Sertifikaların Sistemden Edinilmeleri : Sistemdeki depodan sertifika alınabilmesine yönelik işlemi içerir.

- Sertifika Özellik/İptal Listesi Talepleri : Sistemde yer alan sertifika ya da özellik iptal listelerini içeren depodan iptal listelerinin edinilmesini kapsar.
- Askıdaki Talep Durum Sorgulama Talepleri : Bir kullanıcının, sistemde askıda bulunan sertifika talep isteğininin durumunu sorgulamasıdır.

Kullanıcı, içinde bulunduğu sistemden, bir sertifika geçerleme / iptal / yenileme talebinde bulunur. Bu talep, Sertifika Makamı tarafından geçerlenmesi için, sistem içerisinde saklanır ve her bir talep için sistemde bir seri numarası üretilir. Bu seri numaraları, askıdaki talebin sorgulanmasına yönelik mesajlar içerisinde kullanılarak, talebe ait durumun, sistem birimlerince takip edilebilmesini sağlarlar. Benzer şekilde sertifikaların ya da iptal listelerinin edinilmesine yönelik servislerde de işleyiş aynıdır. İstemci, bir sertifika / iptal listesi talebinde bulunur. Sistem, istemci sertifikası ya da sertifika / özellik iptal listesini PEM formatında istemciye iletir. Herhangi bir güvenli iletişim kanalının kurulmasının öncesinde, bir istemci, sistemdeki geçerleme servisleri aracılığıyla, kullanılacak sertifikaları geçerlemelidir. Geçerleme servisi, sistemde bulunan birimlerin açık anahtarlarının sistemde kullanılması suretiyle, potansiyel tehlikelerden korunduğunu garantileyen, temel yapılardan biridir. Geçerleme servisi, altında yer alan birimlerce kullanılması planlanan sertifikaların geçerlenmiş olup olmadıkları, güvenilir bir makam tarafından yayınlanıp yayınlanmadıkları, kullanımda olup olmadıkları ve planlanan iş için kullanılmasının doğru olup olmadığı gibi kritik konuların sistem birimlerince garantilenmesini sağlayan önemli bir araçtır. Çoklu etki alanına sahip senaryolar içerisinde, çapraz sertifikaların kullanımının kritik olduğu durumlarda bu husus daha fazla öneme sahiptir. Bu tip çoklu etki alanları içeren ortamlarda, farklı etki alanları arasındaki güven seviyesi kesin olarak belirlenmiş olmalıdır. Dahası, bunun sağlanması, sistemdeki son kullanıcıların ve cihazların sertifika ve veri geçerleme proseslerini en aza indirgeyebilmelerine olanak sağlar.

CMC istemci uygulaması, sertifika kayıt, iptal, yenileme, sertifika edinme, sertifika ya da özellik iptal listelerini edinme ve askıdaki sertifika taleplerinin durumlarını sorgulama işlemlerini içeren servisler sunmalıdır.

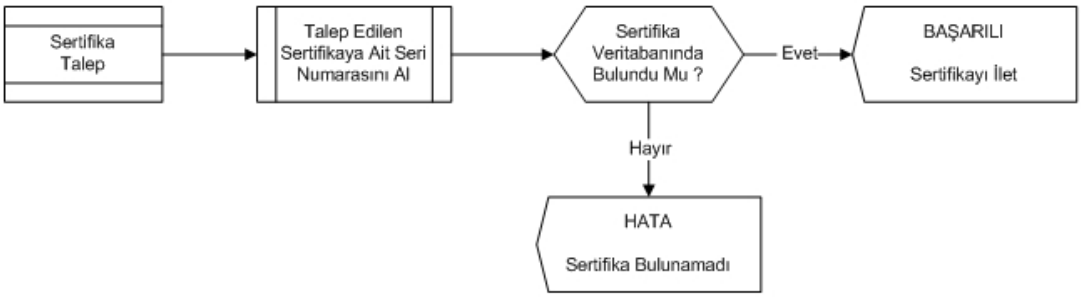


Şekil 3.1 CMC istemci sunumcu ilişkisi

3.3.1.1 Sertifika Talepleri

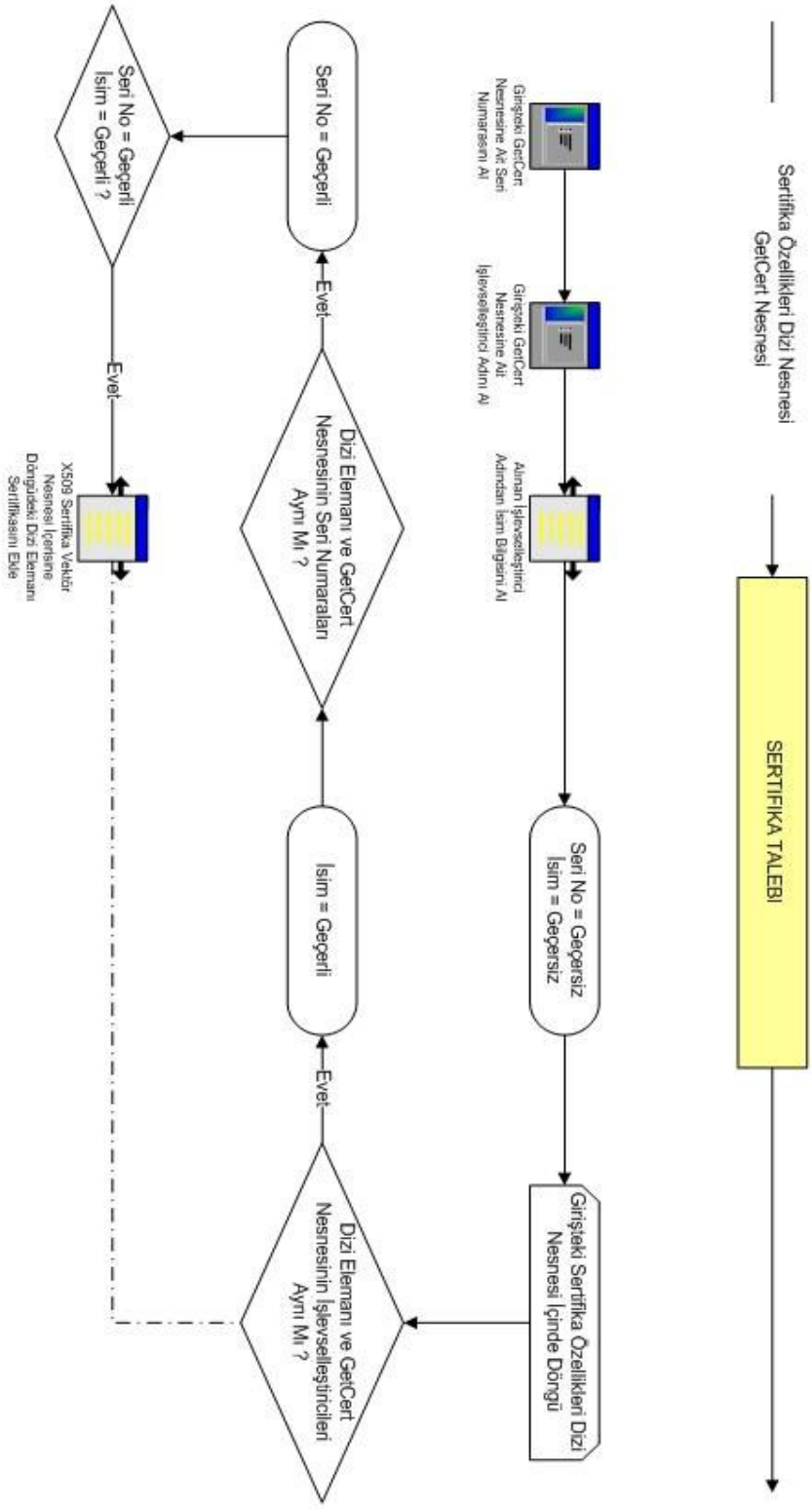
Sertifika taleplerinin sisteme CMC protokolü gereğince uyarlanması için öncelikli olarak sistem süreç modelinin tasarlanması ve bu sürece ait metodolojinin entegrasyonu gereklidir. Sertifika talep sürecini basit bir süreç modeli içerisinde, CMC protokolü tanımlamalarına uygun olarak geçerlenmesi için, ilk aşamada, taleplerin istemcide oluşturulduktan sonra, sunumcuya iletimi gerekmektedir. Sunumcu, gelen talepteki bilgiler içerisinden, talepte yer alan sertifikaya ait seri numarasını alarak, talep sunumcuda bulunan veritabanında, bu seri numarasıyla ilişkili sertifika bilgilerini sorgular. Bu sorgulama sonucu eğer veritabanında geçerli bir sertifika verisine ulaşabilirse, bu sertifika bilgilerini istemciye iletir. Eğer talep seri numarasına ait veritabanında herhangi bir kayıda erişemezse, bu durumda istemci, sertifikanın bulunamadığına dair sistem tarafından iletilecek bir hata mesajı alır. Temel sertifika talep süreci en basit haliyle sistemde bu süreç modeli mantığında entegre edilebilir.

Sertifika taleplerine ait süreci içeren entegrasyon modeli Şekil 3.2'de gösterilmektedir.



Şekil 3.2 CMC sertifika talep süreci entegrasyon modeli

CMC protokolünün UMU-PKI v6 sistemi içerisine operasyonel olarak uyarlanabilmesine yönelik tasarlanan bu sürecin, uygulama tabanlı olarak sisteme entegrasyonunda, temel bir sertifika talebi metodu içerisinden, sürecin tümüyle öngörülmesi gerçekleştirilmiştir. Buna yönelik olarak metodun girdileri, bir sertifika özellik dizisi nesnesi ve bir sertifika edinme (yayınlayıcı adı ve sertifika seri numarasına sahip) nesnesi olarak öngörülmüştür. Belirtilen operasyonel süreçteki tasarıma uygun olarak, ilk aşamada metod, getSerialNumber() metodunu kullanarak, talep içerisindeki sertifika edinme nesnesinden, talebe ait seri numarasını alır. Benzer şekilde aynı nesnenin içerdiği yayınlayıcı adı, getIssuerName() metodu aracılığıyla, nesneden edinilir. Bu bilgilerin alınmasının ardından, sertifika talep metodunun diğer bir girdisi olan, sertifika özellik dizisinin sahip olduğu sertifikalar, seri numarası ve yayınlayıcı adı ile kontrol edilir. Eğer bu iki verinin eşleşebileceği bir sertifika, bu dizi içerisinde saptanabilirse, bu durumda eşleşen sertifika, taleplerin toplandığı sertifika vektörüne eklenerek sistemde saklanır ve CMC yönetim mekanizmalarınca değerlendirilir. Uygulama tabanlı tasarlanan bu entegrasyon sürecinin detayları, Şekil 3.3'deki tasarım modelinde belirtilmiştir. (Bkz. Şekil 3.3)



Şekil 3.3 CMC sertifika talep süreci uygulama entegrasyon modeli

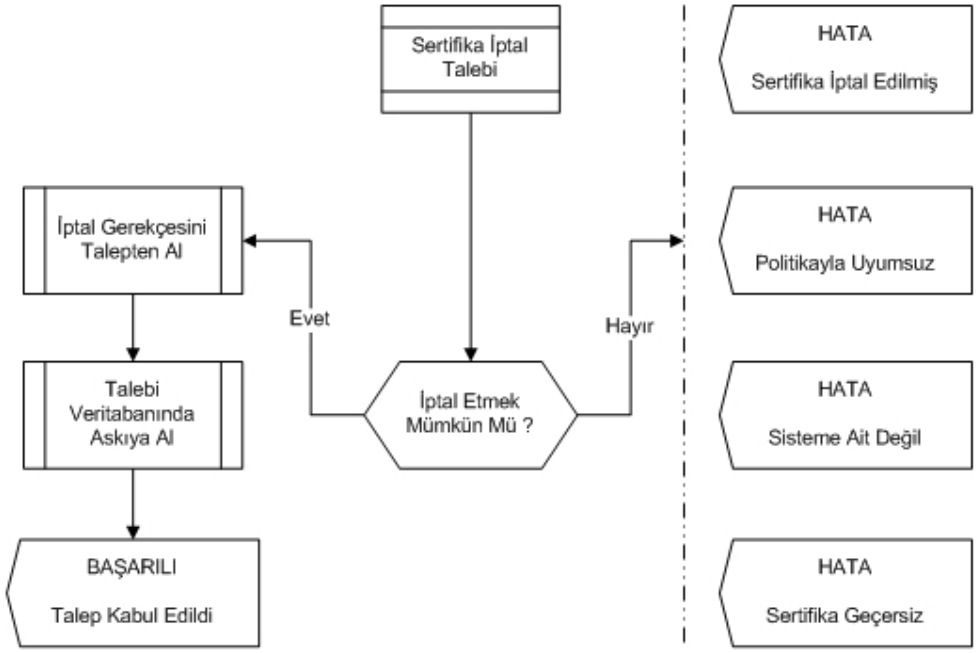
3.3.1.2 Sertifika İptal Talepleri

Sertifika iptal taleplerinin yönetilmesine yönelik, CMC protokolü entegrasyonu tasarımı, istemciler tarafından gerçekleştirilecek sertifika iptal taleplerinin, sistem içerisinde ne şekilde değerlendirileceğini tanımlayacak bir süreci öngörmektedir. Bu süreç içerisinde istemciler, sertifika iptal talebiyle ilgili verileri, tümüyle talepte eksiksiz olarak oluşturmuş ve sonrasında bu taleplerle ilgili sistemdeki kontroller ve sistemin istemciye vereceği yanıt, koşullu olarak protokole uygun şekilde sistemde tasarlanmıştır.

Sistemdeki bir istemci tarafından yaratılırak, sunumcuya iletilen bir sertifika iptal talebi, bu talepleri değerlendiren sunumcu tarafından öncelikli olarak politika uyum kontrolüne tabi tutulur. Bu kontrol içerisinde, talepte yer alan sertifika veya sertifikalar, sistemin ait olduğu organizasyon tarafından belirlenerek tüm sistem birimlerince uygulanan politika içerisindeki sertifika iptal politikası gereğince, iptal edilip edilemeyeceğinin tetkikine tabi tutulurlar. Bu kontrol, sistemde uygulanan politikada yer alacak sınırlamalar ve iptal geçerleme opsiyonlarının, talepte yer alan sertifika veya sertifikalar için geçerli olup olamayacağı bilgisini sağlar. Kontrol sonucunda, eğer politika gereğince sertifikanın iptal edilmesi mümkün değilse, talepteki sertifikanın, sistem tarafından iptaline izin verilmediği gerekçesi, istemciye bildirilir. Bu bildirim, iptal edilememe gerekçesini ve talep seri numarasını içerecek bir mesaj formatında gerçekleştirilir. İptal edilememe gerekçeleri, sertifikanın sistemde kayıtlı sertifikalar içerisinde yer almaması, sertifika verisinin bütünlük arz etmeyerek geçersiz olması ya da sertifikanın hali hazırda iptal edilmiş olması gibi çeşitli türde bilgiler içerir.

Eğer uygulanan organizasyonel sistem politikası gereğince, talepte yer alan sertifika, iptal edilmeye uygunsa, bu durumda öncelikli olarak sertifikanın iptal edilme gerekçesi, istemci tarafından iletilen talep içerisinden alınır. Talep, sunumcu içerisindeki veritabanında, taleple ilgili bilgiler kaydedilerek askıya alınır ve istemciye talebinin kabul edildiğine dair bilgi ve talebe ait seri numarası, mesaj formatında iletilir.

Sertifika iptal taleplerinin yönetimine ilişkin sürece ait operasyonel süreç entegrasyon modeli Şekil 3.4'de gösterilmiştir.

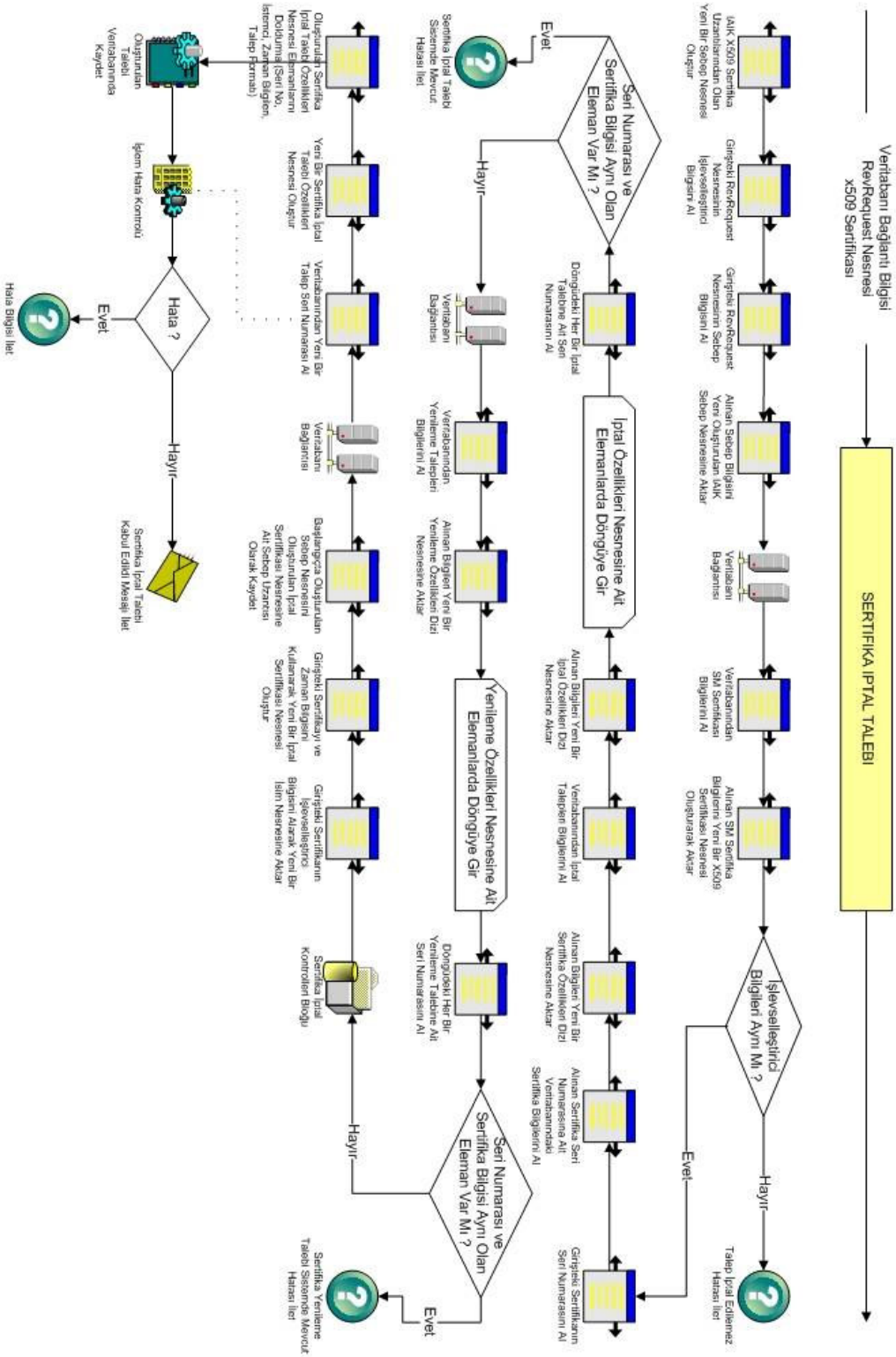


Şekil 3.4 CMC sertifika iptal talebi süreci entegrasyon modeli

Sertifika iptal taleplerinin uygulama tabanlı entegrasyon tasarımında, önceki bölümde ifade edilen operasyonel süreç öngörülerek gerçekleştirilen bir modelle, sistem içerisinde, CMC protokolü mekanizmalarına uygun olarak, iptal taleplerinin yönetimi sağlanmıştır. Uygulama entegrasyon tasarımındaki iptal talepleri sürecini içerecek metodun girdileri arasında, veritabanı bağlantı nesnesi, Java'da tanımlı RevRequest sertifika iptal nesnesi ve iptal edilmesi istenen ilgili X.509 sertifikası yer alır. Önceki tasarımlarda aktarılan şekilde, girdiler arasında yer alan iptal nesnesindeki yayınlayıcı ve iptal gerekçesine ait bilgiler, talep içerisinden edinilerek IAIK.X.509.Extensions sınıfı altında yer alan ReasonCode sebep nesnesine aktarılır. Veritabanında gerçekleştirilecek bir sorgulamayla, sistemdeki Sertifika Makamı'na ait sertifika bilgisi alınır. Talepten elde edilen yayınlayıcı bilgisi, Sertifika Makamı'na ait sertifika ile birlikte, Java.Security.Cert.Certificate sınıfına ait metodlarla karşılaştırılarak, yayınlayıcı bilgilerinin eşleşip eşleşmeyeceği kontrol edilir. Eğer yayınlayıcı bilgilerinde bir eşleşme oluşmazsa, sistem, istemciye sertifikanın iptal edilemeyeceği hatasını iletir. Eğer yayınlayıcı bilgileri eşleşebiliyorsa, bu durumda metodun girdileri arasında yer alan X.509 sertifikasına ait seri numarası alınarak, veritabanındaki bu sertifikaya ait bilgiler,

bir sertifika özellikleri dizisi nesnesine aktarılır. Benzer şekilde veritabanından edinilecek iptal taleplerine ait veriler de, yeni bir iptal özellikleri dizisi nesnesine aktarılır. Bu iki dizi elemanları arasında gerçekleştirilecek bir kontrol döngüsü vasıtasıyla, sertifika iptal talebinin, sistemdeki mevcut sertifika iptal talepleri arasında yer alıp yer almadığı belirlenir. Eğer talep sistemde mevcutsa, istemci, talebinin sistemde yer aldığını belirtecek bir hata mesajıyla uyarılır. İptal talebi sistemde mevcut değilse, aynı işlem, sistemdeki sertifika yenileme talepleri için gerçekleştirilir. Benzer şekilde sertifika özellikleri dizisi nesnesi ile veritabanından alınacak sertifika yenileme özellikleri dizisi elemanları arasında yapılacak kontrol döngüsü ile talepte yer alan sertifikayla alakalı olarak, sistemde herhangi bir yenileme talebinin mevcut olup olmadığı kontrolü gerçekleştirilir. Eğer bu iki dizi elemanları arasında eşleşen bir durum oluşursa, sistem, istemciye, iptalini talep etmiş olduğu sertifikayla ilgili olarak, sistemde bir yenileme talebinin mevcut olduğunu, bir hata mesajı halinde ekranında bildirir.

Kontroller sonucu iptali talep edilen sertifikayla ilişkili olarak sistemde herhangi bir uygunsuzluk mevcut değilse, bu durumda sertifika yayınlayıcı adı, sertifika bilgileri, sistemin güncel zaman bilgileri ve JAVA'da tanımlı IAIK.X.509.RevokedCertificate sınıfına ait metodlar kullanılarak, yeni bir iptal sertifikası nesnesi oluşturulur. Başlangıçta oluşturulan sebep nesnesi verileri, bu iptal sertifikasına ait sebep uzantısına aktarılır. Veritabanından yeni bir talep seri numarası alınarak, yeni bir iptal talebi nesnesi oluşturulur. Bu iptal talebi nesnesi, tanımlar gereğince edinilen verilerle doldurularak, veritabanında kaydedilir. Eğer bu aşamalarda bir hata oluşmamışsa, sistem, istemciye, iptal talebinin kabul edildiğine ve veritabanında askıya alındığına dair bir işlem mesajı iletir. Bu uygulama entegrasyon tasarım modeline yönelik detaylar ve süreçte yer alan diğer işlemler Şekil 3.5'de gösterilen örnek tasarım modelinde sunulmuştur. (Bkz. Şekil 3.5)



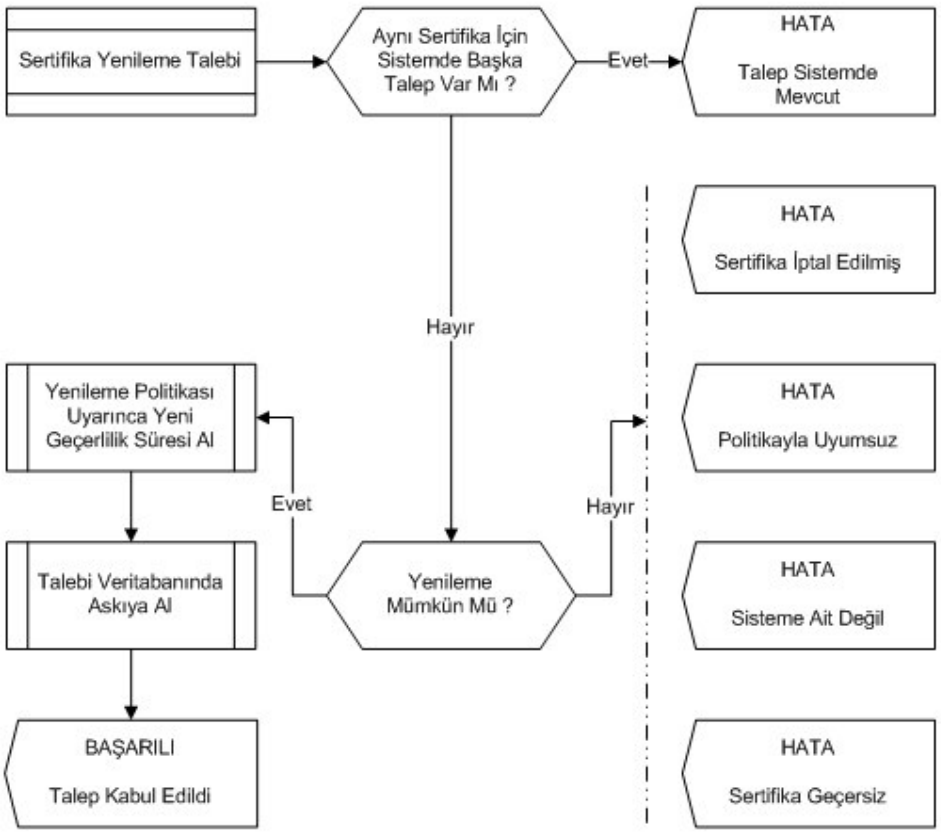
Şekil 3.5 CMC sertifika iptal talebi süreci uygulama entegrasyon modeli

3.3.1.3 Sertifika Yenileme Talepleri

Bu servis, kişisel bir sertifikanın yenilenmesi işleminde kullanılmaktadır. Bunun için uygulama, kullanıcıdan, hangi sertifikanın iptal edileceği (sistemde geçerli bir sertifika olması gerekir) bilgisini, sertifikaya ait gizli anahtarı ve gizli anahtarı korumakta olan giriş ifadesi bilgilerini ister. Burada unutulmamalıdır ki, kullanıcı, yeni sertifika için aynı gizli anahtarı kullanacaktır. Yeni sertifikaya ait son geçerlilik tarihi, açık anahtar altyapısına ait sistem içerisinde uygulanmakta olan politika gereğince kaydedilir.

Sertifika yenileme taleplerinin sisteme entegrasyonu sürecinde talep, istemci tarafından gönderildikten sonra, sistem içerisinde talepte belirtilen sertifikaya ait başka bir yenileme talebinin bulunup bulunmadığının kontrolü gerçekleştirilir. Bu kontrol sonucunda, eğer sistemde aynı sertifikayla ilgili bir başka yenileme talebi bulunuyorsa, sistem, istemciye bu durumla ilgili hata mesajı iletir. Eğer sistemde, bu sertifikayla ilgili bir başka yenileme talebi bulunmuyorsa, bu durumda sistemin ait olduğu organizasyona ait sertifika yenileme politikası gereğince, ilgili sertifikanın yenilenmesinin mümkün olup olmadığı kontrol edilir. Bu kontrol sonucu olumsuz olarak değerlendirilmekteyse sistem, istemciye sertifikanın politikanın hangi maddesince yenilenemez olduğunu bildirir. Bu bildirim işlemi, bir hata mesaj ekranı uyarınca gerçekleşir. Politika gereğince bir sertifikanın yenilenememesi gerekçeleri arasında, sertifikanın daha önce geçersiz hale getirilmiş olması, sertifikanın sisteme ait olmaması ya da ilgili organizasyonel politikanın bir maddesi olarak belirtilebilir. Eğer sistem politikası gereğince, sertifikanın yenilenmesinde herhangi bir olumsuzluk bulunmuyorsa, bu durumda ilk olarak politikada tanımlı olan yenileme prosedürü uyarınca, ilgili sertifika için yeni bir geçerlilik süresi hesaplanır. Hesaplanan bu geçerlilik süresi, sertifikayla ilişkilendirilerek, talep sunumcu veritabanında bu yenileme talebi askıya alınır. Son olarak istemciye, talebinin değerlendirilip uygun bulunduğuna dair bir mesajla bildirimde bulunularak, süreç sonlandırılır.

Bu operasyonel yenileme taleplerine yönelik sürece ait şematik gösterim Şekil 3.6'da sunulmuştur. (Bkz. Şekil 3.6)

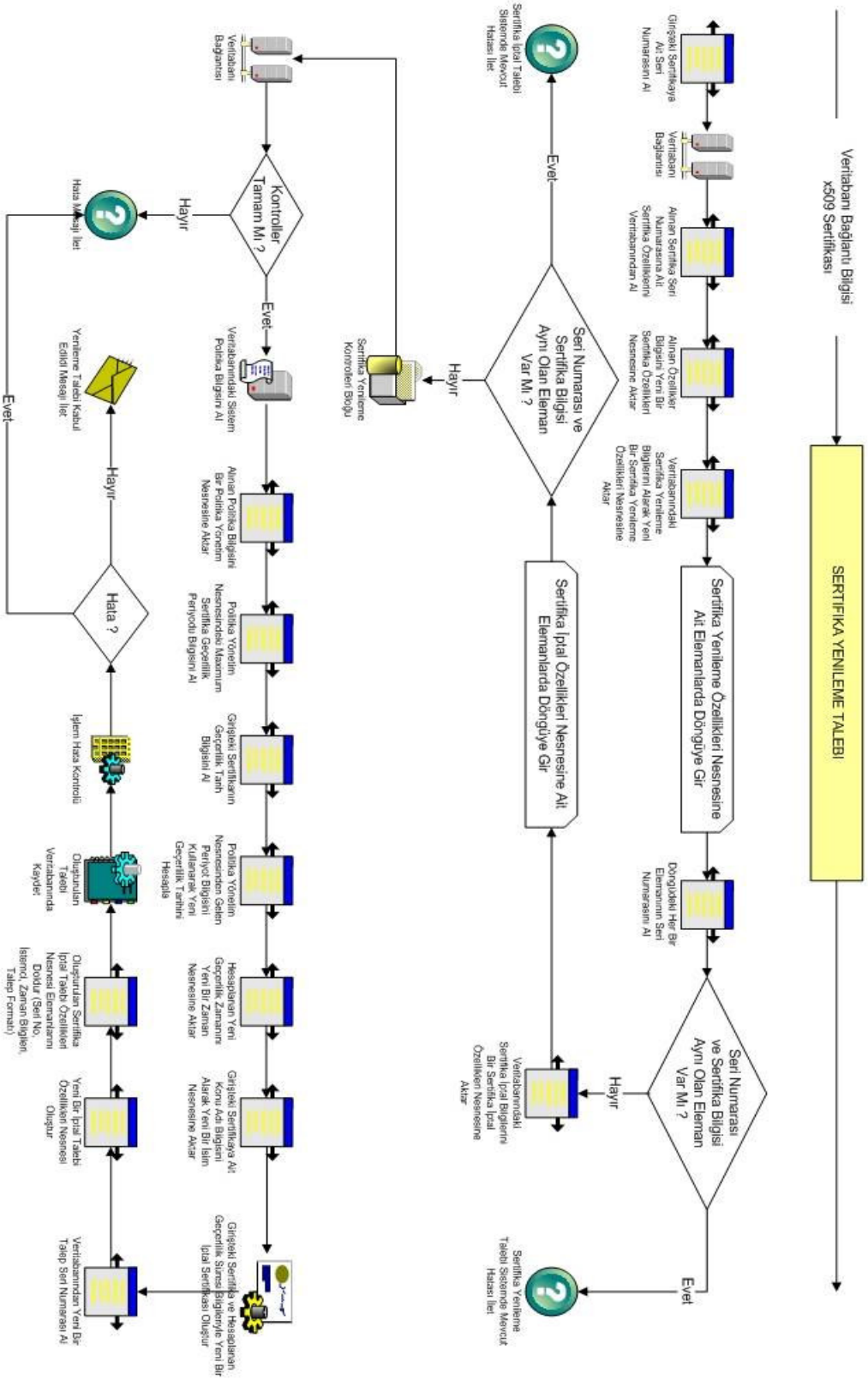


Şekil 3.6 CMC sertifika yenileme talebi süreci entegrasyon modeli

Sertifika yenileme taleplerinin CMC protokolü gereklerince sisteme entegrasyonunda tasarlanan operasyonel süreç tabanlı uygulama entegrasyon modeli, önceki bölümde yer alan sertifika talep entegrasyon modeline kıyasla, süreç içerisinde yer alan kontrol mekanizmaları sayısının fazla olmasından dolayı daha kapsamlı bir tasarımı içerir. Tasarlanacak yenileme taleplerini yönetici metodun girdileri arasında, veritabanı bağlantı nesnesi ve yenilenmesi için iletilen X.509 sertifikası yer alır. Talepten alınan X.509 sertifikasına ait seri numarası, IAIK.X.509 sınıfı altında yer alan metodlar aracılığıyla edinildikten sonra, girdiler arasındaki veritabanı bağlantı nesnesi kullanılarak talep sunumcudaki veritabanı ile bağlantı kurulur.

Veritabanından, ilgili sertifikaya ait veriler alınarak Java Crypto sınıfına ait nesnelerle yeni bir sertifika özellik (CertificateProperties) nesnesine aktarılır. Veritabanında saklanan yenileme taleplerine ait veriler, yeni bir sertifika yenileme özellikleri dizisi nesnesine aktarılarak, oluşturulan bu iki dizi nesnesi elemanları içerisinde bir kontrol döngüsü yardımıyla eşleşen öğelerin olup olmadığı saptanır. Eğer bu kontrol sonucunda, seri numaraları eşleşen sertifikalara rastlanırsa, bu durumda veritabanında, belirtilen sertifikaya ait başka bir yenileme talebinin yer aldığı anlaşılarak, istemciye durumla ilgili hata mesajı iletilir. Aynı kontroller ikinci aşamada, veritabanından alınacak sertifika iptal özellikleri dizisi nesnesiyle, ilk aşamada oluşturulan sertifika özellikleri dizisi nesnesi arasında gerçekleştirilir. Bu kontrol sonucunda da, yenileme talebinde yer alan sertifikanın sistemde iptal edilip edilmediği bilgisine ulaşılır.

Bu kontrollerin gerçekleştirilmesi sonucunda uygulama, yenileme talebinde yer alan sertifikanın sistemde uygun bir şekilde yenilenip yenilenemeyeceği kararına ulaşır. Eğer herhangi bir uygunsuzluk sonucu süreç sonlandırılmamışsa, bu durumda süreç, organizasyon politikasının sistemden alınmasıyla devam eder. Java'da tanımlı PolicyManager sınıfına ait nesnelerin kullanımıyla, politikada tanımlı yeni geçerlilik süresi hesaplanır. Geçerlilik süresinin hesaplanmasıyla, sistemde yeni bir iptal sertifikası oluşturulur. Java.Security.Cert sınıfı altında yer alan RevokedCertificate sınıfı aracılığıyla oluşturulan iptal sertifikası için veritabanından yeni bir seri numarası edinilir. İptal sertifikasına ait sertifika özellikleri nesnesi, edinilen seri numarası, istemci adı, talep bilgisi gibi verilerle doldurularak sistemde kaydedilir. Eğer bu aşamalarda bir hata oluşmamışsa, sistem, istemciye talebinin kabul edildiğine ve veritabanında askıya alındığına dair bir işlem mesajı iletir. Bu uygulama entegrasyon tasarım modeline yönelik detaylar ve süreçte yer alan diğer işlemler Şekil 3.7'de gösterilen örnek tasarım modelinde sunulmuştur. (Bkz. Şekil 3.7)

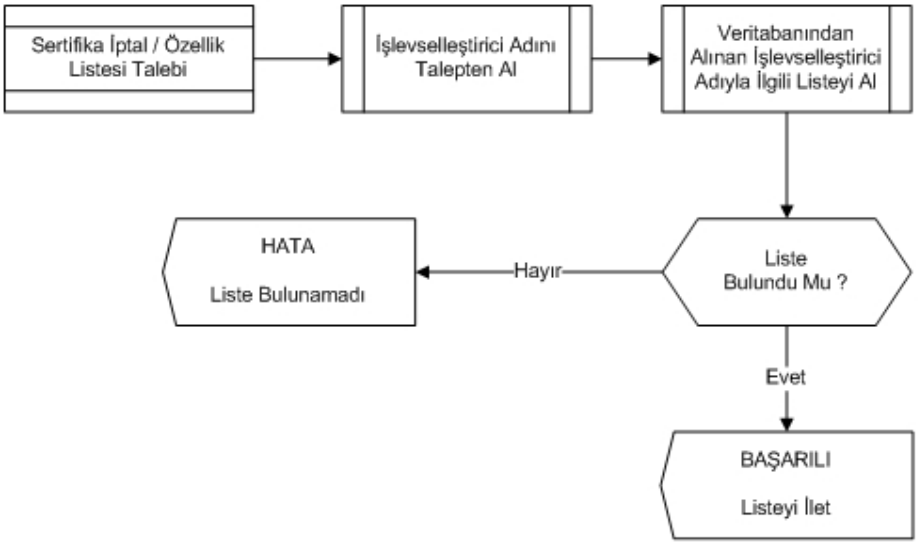


Şekil 3.7 CMC sertifika yenileme talebi süreci uygulama entegrasyon modeli

3.3.1.4 Sertifika İptal Listesi Talepleri

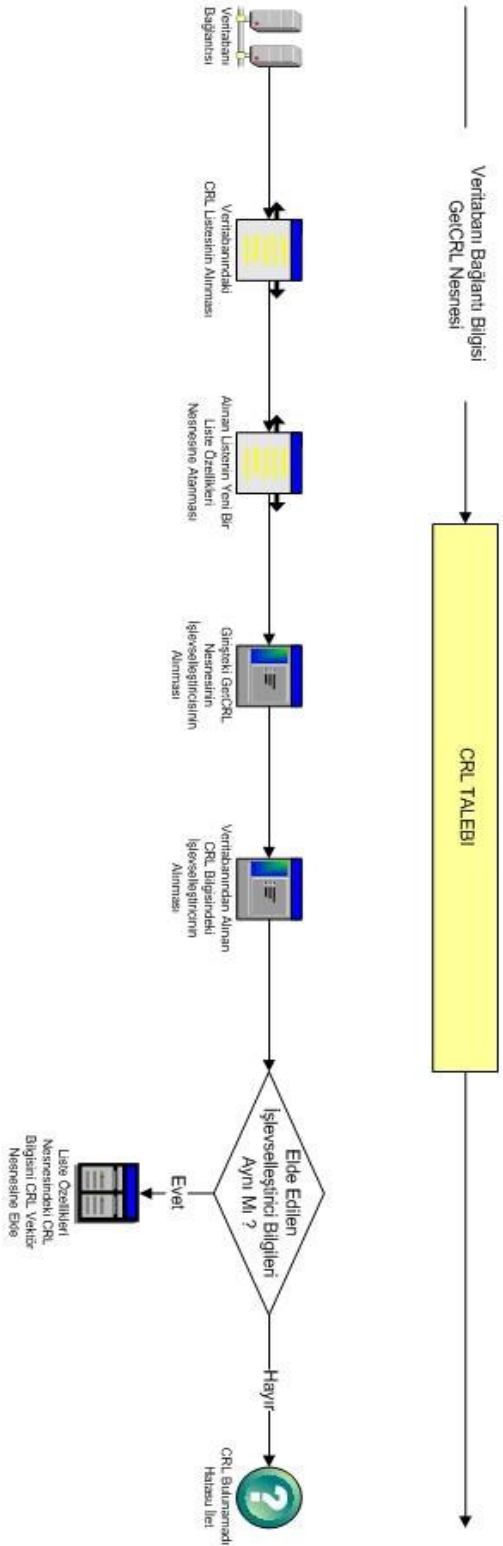
Bu servis, iptal listelerinin bulunduğu ortak depodan sertifika / özellik iptal listelerinin alınması mekanizmalarını yönetmede görev alır. Sertifika iptal listelerinin sistem içerisindeki istemciler tarafından talep edilmelerini öngören süreçte ilk olarak, istemci tarafında sertifika iptal listesi talebi oluşturulur. Sistem, istemciden gönderilen talepteki yayınlayıcı birim adını, DN formatında önceki bölümlerdeki modellerde belirtildiği şekilde talepten alır. Sonrasında, alınan bu yayınlayıcı adıyla ilgili olarak veritabanında herhangi bir liste olup olmadığının kontrolü gerçekleştirilir. Yapılan veritabanı sorgusu sonucunda eğer talepteki yayınlayıcı adıyla ilişkili herhangi bir listeye ulaşılamazsa, bu durumda sistem, istemciye listenin bulunamadığına dair bir hata mesajı iletir. Eğer veritabanında, ilgili yayınlayıcı adına sahip bir liste bulunabilirse, bu liste sistem tarafından istemciye iletilir. Sertifika iptal listelerinin sistem içerisindeki istemci birimler tarafından talep edilmelerini sağlayacak bu süreç, benzer şekilde özellik listelerinin istemcilerce talep edilebilmelerinde de uygulanır.

Seritifika iptal ya da özellik listelerinin talep edilmelerini öngören bu sürece ait model Şekil 3.8'de sunulmuştur. (Bkz. Şekil 3.8)



Şekil 3.8 CMC sertifika iptal listesi talep süreci entegrasyon modeli

Sertifika iptal listelerinin sistem içerisindeki istemciler tarafından talep edilmelerine yönelik CMC protokolü temelli uygulama entegrasyon tasarım modelinde, önceki bölümde tasarlanan operasyonel süreç bazında oluşturulan metodun girdileri arasında, veritabanı bağlantı nesneleri ve ilgili iptal listesine ait verileri içerecek bir liste nesnesi yer alır. Tasarımda ilk olarak veritabanı bağlantısı gerçekleştirilerek sonrasında veritabanından, ilgili listeye ait seri numarasını içeren veriler, yeni oluşturulacak bir liste özellikleri dizisi nesnesinde toplanır. Metodun girdileri içerisinde yer alan liste nesnesindeki yayınlayıcı adı, veritabanından edinilen verilerle oluşturulmuş liste dizisi nesnesine ait elemanlarla gerçekleştirilecek bir kontrol döngüsü içerisinde karşılaştırılır. Eğer bu döngü içerisinde yayınlayıcı adları eşleşen bir liste bulunabilirse, bu durumda liste, istemciye iletilerek süreç sonlandırılır. Aksi taktirde sistem, istemciye, talep etmiş olduğu listenin bulunamadığını, bir hata mesajı olarak bildirir. Bu uygulama entegrasyon sürecine ait detaylar ve ilgili diğer bilgiler Şekil 3.9'da gösterilen örnek tasarım modelinde sunulmuştur. (Bkz. Şekil 3.9)



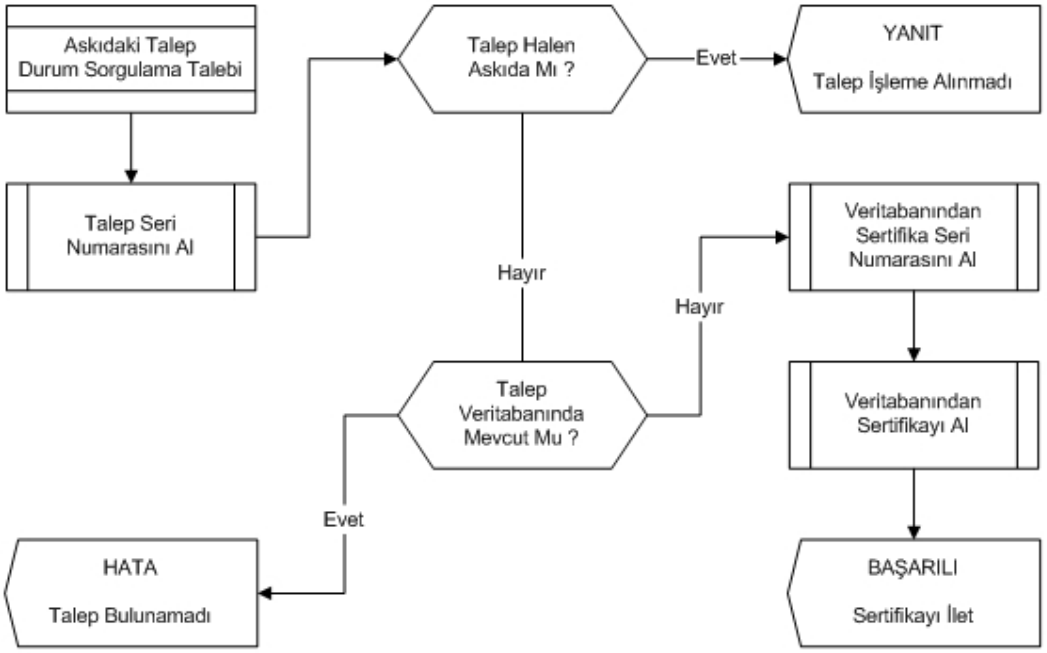
Şekil 3.9 CMC sertifika iptal listesi talep süreci uygulama entegrasyon modeli

3.3.1.5 Askıdaki Talep Durum Sorgulama Talepleri

Bu servis, kullanıcıların PKI sistemini sorgulayarak kendilerine ait askıdaki bir sertifika talebinin durumunu öğrenebilmelerine imkan sağlar. Sertifika Talep servisinde belirttiğimiz gibi, kullanıcının askıdaki talebinin durumunu sorgulayabilmesinde, talebe karşılık oluşturulan seri numarası bilgisi kullanılır. Bu nedenle talep henüz açık anahtar altyapısı yöneticisi tarafından işleme alınmadan, kullanıcı bu servisi aracılığıyla talebin işleme alınmadığı mesajını alabilir. Talep, geçerlendiğinde ve sertifika yayınlandığında, kullanıcı, CMC yanıtı içerisinde yeni sertifikasını alabilecektir.

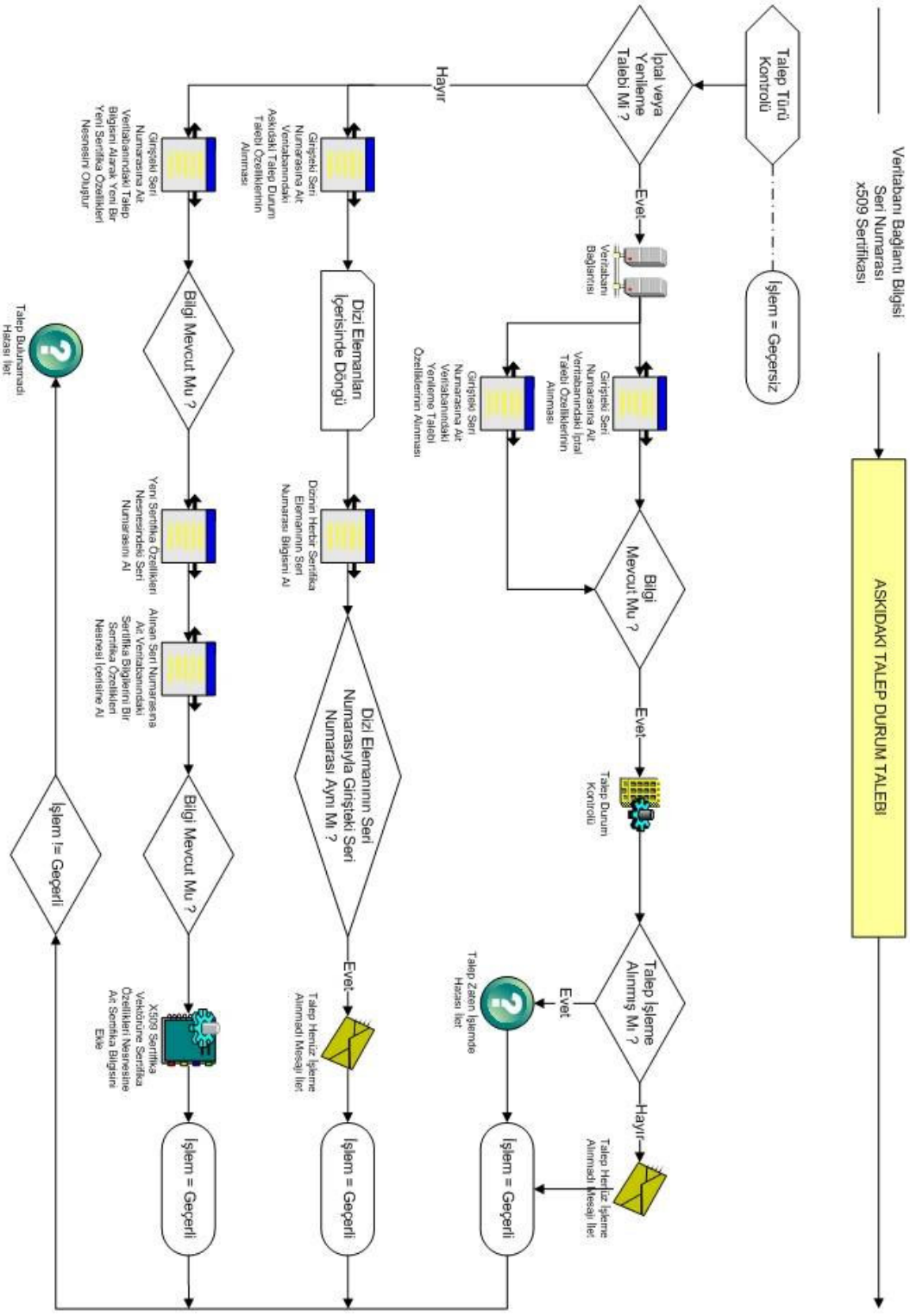
Sistemdeki istemci birimler tarafından gönderilen sertifika talep, yenileme, iptal ve benzeri tüm talepler, önceki bölümlerde üzerinde durulan modellerdeki yapılar içerisinde, sistemde talep edildikten sonra değerlendirilerek, eğer talepte herhangi bir uygunsuzluğa rastlanmamışsa, talep sunumcuya ait veritabanında askıya alınarak kaydedilirler. Veritabanında askıya alınan bu taleplerin durum bilgileri, istemciler tarafından belirli aralıklarla edinilerek, gönderilen taleplerin hangi durumda oldukları kontrol edilir. Bu tip durum kontrolleri ve sorgulamalarına yönelik olarak istemciler, sistem içerisinde talep oluştururlar. Bu talepler temelde, durumlarını sorgulamak istedikleri taleplere ait seri numarasını ve durum sorgulama talebine ait talep türü bilgilerini içerir. İstemcilerce oluşturularak gönderilen bu talepler, sistemde öncelikle talepte yer alan durum sorgulamasının gerçekleştirileceği talebin, halen veritabanında askıda bulunup bulunmadığının kontrolüne tabi tutulurlar. Eğer bu kontrol sorgulaması sonucunda, ilgili talebin işleme alınmadığı bilgisi sistemden edinilirse, bu durumda istemciye, talebin henüz işleme alınmadığı yönünde bir yanıt mesajı iletilir. Aksi halde, yani talebin işleme alındığı bilgisi sistemdeki kontrol sonucu alındığı taktirde, talebin veritabanında yer alıp almadığının kontrolü gerçekleştirilir. Veritabanında talebe ait bir bilgi bulunamazsa, sistem, istemciye talebin bulunamadığına dair bir hata mesajı iletir. Eğer talep, veritabanında bulunmuşsa, bu durumda veritabanından, talebin ilişkili olduğu seri numarası bilgisi alınır ve bu seri numarasına sahip sertifika bilgisi istemciye iletilir. Sertifikanın istemciye iletilmesiyle, askıdaki taleplere ait süreç sonlandırılmış olur.

Sistemdeki askıdaki talep durum sorgulama taleplerinin yönetilmesine yönelik sürece ait işlevsel model, Şekil 3.10'da gösterilmektedir. (Bkz. Şekil 3.10)



Şekil 3.10 CMC durum sorgulama talebi süreci entegrasyon modeli

Askıdaki taleplerin durumlarının istemciler tarafından sorgulanmasına yönelik CMC protokolü tanımlarını öngören operasyonel süreç tasarımı, yukarıdaki şekilde genel olarak sunulmuştur. Bu sürecin uygulama tabanlı entegrasyonu örneği, Şekil 3.14'de tasarlanan modelde yer almaktadır. Bu model incelendiğinde, oluşturulacak metodun girdileri arasında asgari olarak, durumunun sorgulanacağı talep seri numarası, sertifika bilgileri ve veritabanı bağlantı nesneleri yer alan ve içeriğinde tamamen veritabanındaki bilgilerle, girdiler arasındaki talep bilgileri arasında gerçekleştirilecek karşılaştırmalı kontrol mekanizmaları aracılığıyla istemciye, talebinin güncel durumunun doğru bir şekilde mesaj bildirimleriyle sağlanması tasarlanmıştır. Bu tasarımın detayları ve ilgili ek mekanizmalar Şekil 3.11'deki örnek tasarım modelinde sunulmuştur. (Bkz. Şekil 3.11)



Şekil 3.11 Askıdaki talep sorgulama talebi süreci uygulama entegrasyon modeli

3.3.1.6 Tüm Yanıt Yönetim

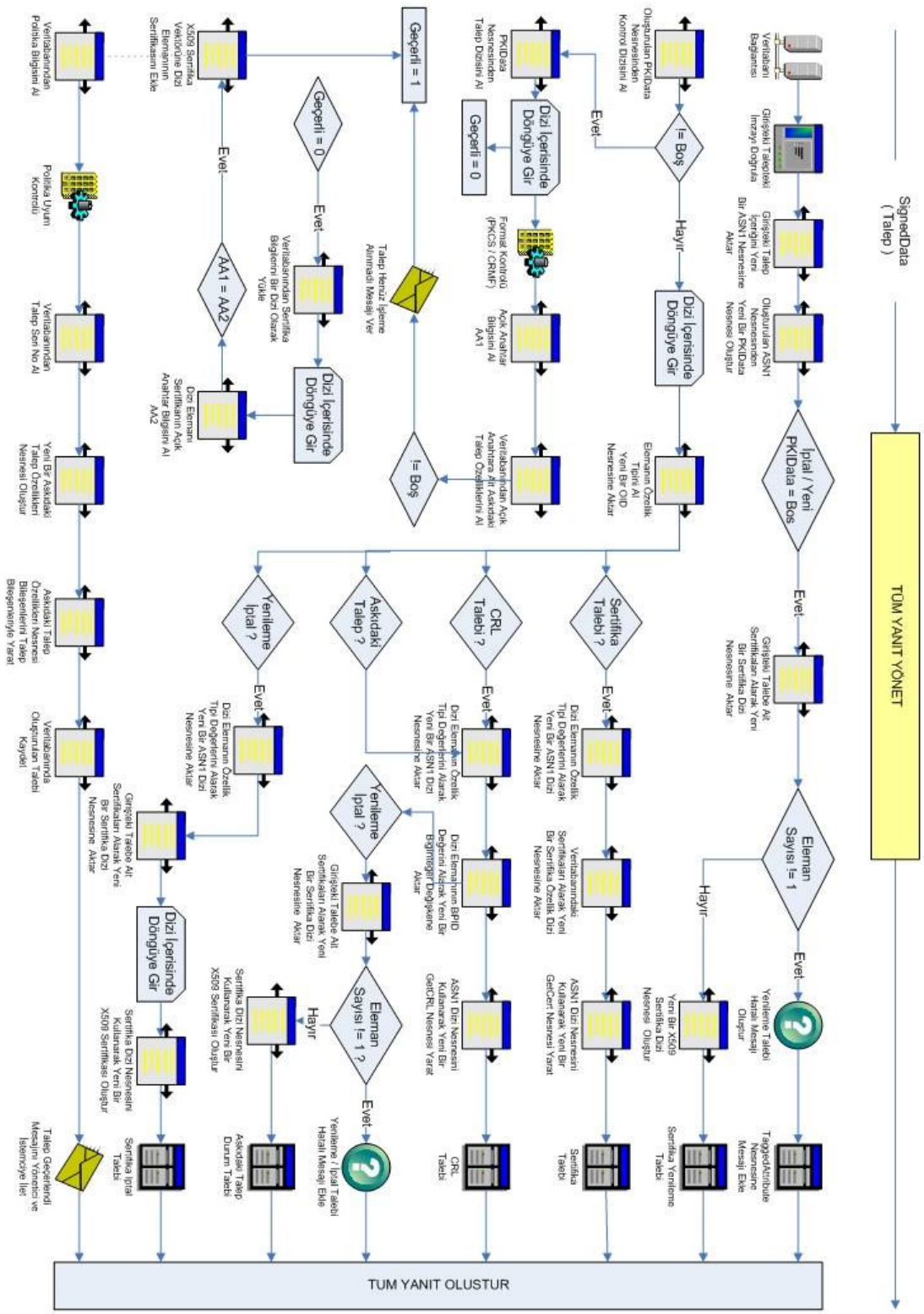
CMC protokolünün sisteme entegrasyonuna yönelik tasarlanan sertifika talep, yenileme, durum sorgulama, iptal ve liste talebi mekanizmaları, uygulama tabanlı yapılandırıldıklarında, burada tanımlanan mekanizmaların da üstünde bir yanıtlama metoduyla bağlantılı olmalıdırlar. Bu sayede, bu beş mekanizma tasarımında oluşturulan ya da içeriklendirilen nesneler, anlamlı bir şekilde CMC protokolü bünyesindeki fonksiyonlara uygun olarak değerlendirilerek, ilgili mekanizmanın gereksinimlerine göre sistem yanıtı oluşturulur. Bu aşama, tüm CMC süreci içerisinde en fazla kontrol ve değerlendirme mekanizması içeren alt süreci oluşturacaktır. Bu alt süreç, temelde bir metod olarak tasarlanmıştır. Bu metodun girdisi, ilgili mekanizmalardan gelen talep nesnesi olurken, çıktısında, bir sonraki aşama içerisinde yapılandırılacak olan, ilgili mekanizma türüne göre değişken yapıya sahip, yanıt bilgisi yer almıştır. Tüm yanıt yönetim sürecinde, sertifika deposu ve talep sunumcuya ait veritabanı üzerinden kontroller gerçekleştirilerek sonuçta, örneğin gelen talep, bir sertifika talebiyse, bir sonraki aşama olan Tüm Yanıt Oluşturma aşamasında, sertifika talebine ait oluşturulacak yanıta ilişkin nesnelerin, bu aşamada oluşturularak iletilmesi sağlanmıştır. Benzer şekilde eğer tasarlanacak metodun girdisindeki talep, bir sertifika iptal listesi talebiyse, bu taktirde, sonraki aşamaya geçirilen nesneler, CMC protokolüne uygun olarak tasarlanmış sertifika iptal listesi talebine ait yanıtla ilişkili nesneler olarak yapılandırılmıştır.

Tüm Yanıt Yönetim metodunun girdisi, önceki bölümlerde tasarlanan CMC mekanizmalarından herhangi birine ait talep formatında SignedData nesnesidir. Bu süreç içerisinde ilk olarak gelen talepteki imza doğrulanarak, talep içeriği, önceki bölümlerde incelenen bir PKIData nesnesine aktarılmıştır. Oluşturulan PKIData nesnesi eğer boş ise, bu durumda talebe ait sertifikalar bir sertifika dizisi nesnesine aktarılır. Eğer dizi hatasız bir şekilde oluşturulursa, bu durumda yaratılan X.509 sertifika dizisi, sertifika yenileme talebine ait olarak bir sonraki Tüm Yanıt Oluşturma aşamasına iletilir. Eğer oluşturulan PKIData nesnesi boş değilse, bu durumda nesnenin içerisindeki kontrol dizisi alınarak dizinin boş olup olmadığı kontrol edilir. Eğer dizi boşsa, PKIData nesnesi içerisinden, talep dizisi incelenerek, dizi içerisindeki elemanlar kontrol edilerek talep format türü bilgisi alınır. Bu format türü CRMF ya da PKCS tabanlı olabilir.

Bu aşamanın sonrasında, talepteki açık anahtar bilgisi ile veritabanındaki açık anahtarla ilgili askıdaki talep durum sorgulama taleplerine ait özellikler dizisi alınır. Eğer askıdaki talep durum sorgulama talepleri dizisi boş ise, bu durumda talepte yer alan açık anahtara ait talebin, sistemde henüz işleme alınmadığı anlaşılarak, istemci bilgilendirilir. Aksi halde, veritabanından sertifikalarla ilgili bilgiler, bir dizi halinde alınarak, talepteki açık anahtarla ilişkili bir sertifikanın sistemde var olup olmadığı kontrol edilir. Eğer açık anahtar bilgisi, talepteki açık anahtar bilgisiyle eşleşen bir sertifika bulunursa, bu sertifika bilgisi X.509 Sertifika Vektörü'ne eklenir. Sonrasında, veritabanından sistem politikası ile ilgili bilgiler alınarak, politika uyumluluğu kontrolü gerçekleştirilir. Eğer politikayla ilgili bir problem yoksa, veritabanından yeni bir talep seri numarası alınarak, taleple ilgili bilgiler, askıdaki talep özellikleri nesnesine aktarılır. Oluşturulan talep veritabanında kaydedilerek saklanır ve istemci, talebinin geçerlendiğine dair bir mesajla bilgilendirilir. Eğer PKIData nesnesine ait kontrol dizisi nesnesi boş değilse, bu durumda dizi içerisindeki elemanlarda döngüye girilerek elemanın özellik tipi değeri edinilir. Bu özellik tipi kullanılarak, girdideki talebin türünün ne olduğu, sertifika talebi, iptal listesi talebi, askıdaki talep durum sorgulama talebi ya da yenileme / iptal talebi seçeneklerinden birisi olarak elde edilir.

Talep, eğer bir sertifika talebiyse, dizi elemanının özellik tipi değerleriyle yeni bir ASN.1 dizisi oluşturulur. Veritabanından sertifikalar alınarak bir sertifika özellik dizisine aktarılır. Son olarak, oluşturulan ASN.1 dizisine ait elemanlarla yeni bir GetCert nesnesi oluşturulur ve Sertifika Talebi türünde Tüm Yanıt Oluştur aşamasına iletilir. Benzer şekilde eğer talep bir sertifika iptal listesi talebiyse, yine aynı mantıkta dizi elemanının özellik tipi değerleri, yeni bir ASN.1 nesnesine aktarılır. Dizi elemanları kullanılarak oluşturulacak GetCrl nesnesi İptal Listesi Talebi türünde Tüm Yanıt Oluştur aşamasına iletilir.

Eğer talep, sertifika iptal talebiyse, dizinin elemanları, önceki işlevlerde olduğu gibi bir ASN.1 dizisi içerisine aktarılır. Talebe ait sertifikalar alınarak bir sertifika dizisi nesnesi oluşturulur. Dizi içerisindeki sertifikalarda döngüye girerek her sertifika için yeni bir X.509 sertifika nesnesi oluşturularak Sertifika İptal Talebi türünde Tüm Yanıt Oluştur aşamasına iletilir. (Bkz. Şekil 3.12)



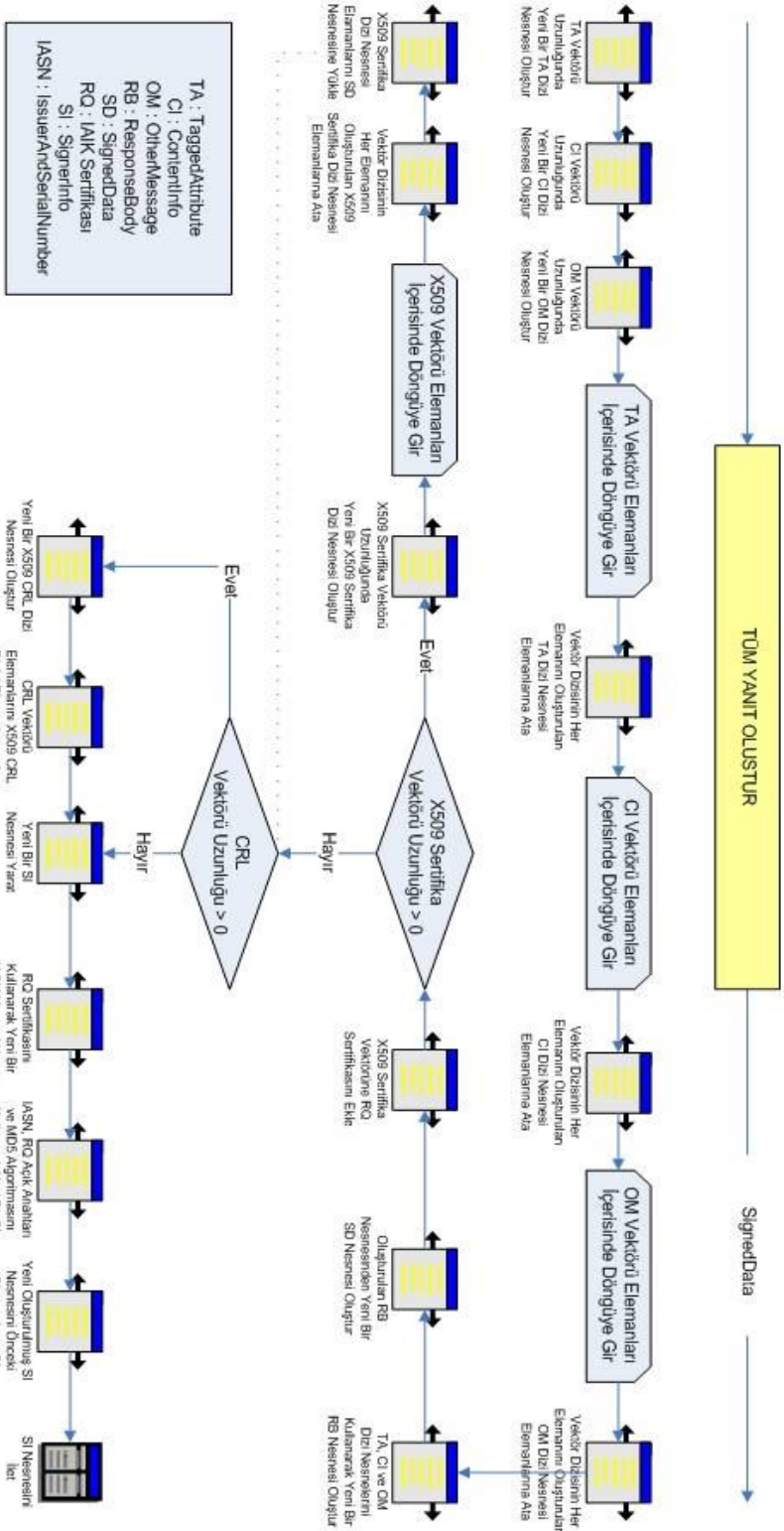
Şekil 3.12 CMC tüm yanıt yönetim süreci uygulama entegrasyon modeli

3.3.1.7 Tüm Yanıt Oluşturma

Bu aşamadan bir önceki aşama olan, Tüm Yanıt Yönetim aşamasında, taleple ilgili yanıt bilgisini oluşturacak veriler yaratılmış ve işlenmek üzere formatlanmıştı. Aynı aşama içerisinde talep türü belirlenmiş ve bu talep türüne uygun olarak sunumcu tarafından iletilecek yanıt bilgisinde yer alması gereken kısımlar, ilgili nesne türleri doğrultusunda oluşturulmuştu. Bu aşamada, basit bir şekilde, oluşturulan bu nesneleri birleştirip değerlendirerek bir yanıt nesnesi oluşturulmuştur. Yanıt nesnesi, daha önceki bölümlerden hatırlanacağı üzere bir SignerInfo nesnesi olarak yer almıştır. SignerInfo nesnesi, imzalanmış bir SignedData nesnesidir. Sonuç olarak bu sürecin sonunda, yine bir SignedData nesnesi türevi oluşacaktır. (Bkz. Bölüm 3.1.3.6)

Tüm Yanıt Oluşturma sürecinde öncelikle TaggedAttribute, ContentInfo ve OtherMessages nesneleri oluşturulur. Sistemde Tüm Talep Yönetim aşamasında oluşturulmuş olan TaggedAttribute, ContentInfo ve OtherMessages dizilerine ait elemanlar, oluşturulan aynı boyuttaki bu yeni dizi nesnelerine aktarılır. Tüm dizilere ait veriler aktarıldıktan sonra, bu üç nesne kullanılarak, önceki bölümlerden hatırlanabilecek olan, yeni bir ResponseBody nesnesi (Bkz. Bölüm 3.1.3.2) yaratılarak SignedData nesnesine dönüştürülür. Önceki aşamadan gelen X.509 Sertifika Vektörü içerisine IAIK sertifikası eklenir ve vektör boyutu kontrol edilir. Eğer vektör boşsa, bu durumda aynı işlemler X.509 CRL Vektörü için gerçekleştirilir. Eğer sertifika vektörü boş değilse, bu durumda vektör elemanları bir diziye aktarılarak dizi içerisindeki elemanlar SignedData nesnesine atanır.

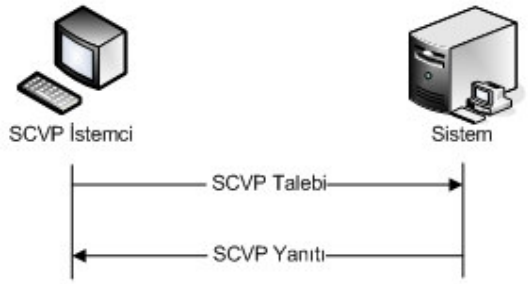
IAIK sertifikası ile yeni bir IASN nesnesi yaratılır. IASN nesnesi, IAIK sertifikası ve ilgili algoritma kullanılarak yeni bir SignerInfo nesnesi oluşturulur ve istemciye iletilir. Bu sürece ait detaylı uygulama entegrasyon modeli gösterimi Şekil 3.13'de sunulmuştur. (Bkz. Şekil 3.13)



Şekil 3.13 CMC tüm yanıt oluşturma süreci uygulama entegrasyon modeli

3.3.2 Sertifika Doğrulamada SCVP Entegrasyonu Modelleri

Çalışmanın bu aşamasında, sisteme entegre edilmek istenilen, SCVP protokolünün öngördüğü sertifika doğrulama mekanizmalarının, sistem içerisinde ne şekilde tanımlanacağı ve yer alacağına yönelik tasarımlar ve modeller sunulmuştur. İlk olarak doğrulama servisi adı altında, sistemde işletilmesi gereken doğrulama işlevlerinin operasyonel süreç modeli oluşturulmuş ve bu servis modeli üzerinden SCVP protokolünün talep ve yanıt fonksiyonlarının entegrasyon modellemeleri gerçekleştirilmiştir.



Şekil 3.14 SCVP İstemci Sunumcu İlişkisi

3.3.2.1 Doğrulama Servisi Süreç Modeli

Sistemde yer alacak doğrulama servisi, önceki bölümlerde üzerinde durulan SCVP protokolüne ait tanımlar gereğince, bir sertifikanın doğruluğunun ya da geçerliliğinin kontrolüne yönelik bir talep aldığında, sistem içerisinde şu adımları izlemelidir :

- Gelen sertifikanın doğrulama servisine ait geçmiş dönem bilgilerinin tutulduğu bellekte yer alıp almadığı kontrol edilir. Eğer sertifikaya bu bellekte rastlanırsa ve ilgili bilgilerin geçerliliği devam etmekteyse, doğrulama servisi talepçi birime olumlu yanıt vererek süreci sonlandırır.
- Sertifikanın güvenilir bir Sertifika Makamı tarafından yayınlanıp yayınlanmadığı kontrol edilir.
    - Eğer sonuç olumlu ise, sertifikanın uzantısında yer alan AuthorityInformationAccess alanındaki bilgi, Sertifika Deposu'ndan yeniden edinim işlemi için alınır, karşı Sertifika Makam tarafından yayınlanan iptal listeleri ve eğer gerekirse Yerel Sertifika Makamı'ndan karşı Sertifika Makamı'na giden yolda yer alan tüm iptal listeleri alınır.
    - Tüm sertifika yolu doğrulanır

    - Eğer doğrulama sonucu olumluysa, algoritma olumlu olarak bir yanıt mesajı ile sonlanır ve bellekte yer alan yeni sertifika yolu ve ilgili bilgiler iletilir.
    - Eğer doğrulama sonucu olumsuz ise, algoritma sonucunda başarısız olarak yanıt döner.
- Hedef sertifikanın AuthorityInformationAccess uzantısındaki bilgi alınır (B)
    - Eğer bu uzantı tanımlanmamışsa bu durumda ;
        - Eğer hedef sertifika bir Kök Sertifika Makamı ise, (C) adımına gidilir
        - Kök Sertifika Makamı değilse, algoritma, bu talebi işlemeye devam edemez ve talepçiye başarısız mesajı gönderir
    - Eğer uzantı tanımlanmışsa, bu uzantıda belirtilen Kök Sertifika Makamı, iptal bilgisi ve çapraz sertifika listesi alınır (tek sorguda bu bilgiler alınabilir) (A)
- Edinilen Kök Sertifika Makamı, iptal listesi ve çapraz sertifika bilgileri, bu talebe istinaden oluşan geçici yapı içerisine aktarılır
- Sertifika Makamı'nın, güvenilir makamlar listesinde olup olmadığı kontrol edilir
    - Eğer bu listedeyse, şimdiye değin oluşturulan yola karşılık bu sertifika doğrulanır
        - Doğrulama sonucu olumluysa, algoritma, olumlu yanıt göndererek başarılı olduğunu talepçiye iletir. Yeni sertifika yolu ve bilgiler doğrulama servisi belleğinde saklanır.
        - Doğrulama başarısız ise, algoritma başarısız mesajı döndürür
    - Eğer listede yer almıyorsa ve Kök Sertifika Makamı na ait AuthorityInformationAccess uzantısı varsa, bu Sertifika Makamı, bir alt Sertifika Makamı olup algoritmanın üst Sertifika Makamı'na ulaşması gerektiği anlaşılarak B adımına dönülür
- Geçerli Kök Sertifika Makamı'nın herhangi bir çapraz sertifika listesine (Bkz. A) sahip olup olmadığı kontrol edilir.
    - Eğer yoksa, algoritma başarısız olduğuna dair yanıt döndürür.
    - Eğer varsa, listedeki çapraz sertifikaların herhangi birinin, güvenilir bir Sertifika Makamı nca imzalanıp imzalanmadığı kontrol edilir
        - Eğer sonuç olumluysa, algoritma başarılı olarak yanıt verir ve yeni sertifika yolu, doğrulama servisine ait bellekte saklanır.
        - Eğer sonuç olumsuzsa, listedeki her sertifika için B adımına dönülerek adımlar tekrarlanır. Herhangi bir anda, yolun geçersiz olduğu doğrulanırsa, son geçerli Kök Sertifika Makamı'na dönülür.

3.3.2.2 Doğrulama Servisi Mekanizmaları

SCVP protokolü tanımında da belirtildiği şekilde, sistemde yer alacak doğrulama servisi, temelde iki farklı tip uygulamaya yönelik doğrulama işlevleri sağlayacaktır. Bu uygulamalar, sertifika doğrulama yapabilen ancak sertifika yolu oluşturamayan uygulamalar ve sertifika yolu oluşturma ve doğrulama işlemlerinin her ikisini de gerçekleştiremeyen uygulamalardır. Bu iki tip uygulamanın desteklenebilmesine yönelik, sistemdeki doğrulama servisi tasarımı, iki temel mekanizmayı içermelidir. Bunlar, sertifika yolu oluşturma ve sertifika yolu oluşturma ve doğrulama mekanizmalarıdır.

Bir sertifika, her ne kadar içeriğinde geçerliliğinin kontrol edilebilmesi amacıyla bir zaman periyodu bilgisine sahip olsa da, sistem içerisinde farklı nedenlerden ötürü tamamen geçersiz hale gelebilir. Örneğin, sertifika, ait olduğu Sertifika Makamı tarafından bir sertifika iptal listesi yayını ile iptal edilebilir. Benzer şekilde bir sertifikanın, uzlaşma gereği, aynı makam ya da farklı bir makam tarafından yayınlanan bir diğer sertifikayla değiştirilmesi gerekebilir. Bu nedenler sonucunda sertifika doğrulama, bağlantılı olduğu açık anahtarın, sistem içerisindeki her kullanımının öncesinde yer alan bir işlev olarak tasarlanmalıdır. Bu tip bir prosedür, içeriğinde bazı doğrulama kontrollerini barındıracaktır. Bunlar temelde, güven zincirinin doğrulanması, sertifika iptal durumlarının saptanması ve sertifika kullanımının belirlenmesi olarak tanımlanabilir.

3.3.2.3 Güven Zincirinin Doğrulanması

Kök Sertifika Makamı'ndan, uç birime uzanan güven zincirinin saptanması işlemidir. Bu zincir, hem yatay hem de çapraz etki alanları arası hiyerarşik bir modelde hibrit bir yapıyı içerebilmektedir. Tamamen hiyerarşik bir güven mimarisi içerisinde, ilk olarak, sistem içerisinde oldukça fazla güvene sahip Kök Sertifika Makamı, hemen altında yer alan Sertifika Makamı'na ait sertifikanın bütünlüğünü ve güvenilirliğini doğrulamak zorundadır. Bu işlem, hiyerarşik ağaç yapısı içerisinde dallanarak, uç birime ait sertifikanın doğrulanmasına kadar benzer şekilde işletilir. Yatay olarak çapraz sertifikaların müdahil olduğu ortamlardaysa, çapraz sertifikaya sahip Sertifika Makamı'nın sertifikası, bu makamı sertifikalayan eş Sertifika Makamı'nın anahtarı kullanılarak doğrulanır.

3.3.2.4 Sertifika İptal Durumlarının Saptanması

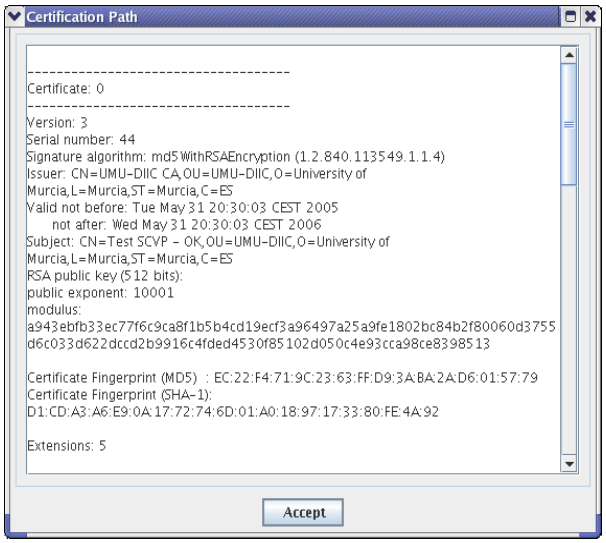
Bir sertifikanın, ilgili sertifika iptal listesi içerisinde yer alıp, yer almadığının belirlenmesi işlemi olarak nitelendirilebilir. Burada adı geçen sertifika, kendisini yayınlayan makamla ilişkili olarak tekil bir seri numarası ile tanımlanır. Doğrulama prosesi, bu iki elamanın, iptal edilmiş sertifikalar arasında olup olmadığını tespit etmek üzere tasarlanır. Genelde sertifika iptal listeleri, sistem içerisinde bir ortak ağ paylaşım noktasında yer alan LDAP deposu içerisinde tutulur. Bu işlem, çoğu zaman bir sunumcuya devredilir.

3.3.2.5 Sertifika Kullanımının Belirlenmesi

Herhangi bir uygulanabilir sistem politikasına karşı sertifikanın kullanımının doğrulanması işlemidir. Örnek olarak, sertifika içerisinde yer alan anahtar kullanım uzantısı bilgisine karşılık kullanılacak olan açık anahtarın, kriptografik fonksiyonunun değerlendirilerek karşılaştırması verilebilir.

3.3.2.6 Sertifika Kullanımının Belirlenmesi

Herhangi bir doğrulama işlemi yapılmaz. SCVP sunumcusu sadece kullanıcı tarafından belirtilen güvenilir tutucu birimleri kullanarak, ilgili sertifikaya uygun sertifika yolunun kurulmasını sağlar. Kullanıcı, daha sonra bağlantısız bir şekilde bu yolu doğrulayabilecektir. Eğer SCVP sunumcu başarılı bir şekilde sertifika yolunu oluşturabilirse, kullanıcı, oluşturulmuş sertifika yoluyla birlikte pozitif bir SCVP cevabı alır. Aksi taktirde kullanıcı, SCVP sunumcudan sertifika yolunun kurulamadığına dair gerekçeli hata mesajını alacaktır.



Şekil 3.15 Örnek Sertifika Yolu Bilgileri

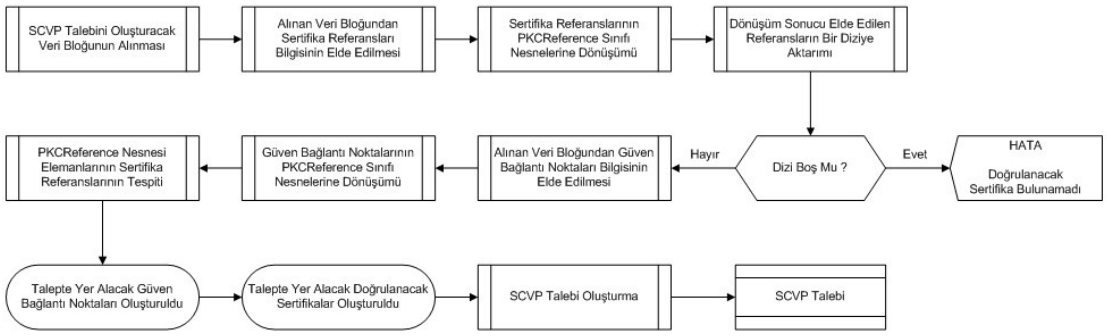
3.3.2.7 Sertifika Yolu Oluşturma ve Doğrulama

Bu servis ise SCVP sunumcunun kullanıcıdan almış olduğu doğrulanacak sertifika ve kullanıcının güvendiği güvenilir tutucu birimler bilgileri doğrultusunda, geçerli bir sertifika yolu kurmaya çalışır. Eğer SCVP sunumcu bu işlemde başarılı olursa, kullanıcı hem pozitif bir SCVP yanıtı hem de sertifika yolu bilgisini alır. Aksi halde önceki serviste belirtilen şekilde kullanıcı, bir hata kodu ve bu hatanın nedeni ile ilgili açıklayıcı bir bilgi alır. Bu servis öncekiyle kıyaslandığında, sistemde yer alan uç birimlerin sertifika ve doğrulama işlemlerindeki yüklerini azaltmaktadır.

3.3.2.8 Doğrulama Talepleri

Sistem içerisinde doğrulama taleplerinin entegrasyonunu sağlayabilmeye yönelik olarak öncelikle, doğrulama ya da SCVP taleplerinin sistemsel süreçler içerisinde ne şekilde oluşturulacağı konusunda çalışma yapmak gerekmektedir. Bu açıdan bakıldığında, SCVP protokolünün doğrulama mekanizmalarının sistemde sağlanabilmesi için sistem birimleri tarafından gerçekleştirilebilecek doğrulama talep formatının belirlenmesi gerekir. Doğrulama talep formatı SCVP protokolünün tanımına uygun şekilde işlevlere sahip olmalı ve aynı zamanda sistemin gereksinimlerini de büyük ölçüde karşılayabilmelidir. Sistem içerisinde uygulama bazlı entegrasyon göz önünde alınarak bu talep formatı düşünüldüğünde, doğrulama taleplerinin standart bir yapıda sistemdeki tüm birimlerce uygulanabilir bir formatta yapılandırılması gerekliliği mevcut olup, bu yönde mümkün oldukça geniş platformlarca uygulanması sağlanabilecek nesnelerin entegrasyonu üzerinde durulmalıdır.

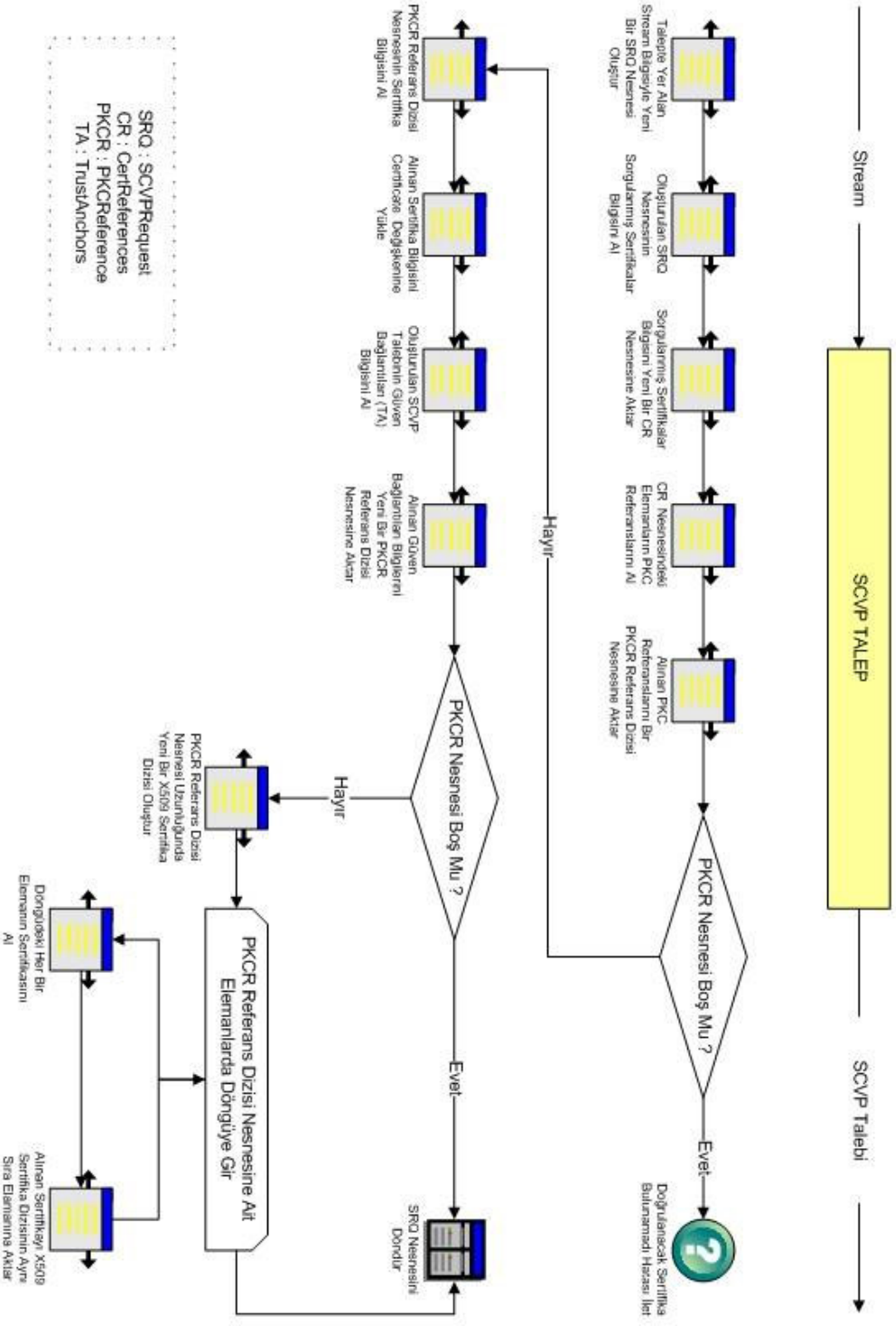
Doğrulama taleplerinin, SCVP protokolü gereklerince sistem içerisinde yönetilmesini içeren süreç entegrasyon modeli Şekil 3.16'da sunulmuştur. (Bkz. Şekil 3.16)



Şekil 3.16 SCVP doğrulama talebi süreci entegrasyon modeli

Bu süreç modelinden yola çıkarak doğrulama taleplerinin sisteme uygulama tabanlı uyarlanmasına yönelik modellemeler gerçekleştirilmiştir. SCVP doğrulama talep süreci, temelde bir girdi ve bir çıktıya sahip olan bir metod olarak tasarlanmıştır. Burada bahsi geçen girdi, en genel haliyle stream tipinde bir veri bloğu, çıktı ise bir SCVP doğrulama talebidir. Metod içerisinde bu arada meydana gelecek işlemlerse, ilgili veri bloğunun alınması ile başlayarak, talebin oluşturulması ile sonlanacak bir süreç içerir. Talebin oluşturulması olarak nitelendirilebilecek bu süreçte, metod içerisinde öncelikle gelen veri bloğu ile ilişkili sertifika referansları oluşturulmuş ve bu sertifika referansları üzerinden talep yapılandırması tamamlanmıştır.

Bu yapılandırma içerisinde öncelikle, metod girdisinde yer alan veri içerisinden ilgili sertifika referanslarına yönelik kayıtların oluşturulması ve bu referansların JAVA da tanımlı PKCReference sınıfındaki nesnelere dönüşümü sağlanmıştır. Buna paralel olarak yine metod girdisinden gelen veri bloğu içerisinden, güven bağlantı noktaları alınarak, bu bilgiler de benzer şekilde PKCReference sınıfı nesnelerine dönüştürülmüştür. Bu iki temel dönüşümün sonucunda elde edilen PKCReference nesneleri bir kontrol döngüsü aracılığıyla eşleştirilerek oluşturulan nesne SCVP talebi olarak sonlandırılır. Uygulama tabanlı örnek bir doğrulama talebi entegrasyonu tasarımı Şekil 3.17'deki modelde sunulmuştur. (Bkz. Şekil 3.17) Önceki bölümdeki süreç modelinin, uygulama içerisinde ne şekilde detaylandırılacağı ve uygulama temelli tasarımı, bu model üzerinden incelenebilir.

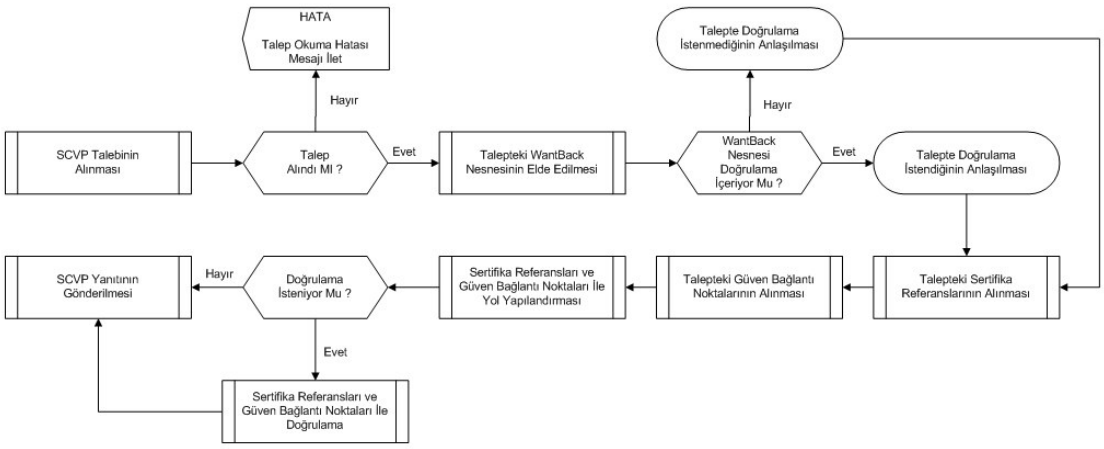


Şekil 3.17 SCVP doğrulama talebi süreci uygulama entegrasyon modeli

3.3.2.9 Doğrulama Mekanizmaları Yönetimi

Sistemdeki sertifikaların ve sertifika yollarının doğrulanmasında, doğrulama taleplerinin sistemde SCVP protokolüne uygun olarak ne şekilde oluşturulacağını içeren süreç modeli önceki bölümde tasarlanmıştı. İkinci aşamada, sistemde gerçekleştirilecek bu tür taleplerin, protokol mantığını içerecek biçimde yönetim ve sistem yanıtının oluşum mekanizmaları, genel anlamda tasarlanmıştır. Talebin sistemde alınması ile başlayan bu süreçte talepten elde edilecek veriler doğrultusunda, doğrulama yanıtının oluşturulmasına yönelik işlevler ele alınmıştır. SCVP protokolüne ait mekanizmaların incelendiği ikinci bölümden de hatırlanacağı üzere, bir doğrulama talebine yönelik yanıtın oluşumunda talep formatında tanımlanmış olan WantBack nesnesi öne çıkmaktadır. Bu nesne tanımlamasındaki veriler doğrultusunda istemci birimler tarafından gerçekleştirilen talebe karşılık sistemden tam olarak ne tip bir yanıtın döndürülmesi gerektiği belirtilmektedir.

Doğrulama mekanizmalarının sistem içerisinde yönetimini öngören süreç modeli Şekil 3.18'deki gösterimde sunulmuştur. (Bkz. Şekil 3.18)

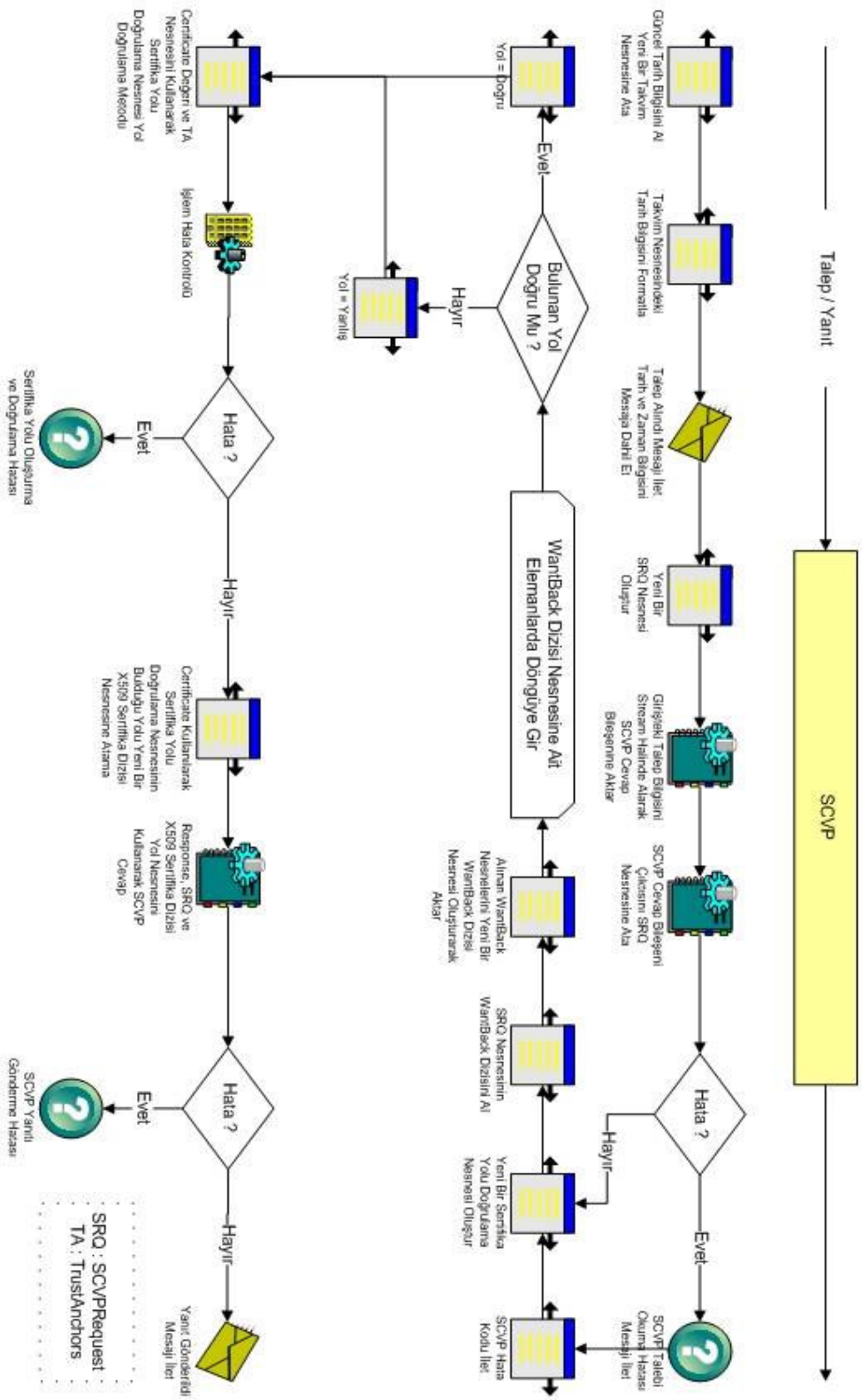


Şekil 3.18 SCVP doğrulama yönetimi süreci entegrasyon modeli

Modeldeki doğrulama talepleri ve yanıtlarına yönelik yönetim işlevleri, SCVP protokolünü temel alan ve sistem mimarisine uygun bir yapıda tasarlanmıştır. Doğrulama talebi sistemde alındığında, ilk olarak talep verilerinin tam ve doğru bir şekilde alınıp alınmadığı kontrol edilir. Sürecin her aşamasında olduğu gibi doğrulama mekanizmalarının yönetimi aşamalarında da işlem hata kontrolleri ve ilgili mesaj bildirimleri gerçekleştirilmiştir.

Talepteki veri bloğundan elde edilecek WantBack nesnesinin içeriği alınarak, talepten sadece sertifika yolunun kurulması mı yoksa sertifika yolunun kurularak sertifikaların doğrulanması mı istendiği anlaşılır. Sürecin nasıl işletileceği anlaşıldıktan sonra, talepte yer alan sertifikalar alınarak sertifikalara ait referans bilgileri elde edilir. Ek olarak sertifika referanslarının paralelinde talepteki güven bağlantı noktaları bilgileri de referans bilgilerine eklenir. Sertifika referansları ve güven bağlantı noktaları bilgileri kullanılarak sertifika yolu doğrulaması gerçekleştirilir. Bu aşamada UMU-PKI SCVP entegrasyonunda yer alan SCVP ASN.1 sınıfına ait metodlardan yararlanılır. Son aşamada edinilen veriler, doğrulama yanıt iletimi aşamasında doğrulama yanıtının oluşum ve istemci birime iletiminde kullanılır.

Doğrulama yönetim mekanizmalarının entegrasyonuna yönelik uygulama bazlı örnek bir tasarım Şekil 3.19'da sunulan modelde sunulmuştur. (Bkz. Şekil 3.19) Bu uygulama modeli, temelde önceki bölümde üzerinde durulan operasyonel süreç modeli temel alınarak geliştirilmiş olup, tasarım içerisindeki mekanizmalar SCVP protokolünün yönetim işlevlerini doğrudan kapsamaktadır.

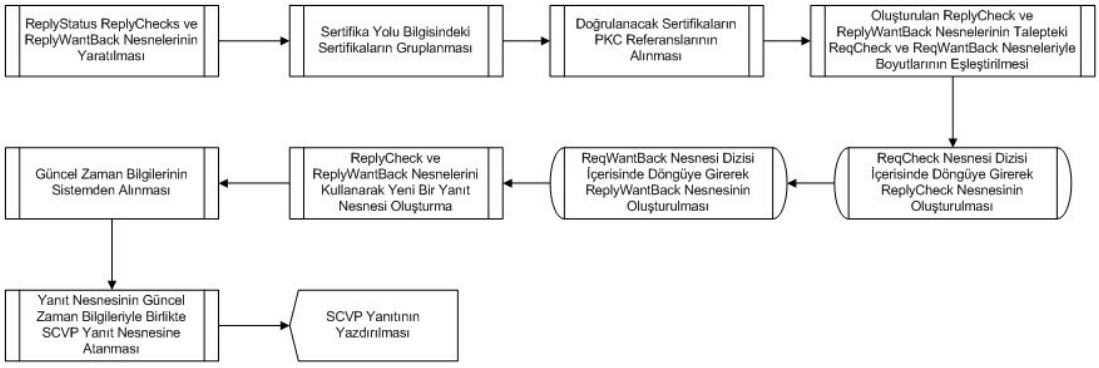


Şekil 3.19 SCVP doğrulama yönetimi süreci uygulama entegrasyon modeli

3.3.2.10 Doğrulama Yanıtları Bildirimi

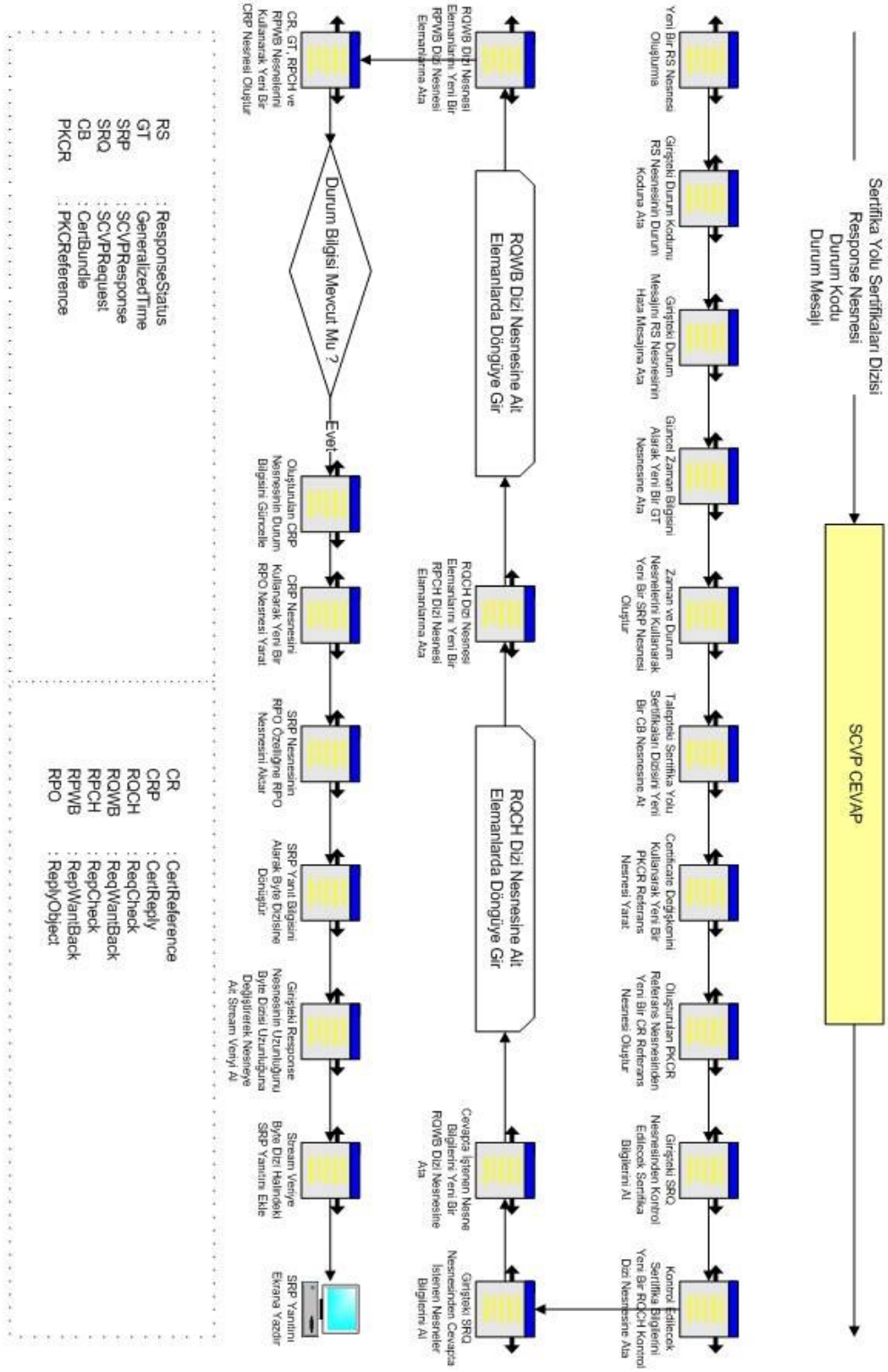
Önceki bölümlerde, doğrulama servisinin entegrasyon tasarımında sistemdeki doğrulama taleplerinin ne şekilde modelleneceği ve talepten elde edilen verilerden yararlanarak doğrulama yanıtının temelini teşkil edecek nesnelerin oluşturulması üzerinde durulmuştu. Sistemdeki doğrulama yanıtının oluşturulacağı bu kısımda, temelde bir SCVP yanıtını oluşturan nesneler tamamlanarak, yanıtın istemciye bildirimi modellenmiştir. Bu süreç içerisinde sertifika yolu oluşturulmuş olup, taleple birlikte bu aşamanın girdisi olarak yer almıştır. SCVP protokolünün tanım ve detaylandırılması bölümlerinde üzerinde durulan ReplyCheck ve ReplyWantBack nesneleri bu süreç içerisinde öncelikli olarak yaratılmıştır. Bu nesnelerin yaratılmasından sonra sertifika yolu bilgisi içerisinde yer alan sertifikalar gruplanarak doğrulanacak sertifikalara ait referanslar yeniden elde edilir. Talepteki ReqCheck ve ReqWantBack nesnelerinin boyutları, oluşturulan ReplyCheck ve ReplyWantBack nesneleriyle uyuşmalıdır. Sonraki aşamada talepten gelen ReqCheck nesnesindeki veriler kullanılarak ReplyCheck nesnesi ve aynı şekilde ReqWantBack nesnesindeki verilerle de ReplyWantBack nesnesine ait tanımlamalar gerçekleştirilir. SCVP yanıtının temelini teşkil edecek ReplyCheck ve ReplyWantBack nesnelerinin oluşması sonucunda, süreçte yeni bir yanıt nesnesi oluşturularak sistemin güncel zaman bilgileriyle birlikte SCVP doğrulama yanıt nesnesine aktarılır. Doğrulama yanıtı yazdırılarak istemciye bildirilir.

Genel olarak teknik entegrasyonu tasarlanan süreç modeline ait gösterim Şekil 3.20'deki yapıya uygun biçimde sistemde işletilir. (Bkz. Şekil 3.20)



Şekil 3.20 SCVP doğrulama yanıt süreci entegrasyon modeli

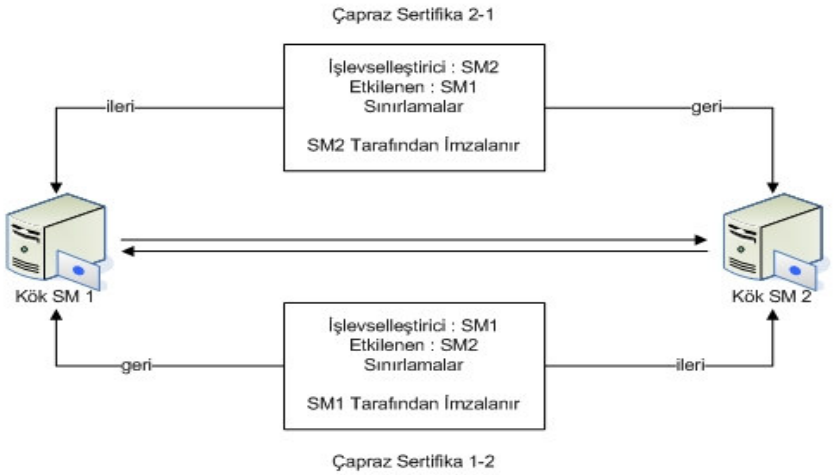
Bu aşamada aktarılan doğrulama yanıtlarına yönelik işlevsel süreç modeli baz alınarak, doğrulama taleplerinin SCVP protokolü uyarınca sisteme entegrasyonunu öngören uygulama bazlı tasarım modeli oluşturulmuştur. Uygulama tabanlı oluşturulan bu model, JAVA tabanlı olup, temelde önceki bölümde aktarılan süreçte yer alan işlevlerin teknik olarak gerçeklenme tasarımını meydana getirmiştir. Örnek bir uygulama tabanlı entegrasyon, Şekil 3.21'deki modelde tasarlanmış olup, buradaki süreçte yer alan mekanizmaların detayları ve uygulama bazlı metodoloji, bu model üzerinden incelenebilir. (Bkz.Şekil 3.21)



Şekil 3.21 SCVP doğrulama yanıt süreci uygulama entegrasyon modeli

3.3.3 UMU-PKI v6 Çapraz Sertifika Desteği

Farklı etki alanlarına sahip yapılar içerisinde yer alan kaynakların, güvenilir bir şekilde ortak olarak kullanılması söz konusu olduğunda, bu etki alanları arasında belirli oranlarda güven ilişkilerinin kurulması gerekmektedir. Örneğin, kendi içinde bulunduğu etki alanındaki kullanıcılara yönelik web servisleri sağlayan ya da kullanıcının tanımlı olduğu etki alanına bağlı olarak, internet erişim imkanı sunan bir servis sağlayıcısını ele alındığında, her bir etki alanının, entegre bir açık anahtar sistemine sahip olduğu varsayılırsa, bu güven ilişkilerinin tanımlanmasında birden fazla alternatif yöntem ortaya konabilir. Temel çapraz sertifikalama modelleri, birbirleri arasında çapraz sertifika işlemlerine izin veren, farklı Sertifika Makamlarının birleşimi sonucunda ortaya çıkmaktadır. Bir çapraz sertifika, birbirleri arasında güven ilişkisi kurmak isteyen Sertifika Makamlarından biri tarafından, diğeri için yayınlanmış bir X.509 tabanlı açık anahtar sertifikasıdır. (Bkz. Bölüm 2.3.3) Bu nedenle bu tip sertifikaların dağıtıcı ve etkilenen özelliklerinin tanımları, farklı Sertifika Makamlarını göstermektedir. A etki alanındaki bir Kök Sertifika Makamı tarafından model incelendiğinde, hem dağıtıcısı B etki alanındaki bir Kök Sertifika Makamı ve etkileneni A etki alanındaki Kök Sertifika Makamı olan, bir ileri yönlü sertifika, hem de dağıtıcısı A etki alanındaki Kök Sertifika Makamı ve etkileneni B etki alanındaki Kök Sertifika Makamı olan, bir geri yönlü sertifikanın, her ikisi de gözlemlenebilir. Tersine, B etki alanındaki Kök Sertifika Makamı tarafından bakıldığında, ileri yönlü sertifika geri yönlü ve tersi olarak izah edilir. Bu çapraz sertifikaların her biri, tanımlanmış oldukları her yön için, sertifika yolu uzunluğu, isim ve / veya sertifika politikası kısıtları ya da yasakları gibi sınırlamaları içerebilir. Şekil 3.22'deki gösterimde, bu tip bir çapraz sertifika senaryosu sunulmuştur. (Bkz. Şekil 3.22)



Şekil 3.22 Çapraz sertifika örneği

3 farklı temel çapraz sertifika modeli vardır. Bunlar, Hiyerarşik, Karşılıklı (P2P) ve Köprü Sertifika Makamı modelleridir. [Dzambasow, Y. et. al, 2005]

3.3.3.1 Hiyerarşik Çapraz Sertifikalama

Günümüz teknolojileri içerisinde sıklıkla tercih edilen bu çapraz sertifika modelinde, uç birimlerin sertifika işlevleri için, sistemde tek bir Kök Sertifika Makamı yer almaktadır. Çapraz sertifika şekli, sistemde tek yönlü olarak işlemektedir. Bu nedenle dağıtıcı tanımlamasında Kök Sertifika Makamı'na ait ismin ve etkilenen tanımlamasında da Sertifika Makamlarının hiyerarşisi içerisinde, ikincil durumdaki Sertifika Makamı'na ait ismin yer aldığı, tek bir çapraz sertifika bulunur. Bu modelde ikincil Sertifika Makamı, sistem hiyerarşisi içerisinde üstünde yer alan Sertifika Makamı'na yönelik bir çapraz sertifikayı yayınlayamaz, ancak kendi seviyesi altında yer alan Sertifika Makamlarına yönelik çapraz sertifika işlemini gerçekleştirebilir.

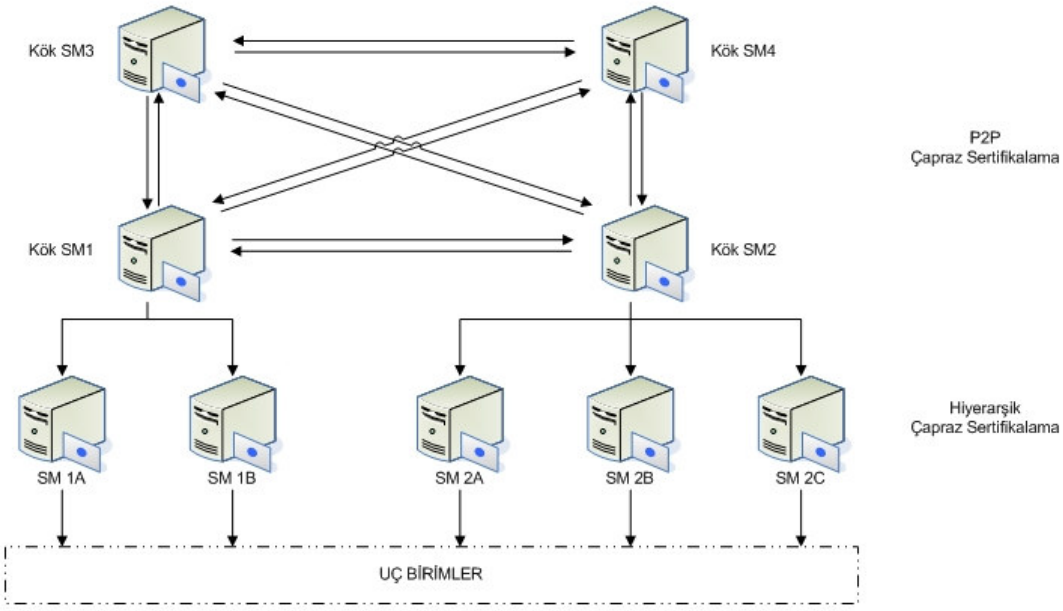
3.3.3.2 Karşılıklı (P2P) Çapraz Sertifikalama

Bu modelde, ilgili Kök Sertifika Makamları arasında, birden fazla çapraz sertifikaya izin verilebilmektedir. Çoklu çapraz sertifika işlemlerinin gerçekleştirilmesinin gerekli olduğu benzer ağlarda, bu modelde ortaya çıkabilecek bir ölçekleme sorunu, söz konusudur. İçinde n adet Kök Sertifika Makamı'nın bulunduğu bir senaryo içerisinde, tamamen güvenilir bir ortamın oluşabilmesi için, n(n-1)/2 adet güven ilişkisinin (Bkz. Bölüm 1.6.11) sistem içerisinde yapılandırılması gerekmektedir. Bu şekilde, sertifika yollarının kurulması ve doğrulanmasına yönelik işlemler, iki birim arasında birden çok sertifika yolu oluşabildiği için, hiyerarşik çapraz sertifikalama modeline kıyasla, daha zor olmaktadır. Örneğin, aşağıdaki senaryoda belirtilen, SM_{1A} makamı altındaki bir uç birim, Kök SM_2 makamı altındaki birimi, Kök SM_2 makamını güven sahibi olarak kullanmak suretiyle doğrulamak isterse, bir sertifika yolu yapılandırma algoritması kullanarak, aşağıdaki gibi bir ya da daha fazla sertifika yolu alternatifi oluşturulabilir.

Burada her ne kadar alternatif olarak bu yollar belirtilmişse de, sistemdeki kısıtlar (sertifika yolu uzunluğu, adı ...) düşünüldüğünde, bu alternatiflerin aslında birer sertifika yolu adayı olarak belirtilmesi daha anlamlı olacaktır.

- SM_{1A} uç birim → SM_{1A} → Kök SM_1 → Kök SM_3 → Kök SM_4 → Kök SM_2
- SM_{1A} uç birim → SM_{1A} → Kök SM_1 → Kök SM_4 → Kök SM_2
- SM_{1A} uç birim → SM_{1A} → Kök SM_1 → Kök SM_3 → Kök SM_2
- SM_{1A} uç birim → SM_{1A} → Kök SM_1 → Kök SM_4 → Kök SM_3 → Kök SM_2

Belirtilen yapıya ait gösterim, Şekil 3.23'de sunulmuştur. Şekilden de anlaşılabileceği gibi, iki farklı çapraz sertifika yöntemine yönelik bileşenler tek bir yapı altında da sağlanabilir.



Şekil 3.23 P2P çapraz sertifikalama

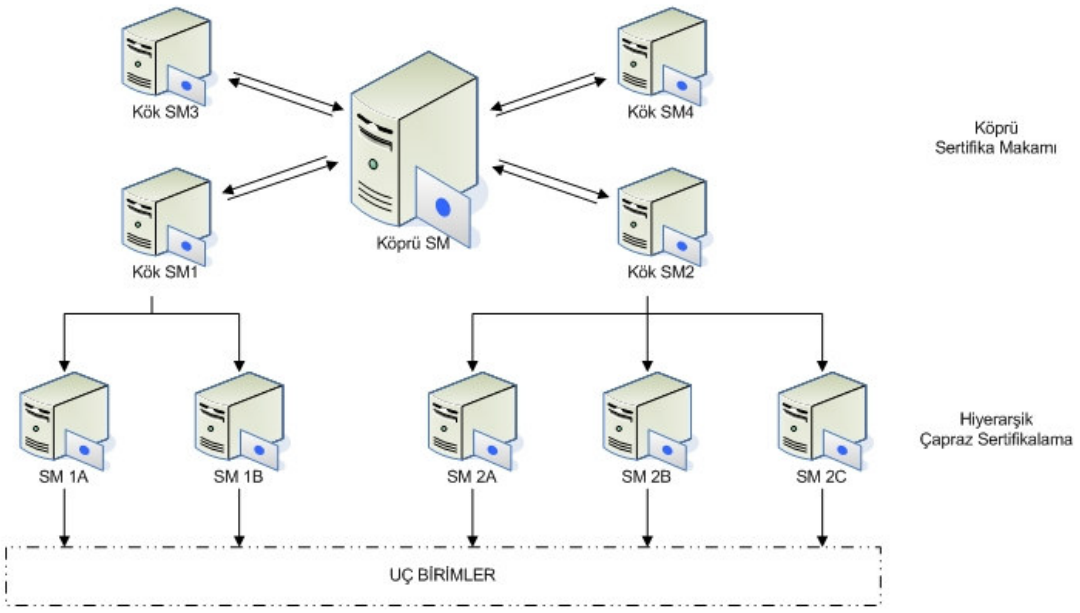
Önceki bölümde tanımlanan formül göz önüne alınırsa, sistemde 6 adet ilişkinin kurulması gerektiği görülebilir. Kurulacak her ilişki, çift yönlü olacağından, toplamda 12 adet çapraz sertifikaya ihtiyaç duyulacaktır.

3.3.3.3 Köprü Sertifika Makamı Modeli

Bu model, karşılıklı (P2P) modelde bahsedilen ölçekleme problemini gidermeye yönelik olarak, etki alanı içerisinde güven ilişkilerinin kurulabildiği bir yapı sağlamaktadır. Sistem içerisindeki bir birim, kendine bağlanan farklı etki alanları arasındaki güven ilişkilerini kurar. Önceki bölümde tanımlanmış olan çapraz sertifika modellerini temel alır. Sistem içerisindeki ilişkilerin sayısı, bağlantılı olan Kök Sertifika Makamı sayısına (n) düşmektedir. Bu modelin

bir başka avantajı ise, sertifika yolu sayısının düşmesidir. Kurulabilecek güven ilişkisi sayısında düşüş olduğundan, oluşabilecek sertifika yolları daha kolay yapılandırılabilir ve doğrulanabilir. Aşağıdaki şekilde, bu yeni modelde kullanılan sertifika yolu kurma algoritması sonucunda tek bir aday sertifika yolu olasıdır. Bu nedenle köprü sertifika makamı modeli kullanılarak sertifika yolu oluşturma ve doğrulama çok daha kolay gerçekleşir.

- SM_{1A} uç birim → SM_{1A} → SM_{1A} → Kök SM_1 → Köprü SM → Kök SM_2



Şekil 3.24 Köprü sertifika makamı modeli

Yukarıdaki şekilde temel bir Köprü Sertifika Makamı Çapraz Sertifika senaryosu tasarlanmıştır. Karşılıklı (P2P) Çapraz Sertifika modeliyle kıyaslandığında bu modelde, her bir Kök Sertifika Makamı ve Köprü Sertifika Makamı arasında 4 adet ilişki tanımlamasına gerek vardır. Bundan dolayı her bir ilişkinin çift yönlü olarak ele alındığı varsayılırsa toplamda 8 adet çapraz sertifikaya gerek olur.

UMU-PKI v6 sistemi, Hiyerarşik, Karşılıklı (P2P) ve Köprü Sertifika Makamı olarak önceki bölümde incelenmiş olan 3 farklı çapraz sertifikalama modelini de desteklemektedir.

4. SONUÇLAR

Geleceğin dağıtık uygulamaları, büyük ölçekli, çok geniş ortamlara açık, çoğu zaman karmaşık ve birbirleriyle fazlasıyla etkileşimli olma özelliklerine sahip uygulamalar olarak, gelecekte yerlerini alacaktır. Bu tip uygulamaların geliştirilmesindeki kritik gereksinimlerin başında, güçlü, ölçeklenebilir, esnek ve geliştirilebilir bir yetkilendirme yönetim altyapısı oluşturma gelmektedir. Özellikle bilgi güvenliği alanında, optimum bir çözüm sağlayacak yol ya da yönteme karar vermek, birçok açıdan oldukça zordur. Bu zorluğun temelinde, devasa boyutlarda değişim ve gelişime maruz kalan teknolojik kavram ve uygulamalara karşı geliştirilen yöntemlerin, çoğu zaman yetersiz kalarak, amacına uygun şekilde bir destek sağlayamamasıdır. Açık anahtarlar ve sayısal sertifikaların dağıtım ve yönetimi merkezli yapılar sağlayan açık anahtar altyapıları, güvenli iletişim ve elektronik tabanlı sayısal işlemlerin, gerek siber ortamda gerekse gerçek hayatta en yaygın kullanılan ve en itibarlı temel teknolojisi olarak yerini almıştır.

1981 yılında RFC-791 ile standartlaşan internet protokolü, takip eden yıllar içerisinde katlanarak büyüyen bir bilgi ve iletişim ağının oluşmasını sağlamasına rağmen, sürekli kendini geliştiren ve yenileyen bilişim yapılarına altyapı oluşturmadaki eksiklikler ve yetersizliklerle, geleceğin teknoloji tabanlı dünyasına artık cevap veremeyecek hale gelmiştir. Mobil teknolojilerin artık hemen her alanda kendini kabul ettirmesi, iletişim çağı olarak tabir edilen 21. yüzyıl araçlarının, sektör ayrımı olmaksızın, uygulayıcılar tarafından piyasa talebi doğrultusunda, birçok teknolojiyi bir bütün olarak içerme gerekliliği ve bilgi paylaşım hız ve yöntemlerinin, geçtiğimiz yüzyıla kıyasla, yeni bir çağ olarak nitelendirilebilecek düzeyde değişim geçirmesi gibi pek çok sebep doğrultusunda, bu kavramların teknolojik olarak altyapısını teşkil eden internet yapısının da yenilenmesi ve geliştirilmesi yönündeki çalışmaların artması kaçınılmazdır. Bu çalışmaların başındaysa, güncel internet protokolü sürümünün geliştirilmesiyle ortaya çıkan ve IPv4 yetersizliklerini, sahip olduğu tasarım ve entegrasyon mantığıyla, günümüzde ve yakın gelecekte giderebilecek özelliklere sahip, yeni nesil internet protokolü olarak da adlandırılan IPv6 protokolü çalışmaları gelmektedir. [Deering, S. and R. Hinden,1998]

Geliştirilen IPv6 tabanlı teknolojik platformlar, günümüz interneti ve bilgi iletişim araçları göz önünde bulundurulduğunda, mutlak surette farklı güvenlik unsurlarına karşı uygulanabilecek tedbir mekanizmaları ve gizlilik altyapılarını içermek zorundadır. Sertifika temelli yöntemler, sayısal verinin simetrik anahtarlarla iletimi ve açık anahtar kriptografisinin günümüzdeki güçlü ve dağıtık uygulamaları, geleceğin IPv6 tabanlı internet altyapısının, sağlam ve geliştirilebilir güvenlik mekanizmalarına sahip olmasında, güncel çözümler arasında, üzerinde en çok düşünülmesi gereken yöntemlerin başında gelmektedir. Henüz üzerinde çalışmaların sürdürüldüğü UMU-PKI sistemi, bu amaçla geliştirilmiş bir açık anahtar altyapısı olup, sahip olduğu IPv6 desteğiyle bir ilki gerçekleştirmeyi hedeflemektedir.

Bu çalışmada, JAVA temelli UMU-PKI altyapısının öngördüğü sistem, detaylı olarak incelenmiş, bu sisteme ait bileşenler ve servisler, sistemin içerdiği güvenlik ve veri gizliliği koruma karakteristikleriyle, sertifika teknolojileri bazında detaylandırılarak, bu altyapı üzerinde yapılandırılabilecek servisler tanıtılmıştır. Açık anahtar altyapısı olarak geliştirilmekte olan bu model içerisinde, hiç şüphesiz en temel işlevler, sayısal sertifikalar bazında teşkil eden mekanizmalar olacaktır. Sistem güvenilirliği ve işletilen verilerin güvenliği düşünüldüğünde, altyapı bünyesindeki veri haberleşmesi ve süreç yönetimlerinin içerdiği mekanizmaların, sistem bütününün güvenliğini ve uygulanabilirliğini direk olarak etkilediği bir gerçektir. Sertifika yaşam döngüsü olarak adlandırılan, sertifika tabanlı bu işlevlerden bazıları, sistemde sertifika taleplerinin nasıl ifade edileceği, sertifikalar üzerinde altyapının ne gibi servisler sunabileceği, farklı güvenlik politikalarının ne şekilde sistem içerisinde yer alabileceği, sistem içerisindeki temel güvenlik ve gizlilik mekanizmalarının hangi faaliyetlerle ne gibi servisleri destekleyebileceği ve bu birimlerin otorizasyona bağlı bilgileri birbirleri arasında ne şekilde aktaracağı sorularına cevap olabilecek yapıların geliştirilmesi gerekmektedir. Sertifika yaşam döngüsüne ait fonksiyonlar, tanım itibariyle oldukça geniş bir süreçler bütününde ele alınmıştır. Ancak bu işlemler, her ne kadar sistem içerisindeki süreçler topluluğunun merkezinde yapılandırılacak olsalar dahi, tasarım ve geliştirilebilirlik açısından da, bir o kadar basit ve sistemin bütününe entegre olma zorunluluğunda geliştirilmişlerdir. Bu açıdan ele alındığında, IPv6 tabanlı yeni geliştirilmekte olan bir açık anahtar altyapısı içerisindeki sertifika yaşam döngüsü süreçlerinin tasarlanması, sistemin gelecekte uygulanabilirliği açısından da kritik öneme sahip olacaktır.

Bu çalışma içerisinde yer alan çalışma konularından olan, sertifika yaşam döngüsü süreçlerinin UMU-PKI içerisine entegrasyonuyla ilgili olarak, güncel yönetim protokolleri incelenmiş ve bu protokoller arasında CMC protokolünün, diğer protokollere kıyasla sahip olduğu avantajlar göz önünde bulundurularak, sistem içerisine dahil edilmesi gerekliliği belirtilmiştir. UMU-PKI sisteminin sertifika yönetim mekanizmalarında, CMC protokolüne ait işlevlerin entegrasyonları üzerine tasarlanan bu modeller, protokolün henüz geliştirilmekte olan yeni eklentileriyle birlikte, günümüz teknolojilerindeki güncel yapıları geniş ölçeklerde kapsayabilmesi ve UMU-PKI sisteminin, bu çalışmada detaylı olarak incelenen, sertifika yaşam döngüsündeki tüm süreçlerini içerecek metodlara sahip olmasının yanı sıra, UMU-PKI sisteminin önceki sürümlerinde üzerinde çalışılan bir diğer sertifika yönetim protokolü olan, CMP protokolüne kıyasla, daha hızlı entegre edilebilir olması, uygulama kodunun daha etkin bir şekilde geliştirilebilmesi ve bu sayede yaklaşık %10 oranında daha düşük kod yükü ile sistem entegrasyonunun sağlanabilmesi gibi avantajlarla, sertifika yönetim mekanizmalarının sağlanmasında, CMC protokolünün tercih edilmesi gerekliliği belirtilmiştir. [SSH Communications, 2002]

CMC protokolünün sisteme entegrasyonunda öncelikli olarak, protokol içerisinde yer alan işlev, tanım, metod ve nesneler detaylı olarak incelenerek tanıtılmış, protokol yapısındaki bu birimlerden, UMU-PKI sistemi entegrasyonunda öne çıkacak nesne ve bileşenlerin kapsamlı bir şekilde analizleri gerçekleştirilmiştir. Yapılan analizlerde, protokolün entegrasyon tasarımlarında öneme sahip olacak yapıların içerikleri ve bu nesnelerin özellikleriyle ilgili bilgiler, protokol referans dokümanında belirtilen tanımlar çerçevesinde modellenmiştir. CMC protokolü bünyesinde desteklenen iki farklı talep ve yanıt çiftlerinden, tüm talep ve tüm yanıt çifti, UMU-PKI sistemi içerisinde yer alan sertifika yönetim mekanizmalarındaki talep ve yanıt çifti olarak tercih edilmiştir. Basit talep-yanıt çifti yerine, tüm talep-yanıt çifti tasarımlarının tercih edilmesindeki neden, bu formatta biçimlendirilen taleplerin içeriklerindeki nesneler itibariyle, sistem süreçlerinde, yüksek seviyede fonksiyonalite ve esneklik sağlayabilmesidir. (Bkz. Bölüm 3.1.3.9) Sistemde gerçekleştirilecek taleplerde, PKIData nesnesi ve bu nesnenin sahip olduğu içerik nesne yapılandırması düzenlenerek, uygulama tabanlı tasarım modellerine dahil edilmiştir. Sistem yanıtlarında ise benzer şekilde ResponseBody nesnesi üzerinden tasarımlar modellenmiştir. (Bkz. Bölüm 3.1.3.1)

Talep ve yanıt nesnelerinin yanı sıra sistemdeki süreçler içerisindeki nesnelerin içerik tiplerine yönelik CMC protokolü bünyesinde sağlanan EnvelopedData, SignedData ve Data metodları incelenerek, protokolün sisteme entegrasyonunda sıklıkla tercih edilecek, talep ve yanıtların genel taşınma paketi olarak da kullanılabilecek SignedData nesnesi detaylandırılmıştır. Talep ve yanıt mesajlarının her ikisi için de ortak olarak kullanılan kontrol özelliklerinden öne çıkanlar incelenmiş, bu kontrol özelliklerinden sistem entegrasyonunda faydalanılacak olanlar uygulama tabanlı olarak modellenmiştir.

Protokole ait bu yapıların sistemle bütünleştirilmesinde, öncelikle sistem içerisinde gerçekleşecek süreçlerin operasyonel olarak modellenmesi yapılandırılmış, süreçlerin hangi amaçla, ne şekilde işletileceği, hangi mekanizmalara sahip olacakları ve sistemdeki hangi birimlerle etkileşimde bulunacakları, gerek çalışma içerisinde sunulan modellerle, gerekse bu modellerin sözlü anlatımlarıyla sunulmuştur. Bu süreçler, sertifika yaşam döngüsü kapsamı içerisinde yer alan ve UMU-PKI sisteminin gereksinimleri doğrultusunda üzerinde çalışılan, sertifika talepleri, iptal talepleri, yenileme talepleri, sertifikaların istemciler tarafından sistemden edinilmeleri, iptal listesi talepleri ve taleplerin durumlarının sorgulaması gibi mekanizmalar ele alınarak, bu süreçlerin CMC protokolü tanımı ve bileşenlerince operasyonel olarak sisteme dahil edilmesini sağlayan tasarımlar gerçekleştirilmiştir. Bu tasarımlar, günümüzde yaygın bir şekilde kullanılan, akış diyagramları gösterimiyle mümkün olduğu kadar basit ancak tüm işlevleri sağlayacak bir mantıkta tasarlanmış olup, çalışmanın sonraki aşamalarında detaylandırılarak uygulama temelli entegrasyon modelleri haline getirilmiştir. Operasyonel süreçlerin detaylandırılarak uygulama temelli tasarımlar haline gelmesinde, JAVA'da tanımlı nesne yapıları ve özellikle Java.Security.Cert.X509Certificate, IAIK.X509.RevokedCertificate,Piscis.Pki.Shared.Policy.PolicyManager, Java.Security.Cert.X509CRL sınıflarına ait metod ve bileşenlerden faydalanılmıştır.

UMU-PKI sistemine ait sertifika yaşam döngüsü mekanizmalarında CMC protokolünün uyarlanması çalışması sonucunda, sistem bünyesinde farklı kazanımlar elde edilmiştir. Gerek, günümüz teknolojilerinde uygulanmakta olan sertifika yönetim protokollerine kıyasla CMC protokolünün sağladığı avantajlar, gerekse bu protokolün yapısı itibariyle sistemlere kazandırdığı yeni ve geliştirilebilir eklentiler sayesinde, bu protokolün sertifika yönetim konusunda uygulandığı ortamlardaki yapılara, birçok getirisi ve üstünlüğü belirtilmiştir.

UMU-PKI v6 sistemi içerisindeki sertifika yönetimi mekanizmalarında CMC protokolünün entegrasyonunun tasarlanması sonucunda, elde edilecek sistemsel kazançlar aşağıdaki çizelge içerisinde sunulmuştur. (Bkz. Çizelge 4.1)

Kriter	Açıklama
Eksiksizlik	Günümüz açık anahtar teknolojilerinde en fazla kapsama sahip protokollerden biridir. UMU-PKI için tüm mesaj formatları ve akışları protokol bünyesinde sağlanabilir.
Güncellik	Günümüz güncel sertifika tabanlı teknolojilere ait yaşam döngüsü işlevlerinin tümü protokol bünyesinde öngörülmüştür.
Entegrasyon	Özellikle CMP protokolüne kıyasla çok daha hızlı entegre edilebilir ve uygulama kodu daha etkin bir şekilde geliştirilebilir. [SSH Communications, 2002]
Etkinlik	Protokol bünyesindeki akışlarda yer alan döngüler tek kademeli ya da asgari olarak tasarlanma mecburiyetindedir. (Bkz. Bölüm 3.1.1)
Uygulama Kod Yükü	Yaklaşık %10 oranında daha düşük kod yükü ile entegrasyon sağlanabilir. [SSH Communications, 2002]
İşlem Modelleri	İkili, gelişmiş ikili, üçlü, vb. Haberleşme modellerinin tümünü destekler. [E. Gülaçtı, 2006]
Algoritma Bağımsızlık	Algoritma bağımsızdır
Karmaşıklık	Gerçeklenmesi karmaşık ve zordur [E. Gülaçtı, 2006]
Genişletilebilirlik	Yapısal özellikleri itibariyle kolaylıkla geliştirilebilir ve genişletilebilir bir modeldir.
Arşivleme	Tüm mesajlar kriptografik olarak korunabildiği için mesajların arşivlenmesi oldukça kolay ve mantıklıdır.

Çizelge 4.1 CMC Protokolü Entegrasyonu Kazanımları ve Avantajları

Sertifika yönetimine paralel olarak sistemde sağlanması gereken bir diğer önemli husus olan, sistemdeki sertifikaların doğru bir şekilde yapılandırıldığının ve sertifika yollarının güvenilir bir şekilde oluşturulduğunun doğrulanması işlemine yönelik olarak, güncel uygulamalar araştırılarak incelenmiştir. Bu uygulamalar arasında, yaygın bir şekilde kullanımda olan

sertifika doğrulama protokollerinin başında OCSP protokolü öne çıkmaktadır. Ancak bu çalışma içerisinde, UMU-PKI sisteminin sertifika doğrulama mekanizmalarının sağlanmasına yönelik olarak, OCSP protokolü yerine SCVP protokolüne ait işlevler, sistem mimarisi içerisine dahil edilmiştir. Tasarımlar içerisinde, bu tercihin gerekçeleri olarak önceki bölümlerde detaylandırıldığı gibi, SCVP protokolü ile gelen ve entegrasyon modelleri içerisinde sıklıkla yer verilen güncel eklentilerden yararlanabilmenin yanı sıra, SCVP protokolünün, sertifikaların tüm veri bloğunun yerine, özet verisini kullanarak işlem yapabilmesi sonucunda, sistemdeki mesaj boyutlarının ciddi oranda düşmesi ve paralelinde, sistemdeki talep-yanıt sürelerinin kısalması hususları ön plandadır. [Papapanagiotou, K. et. al, 2005]

SCVP protokolünün sisteme entegrasyonu kapsamındaki çalışma içerisinde, protokol referans dokümanında belirtilen gerekler doğrultusunda, sistemde yer alması gereken doğrulama talepleri, doğrulama yanıtları, politika talepleri ve politika yanıtları süreçleri öngörülerek bu süreçlerin sistemdeki modellemesi gerçekleştirilmiştir. Sistemde bu süreçleri yönlendirecek SCVP tabanlı bir doğrulama servisinin, UMU-PKI sistemine ait yapılar göz önünde bulundurulduğunda, içermesi gereken akış modeli tasarlanmış olup (Bkz. Bölüm 3.3.2.1) oluşturulan modelden yola çıkılarak protokolün entegrasyonuna yönelik çözüm önerilmiştir. Tasarlanan bu doğrulama servisi bazında sistem içerisindeki sertifika doğrulama mekanizmaları, uygulama tabanlı olarak detaylandırılmış ve entegrasyon modelleri haline getirilmiştir. UMU-PKI v6 sisteminin SCVP protokolünün öngördüğü mekanizmalarla bütünleştirilmesi sonucunda, özellikle sertifika doğrulama çatısı altında sistem güvenliği ve güvenilirliği arttırılarak, daha güçlü ve doğru bir yapı sağlanmıştır.

SCVP yanıtlayıcı sunumcu mimarilerinin UMU-PKI sistemi entegrasyonuyla, sistematik olarak edinilen kazançlar genel olarak aşağıdaki çizelgede sunulmuştur.

Kriter	Açıklama
Yük Paylaşımı	Doğrulama işlevleri sunumcular tarafından gerçekleştirildiğinde, sistemdeki istemcilerin yükü hafifler ve işlem yükü fazla ortamlar içerisinde paylaşımcı ya da yedekli tasarlanan sunumcu ağlarıyla doğrulama mekanizmalarında performans artar.
Performans	SCVP protokolünün, sertifikaların tüm veri bloğu yerine özet verisini kullanarak işlem yapabilmesi sonucunda mesaj boyutları düşer ve sistemdeki talep-yanıt süreleri kısalır. [K. Papapanagiotou et. al, 2005]
Devamlılık	İhtiyaç duyulan ortamlarda sunumcu sayılarının arttırılması ile sertifika mekanizmalarının sürekliliği ve sistem güvenilirliği artar.
Ölçeklenebilirlik	İstemci sunumcu mimarisinde, organizasyon yapısının konfigurasyonuna bağlı olarak, sistem yükü, istemci sayısı ya da diğer şartlar düşünülerek sertifika doğrulama mekanizmalarına ait birimler daha esnek bir sistem tasarımı içerisinde modellenebilir.
Güvenlik	Talepler, makamlar yerine direk olarak sunumculara iletilir. Bu sayede makamlar, bağlantısız durumda dış tehditlere karşı daha rahat muhafaza edilirler. Sistem, yedekli ve performans ölçekli bir sunumcu ağıyla yapılandırıldığında, dış tehditlere yönelik daha güçlü bir ortam sağlar. Güvenlik, birden fazla sayıda noktaya bağlı olduğundan, sistem başarısızlık riski de asgariye iner.
İstemci Entegrasyonu	Sertifika doğrulama işlevleri, doğrulama talebinin istemci tarafından sunumcuya iletilmesiyle sunumcu tarafından sağlanacağından, OCSP protokolüne kıyasla daha düşük istemci konfigurasyonları ile entegrasyon sağlanabilir.
Merkezi Güven ve Politika Yönetimi	Sertifika doğrulama işlevleri, organizasyon yapısı içerisindeki sunumcular tarafından gerçekleştirildiğinden, sistemdeki güven mekanizmalarının ve doğrulama temelli politikaların yönetimi de merkezileşir.

Çizelge 4.2 SCVP Protokolü Entegrasyonu Kazanımları ve Avantajları

Çizelge 4.1. ve 4.2. ’de sunulmuş olan CMC ve SCVP protokollerinin, UMU-PKI sistemi içerisinde, sertifika yaşam döngüsü ve sertifika doğrulama mekanizmaları olarak tasarlanmalarının sağladığı sistemsel ve süreç tabanlı kazançlarla UMU-PKI v6 sistemi, bu servislerin sağlanmadığı ortama kıyasla, daha güvenli, güçlü ve güvenilir bir yapıya sahip olmuştur. Özellikle sertifika yaşam döngüsü mekanizmalarının yönetiminin, günümüz güncel teknolojileri içerisindeki yapıları destekleyebilen mekanizmalarla tasarlanmış olmasının,

UMU-PKI v6 sisteminin gelecek entegrasyonları öngörülerek edinilecek kazanç faktörü ve sistemin geleceğinin güçlü ve güvenilir bir yapıda olmasında, kritik değer taşıyan etkenlerin başında gelmektedir. Bu açıdan UMU-PKI v6 sisteminin CMC protokolü ile bütünleştirilmesine yönelik bu çalışma içerisinde gerçekleştirilmiş tasarımlar ve analizlerin, özellikle gelecek IPv6 tabanlı yeni nesil açık anahtar sistemleri içerisindeki rolü, oldukça önemlidir.

5. ÖNERİLER

UMU-PKI konusunda özellikle Euro6IX ağı çerçevesinde ve Euro6IX konsorsiyumu sorumluluğunda süregelen birçok çalışma yürütülmeye devam edilmektedir. Bu çalışmalardan bir kısmı sonlandırılmaya yakınken, bir takım çalışmalar, tanım bazında raporlarda yer almaktayken, bu konuları içeren çalışmalar henüz tam anlamıyla gerçekleşmemiştir. Gerek bu konsorsiyum tarafından yürütülen çalışmalar, gerekse içinde bulunulan çalışma içerisinde incelenen konular göz önüne alınarak, bu konuyla ilgilenen araştırmacılara yönelik gelecekte üzerinde çalışmalar yürütülebilecek bir takım öneriler, bu bölümde konu başlıklarıyla sunulacaktır.

IPv6 tabanlı JAVA temelli bir açık anahtar altyapısı öngören UMU-PKI sisteminin, güvenli sanal özel ağlar içerisinde uygulanması ya da kimlik kanıtlama, yetkilendirme, hesaplama servisleri (AAA) olarak adlandırılan mekanizmaların, uygulama bazlı entegrasyonları üzerine çalışmalar geliştirilebilir. Benzer şekilde bu çalışma içerisinde de bilgiler sunularak incelenen ve UMU-PKI desteğiyle ilgili teorik öngörüler ve analizler gerçekleştirilen, çapraz sertifikaların sistem içerisinde uygulanabilmesi konusuyla ilgili olarak, farklı sertifika makamları arasında gerçekleşen çapraz sertifikaların, UMU-PKI altyapısını kullanarak uygulama tabanlı sistemler içerisine entegre edilmesi çalışmaları da, gelecekte üzerinde çalışmalar geliştirilebilecek konular arasında yer almaktadır.

Organizasyonel sistemler içerisinde, dağıtık sertifika deposu olarak uygulanması planlanan DNSsec yapısının uygulamalara entegrasyonu ve yine aynı altyapı içerisinde uygulanması tasarlanan nitelik sertifikalarının organizasyon içerisindeki kullanıcılara ya da proseslere farklı rollerin atanması yönündeki mekanizmaların yönetimi konularında da farklı yöntemler altında geliştirilebilecek uygulama tabanlı çözümler, gelecekte bu konuyla ilgilenen kişilerce öncelikli olarak üzerinde çalışmalar geliştirilebilecek hususlardır.

KAYNAKLAR

[1] Adams, C. ve Farrell, S. "Internet X.509 Public Key Infrastructure Certificate Management Protocols", Internet Engineering Task Force, Jul. 2000. [Adams, C. ve Farrell, S. 2000]

[2] Adams, C., Cain, P., Pinkas, D. ve Zuccherato, R. "Internet X.509 Public Key Infrastructure Time Stamp Protocols (TSP)", Internet Engineering Task Force, Oct. 2000. [Adams, C. et.al, 2000]

[3] Adams, C. ve Farrell, S. "Internet X.509 Public Key Infrastructure Certificate Management Protocols", Network Working Group, Request for Comments: 2510, 1999. [Adams, C. ve Farrell, S. 1999]

[4] Adams, C. ve Myers, M. "Internet X.509 Public Key Infrastructure Certificate Management Message Formats", Internet Engineering Task Force, Jul. 1998. [Adams, C. ve Myers, M. 1998]

[5] Boeyen, S., Howes, T. ve Richard, P. "Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2", Internet Engineering Task Force, Sep. 1998. [Boeyen, S. et.al, 1998]

[6] Clarke, D. "SPKI/SDSI HTTP Server/Certificate Chain Discovery in SPKI/SDSI", Massachusetts Institute of Technology, 2001. [Clarke, D. 2001]

[7] Deering, S. ve Hinden, R. "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998. [Deering, S. ve Hinden, R. 1998]

[8] Dzambasow, Y., Cooper, M., Hesse, P., Joseph, S. ve Nicholas, R. "Internet X.509 Public Key Infrastructure: Certification Path Building", RFC 4158, September 2005. [Dzambasow, Y. et. al, 2005]

[9] Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B. ve Ylonen, T. "SPKI Certificate Theory", RFC 2693, September 1999. [Ellison, C. et.al, 1999]

[10] Gomez, A.F. et al., "The UMU-PKI Design and Architecture, tech. report TR-DIIC 1/2002, Departamento de Ingenieria de la Informacion y las Comunicaciones", Univ. of Murcia, 2002. [Gomez, A.F. et al, 2002]

[11] Gomez, A.F., Martinez, G., Canovas, O. "New Security Services Based on PKI", Future Generation Computer Systems, vol. 19, Elsevier Science, 2003, pp. 251-262. [Martinez, G. et.al, 2003]

[12] Housley, R., Ford, W., Polk, T. ve Solo, D. "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", Internet Engineering Task Force, Sep. 1998. [Housley, R. et. al, 1998]

[13] Housley, R., Ford, W., Polk, W. ve Solo, D. "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002. [Housley, R. et. al, 2002]

[14] Huitema, C. "IPv6: The New Internet Protocol", Prentice-Hall:Upper Saddle River, NJ, 247 pp, 1998. [Huitema, C. 1998]

[15] Liu, X., Madson, C., McGrew, D., Nourse, A. "Cisco Systems Simple Certificate Enrollment Protocol (SCEP)", February 4, 2000. [Liu, X. et. al, 2000]

[16] Kaliski, B. "PKCS #10: Certification Request Syntax Version 1.5", RFC 2314, March 1998. [Kaliski, B. 1998]

[17] Kapoor, A. ve Tschalar, R. "A String Representation of General Name", Internet Engineering Task Force, Nov. 1999. [Kapoor, A. ve Tschalar, R. 1999]

[18] Malpani, A., Turner, S. "Simple Certificate Validation Protocol", Internet Engineering Task Force, Jul. 2000 [Malpani, A. ve Turner, S. 2000]

[19] Myers, M., Liu, X., Schaad, J. ve Weinstein, J. "Certificate Management Messages over CMS", Internet Engineering Task Force, no. 2797, Apr. 2000. [Myers, M ve Liu, X. et. al, 2000]

[20] Myers, M., Ankney, R. ve Adams, C. "Online Certificate Status Protocol, version 2", Internet Engineering Task Force, Sep. 2000. [Myers, M. et.al, 2000]

[21] Papapanagiotou, K., Markantonakis, C., Zhang, Q., Sirett, W.G., Mayes, K. "On the Performance of Certificate Revocation Protocols Based on a Java Card Certificate Client Implementation" SEC 2005, 551-564 [Papapanagiotou, K. et. al, 2005]

[22] Pfleeger, C. ve Pfleeger, S. "Security in Computing", New Jersey: Pearson Education, Inc 2003 [Pfleeger, C. ve Pfleeger, S. 2003]

[23] Prafullchandra, H., Fox, B., Liu, X., Myers, M. ve Weinstein, J. "Certificate Request Syntax", Internet Engineering Task Force, Nov. 1997. [Prafullchandra, H. et. al, 1997]

[24] Ramsdell, B. "S/MIME Version 3 Certificate Handling", Internet Engineering Task Force, Apr. 1999. [Ramsdell, B. 1999]

[25] Schneider, B. "Secret & Lies", Indianapolis: Wiley Publishing Inc. 2004 [Schneider, B. 2004]

[26] Skarmeta, F., Martínez, G., Cánovas, O. "UMU-PKI v6 Description Guide v6.3.4 (draft-v0.1)", December 2002 [Skarmeta, F. et.al, 2002]

[27] SSH Communications Security Discussion "CMP and CMC Comparison", May 2002 Helsinki, Finland [SSH Communications, 2002]

[28] Thompson, J. "6Wind Routers Integrates IPv4 With IPv6", August 14, 2002. [Thompson, J. 2002]

EKLER

EK 1 : X.509 Sertifika Alanları

Bu bölümde RFC 2459'da tanımlanmış X.509 v3 sertifika formatına ait yapıda yer alan bileşenler açıklanmıştır.

- Sürüm (Version) : Sertifikaya ait sürüm numarasını içeren alandır. Tamsayı değerler içerir. X.509 v3 sürümüne ait sertifikaların sürüm alanlarındaki değer 2 olarak gözükür.
- Sertifika Seri Numarası (Certificate Serial Number) : İşlevselleştirici Sertifika Makamı tarafından sertifikaya atanan seri numarasını içeren alandır. Sertifikalar, içerdikleri seri numaralarıyla ayırt edilebilirler. Herhangi bir tamsayı değer içerebilir. Sertifika Makamları genellikle yayınlamış oldukları sertifikalara sıralı seri numarası atarlar.
- İmzalama Algoritması Belirteci (Signature Algorithm ID) : Sertifika Makamı tarafından sertifikanın imzalanması işleminde uygulanacak olan algoritmayı belirten alandır. Bu algoritma RSA, DSA ya da farklı bir algoritma olabilir.
- İşlevselleştirici (Issuer) : Sertifikayı imzalayan ve yayınlayan Sertifika Makamı'na ait alandır. İşlevselleştirici adı alanı, sertifika yayınlayıcısına ait DN tanımlamasını içerir. Örnek bir yayınlayıcı adı alanı tanımlaması : "c=TR, o=UEKAE, ou=TUBITAK, I=Istanbul, cn=UEKAE Kamu SM" olarak belirtilebilir.
- Geçerlilik Periyodu (Validity Period) : Sertifikanın sistemde geçerliliğini sürdüreceği zaman aralığını belirten alandır. Sertifikanın geçerli olduğu ve son geçerli olacağı tarih bilgilerini içerir. Bu alanda tanımlanan tarih bilgileri, saniye birimini de içerecek şekilde gösterilirler. Sertifika tabanlı işlemler gerçekleştiren herhangi bir uygulama, bu alandaki verileri değerlendirerek sertifikaların geçerli olup olmadıklarını tespit eder. Örnek bir alan değeri : "20.05.2007 23:58:00 Z" şeklinde tanımlanabilir.
- Özne (Subject) : Bu alan, açık anahtarı sertifikalanacak olan birime ait tanımlamayı içerir. Özne adı sistemde yayınlanan her birim için tekil olmalı ancak her bir özne için sistemde birden fazla sertifika yayınlanabilmelidir. Özne alanı, sertifika sahibine ait DN ya da SMTP adresi bilgisini içerir. Örnek bir özne alanı tanımlaması : "c=TR, o=UEKAE, ou1=Bilgi Sistemleri, I=Istanbul, cn=Burak Çalışkan" şeklinde belirtilebilir.
- Özne Açık Anahtar Bilgisi (Subject Public Key Information) : Bu alan anahtara ait açık anahtar bilgisini ve algoritma belirteci gibi anahtarın kullanılacağı ilgili diğer verileri içerir. Bu alan verisi hexa-desimal formatta yapılandırılır.
- İşlevselleştirici Tekil Belirteci (Issuer Unique ID) : Opsiyonel bir alan olup, yayınlayıcı adlarının zaman içerisinde birden fazla kez kullanımını sağlar.

- Özne Tekil Belirteci (Subject Unique ID) : Opsiyonel bir alan olup, özne adlarının zaman içerisinde birden fazla kez kullanımını sağlar.
- Uzantılar (Extensions) : Sadece X.509 v3 sürümüne ait bir alan olup, sonraki bölümde belirtilecek uzantı bilgilerini içeren alandır.
- Sertifika Makamı Sayısal İmzası (Certification Authority's Digital Signature) : Bu alanda yer alan Sertifika Makamı'na ait sayısal imza, sertifikaya ait imzalama algoritması belirteci alanındaki değerle, gizli anahtar kullanılarak hesaplanan veridir. Sertifika Makamı Sayısal İmzası alanı, sertifikanın bütünlüğünü garantileyen bir alandır. Sertifika Makamı'na ait açık anahtar, sertifika kullanıcılarına dağıtılarak kullanıcıların herhangi bir açık anahtar işlemi yapmalarından önce, ilgili sertifikanın geçerliliğini, Sertifika Makamı'na ait açık anahtar ve Sertifika Makamı Sayısal İmzası verilerini kullanarak kontrol edebilmeleri sağlanır.

EK 2 : X.509 Sertifika Uzantıları

Uzantılar, açık anahtar kullanıcılarına veya açık anahtarlara farklı ek özellik bilgilerini yükleyebilmek için tasarlanmış bilgi alanlarıdır. Genellikle uzantı alanları, sertifikalama hiyerarşisi, sertifika iptal bilgileri, politika bilgileri ve benzer içeriklere sahiptirler. Ayrıca uzantı alanları konfigure edilerek daha farklı bilgilerin de iletilmesi sağlanabilir. Bir sertifika içerisindeki her uzantı alanı, bir kritiklik belirtecine sahiptir. Kritiklik belirteci içeriği "critical" ya da "uncritical" olarak tanımlanır. Eğer uygulama içerisindeki bir uzantı alanı "uncritical" olarak işaretlenmişse, bu durumda, ilgili sertifika uygulamasının, bu uzantı alanını algılayamadığında veya içeriğini okuyamadığında, yoksayıp yoksayamayacağını gösterir. RFC 2459'da tanımlı uzantılar aşağıdaki bilgilerde açıklanmıştır :

- Makam Anahtar Belirteci (Authority Key Identifier) : Sertifika üzerindeki sayısal imzanın doğrulanması işleminde kullanılacak olan Sertifika Makamı'na ait tekil açık anahtar belirteci bilgisini içeren alandır. Aynı Sertifika Makamı'na atanmış olan birden çok anahtar arasından ayırt etme işleminde ve kendinden-imzalı Sertifika Makamı sertifikalarında da bu alandan yararlanılmaktadır. Sertifika tabanlı uygulamalar, sistemdeki birimlere ait sertifikaları geçerleyebilmek için, sistemin bağlı olduğu etki alanı içerisindeki tüm Sertifika Makamlarına ait anahtarları içermek durumundadır. Sertifika Makamlarından birisi, birden fazla anahtara sahipse, uygulamaların bu anahtarların tümüne sahip olması gerekmektedir. Birden fazla anahtara sahip Sertifika Makamı'nın anahtarlarının belirlenebilmesinde rol oynar.
- Özne Anahtar Belirteci (Subject Key Identifier) : Sertifika içerisinde yer alan açık anahtara ait tekil belirteç bilgisini içeren alandır. Aynı sertifika sahibine atanmış birden çok anahtar arasından ayırt etme işleminde yararlanılır.
- Anahtar Kullanımı (Key Usage) : Bit string veri tipinde bilgi içeren bir alandır. Sertifika içerisindeki açık anahtar tarafından desteklenen fonksiyonlar ve servisleri belirtmek ya da bazılarını engellemek amacıyla kullanılır. Örnek değerler : "(1) sayısal imza, (2) inkar edememe, (5) sertifika iptal listesi imzalama, (7) sertifika imzası" olarak belirtilebilir.
- Genişletilmiş Anahtar Kullanımı (Extended Key Usage) : Sertifika içerisindeki açık anahtarın, belirli bir işlemi gerçekleştirmesine yönelik olarak bir ya da birden çok nesne belirteci ile tanımlanmış bu alandan yararlanılır. Örnek değerler : "(1) TLS sunumcu yetkilendirmesi, (4) e-posta koruması" olarak belirtilebilir.
- İptal Listesi Dağıtım Noktası (CRL Distribution Point) : İlgili sertifikaya ait iptal bilgisinin bulunduğu CRL paylaşımını belirten alandır. Herhangi bir X.500 ya da LDAP dizin sunumcusuna ait adres bilgisini içerebilir.

- Gizli Anahtar Kullanım Periyodu (Private Key Usage Period) : Sertifika içerisindeki açık anahtarla ilişkili gizli anahtarın, sistemde kullanılabileceği zaman aralığını belirten alandır.
- Sertifika Politikaları (Certificate Policies) : Sertifikanın yayınlanması ve diğer fonksiyonel kullanımlarında uygulayacağı politika bilgilerini belirten ve içeriğinde bir ya da daha fazla politika nesne belirtecini içeren alandır.
- Politika Eşleştirmeleri (Policy Mappings) : İki farklı Sertifika Makamı etki alanı arasındaki eş sertifika politikalarını belirten alandır. Çapraz sertifikaların oluşturulmasında ve sertifika yolu doğrulama işlevlerinde yararlanılır. Bu alan, sadece Sertifika Makamlarına ait sertifikalarda bulunmaktadır.
- Özne Alternatif Adı (Subject Alternative Name) : Sertifikanın öznesine ait alternatif isim tanımlamasını içeren alandır. Bu alan içerisinde özneye ait e-posta, ip adresi ya da URI adresi bilgileri saklanabilir.
- İşlevselleştirici Alternatif Adı (Issuer Alternative Name) : Sertifikanın yayınlayıcısına ait alternatif isim tanımlamasını içeren alandır. Bu alan içerisinde yayınlayıcıya ait e-posta, ip adresi ya da URI adresi bilgileri saklanabilir.
- Temel Sınırlamalar (Basic Constraints) : Bu alan, sertifikanın bir Sertifika Makamı sertifikası olup olmadığını belirten bilgiyi içeren alandır. Sadece Sertifika Makamlarına ait sertifikalarda bu alan yer alır. Eğer bir Sertifika Makamı, sadece kendi kullanımı için bir sertifika oluşturmuşsa, bu sertifika " kendinden-imzalı Sertifika Makamı sertifikası " olarak adlandırılır. Yeni bir Sertifika Makamı kurulduğunda, bu makam öncelikli olarak kendine ait bir kendinden imzalı sertifika yaratarak uç birimlere ait sertifikaları imzalayabilmeyi sağlar. Ayrıca Sertifika Makamları, ilgili açık anahtar altyapısı içerisinde farklı mimariler oluşturabilmek için sistemdeki diğer Sertifika Makamlarına ait sertifikaları da oluşturabilirler.
- Yol Uzunluk Sınırlaması (Path Length Constraint) : Sadece Sertifika Makamlarına ait sertifikalarda yer alan bir alan olup, sertifika yolunun oluşturulmasında en fazla kaç adet makam bilgisinin yolda yer alabileceğini belirtir.
- İsim Sınırlamaları (Name Constraints) : Sertifika yolu yapılandırmalarında, "permitted subtrees" ya da / ve "excluded subtrees" tanımlamaları doğrultusunda, gerekli ve / ya da çıkartılmış alt isimlerin kullanımını belirten alandır.
- Politika Sınırlamaları (Policy Constraints) : Bu alan, gerekli politika belirteçlerini ya da yasaklanmış politika eşleştirmelerini "require explicit policy" ve "inhibit policy-mapping" bilgileri doğrultusunda tanımlayan alandır.

EK 3 : Sertifika İptal Listeleri

Sertifika iptal listeleri, iptal edilmiş sertifikalara ait liste bilgilerini ve kullanıcılar için gerekli diğer bilgileri içeren veri yapılarıdır. X.509 sertifika iptal listeleri için "sürüm1" ve "sürüm2" adıyla iki farklı sürüm tanımlanmıştır. Sertifikalarda olduğu gibi iptal listelerinde de, sonraki sürümlerde bazı uzantı bilgileri eklenerek açık anahtar altyapısı alanındaki gelişimler doğrultusunda yeni gereksinimlerin karşılanması sağlanmıştır. RFC 2459'da tanımlanmış olan iptal listelerinin yapısal olarak içeriklerinde bulunan bileşenler aşağıda belirtilmiştir.

- Sürüm (Version) : Sertifika iptal listesine ait sürüm bilgisini içeren alandır. Eğer liste içerisinde herhangi bir uzantı yer alıyorsa sürüm 2, aksi halde sürüm belirtilmez.
- İmza (Signature) : İptal listesinin imzalanmasında kullanılacak algoritmaya ait algoritma belirteci bilgisini içeren alandır.
- İşlevselleştirici (Issuer) : Sertifika iptal listesini yayınlayan ve imzalayan birime ait tekil isim bilgisini içeren alandır.
- Şimdiki Güncelleme (This Update) : Sertifika iptal listesinin yayınlandığı tarih ve zaman bilgilerini belirten alandır.
- Sonraki Güncelleme (Next Update) : Sertifika iptal listesinin geçerliliğini yitireceği tarih ve zaman bilgilerini belirten alandır.
- İptal Edilmiş Sertifikalar Listesi (List of Revoked Certificates) : Bu alan, sistemde geçersiz duruma gelmiş sertifikalara ait listeyi, seri numaraları halinde belirtir. Aynı zamanda bu alan içerisinde, sertifikaların iptal edilme tarih ve zamanları opsiyonel olarak belirtilebilir.
- Uzantılar (Extensions) : Sadece sürüm 2 iptal listelerinde bulunan bu alan, iptal listelerine ait birtakım ek bilgilerin tutulabilmelerine olanak sağlar.

EK 4 : Açık Anahtar Altyapıları Karşılaştırma Tablosu

X.509	SM Özellikleri	Genel X.509 makamları hiyerarşisine uygun şekilde makamlar oluşturulur
	Güven Modeli	Hiyerarşik güven modeli
	İmzalar	Her sertifika, işlevselleştiricisine ait bir adet imza taşır
	Sertifika İptalleri	Sertifika iptal listeleri (CRL) kullanılır
	İsim Alanı	Global
	Sertifika Tipleri	İsim Sertifikaları
	İsim – Anahtar İlişkisi	Tekil bir fonksiyondur. Her isme ait bir anahtar yer alır. (Her kullanıcının bir çift anahtara sahip olduğu varsayılmıştır)
OpenPGP	SM Özellikleri	Eşitlikçi yapıdadır. Her anahtar sertifika işlevselleştirebilir
	Güven Modeli	Dosya tabanlı açık anahtar altyapısı
	İmzalar	Her sertifika birden fazla imza içerebilir. İlk imza sertifikanın işlevselleştiricisine aittir
	Sertifika İptalleri	PGP anahtar sunumcularına, anahtar iptal sertifikası iletilir
	İsim Alanı	Global
	Sertifika Tipleri	İsim Sertifikaları
	İsim – Anahtar İlişkisi	Tekil bir fonksiyondur. Her isme ait bir anahtar yer alır. (Her kullanıcının bir çift anahtara sahip olduğu varsayılmıştır)
SPKI	SM Özellikleri	Eşitlikçi yapıdadır. Her anahtar sertifika işlevselleştirebilir.
	Güven Modeli	Gardiyan yapısı temelli bir güven modeli mevcut olup, talepçi, kendi açık anahtarı için gardiyandan yetkilendirme ister
	İmzalar	Her sertifika, işlevselleştiricisine ait bir adet imza taşır
	Sertifika İptalleri	Kısa geçerlilik süreli avukatlar ve COH uygulanır
	İsim Alanı	Lokal
	Sertifika Tipleri	İsim ve Yetkilendirme Sertifikaları
	İsim – Anahtar İlişkisi	Çoğul bir fonksiyondur. Her isim, hiçbir anahtara ya da bir veya daha fazla anahtara bağlı olabilir

Çizelge Ek 4 Açık anahtar altyapıları karşılaştırma tablosu [Clarke, D. 2001]

ÖZGEÇMİŞ BURAK ÇALIŞKAN

KİŞİSEL BİLGİLER

Doğum tarihi / yeri	05.09.1978 / ANKARA
Medeni Durumu	Evli

EĞİTİM

Lise	1989-1993	Tarsus Amerikan Koleji
Lise	1993-1996	İzmir Y Fen Lisesi
Lisans	1996-2000	İstanbul Teknik Üniversitesi
		Elektronik ve Haberleşme Mühendisliği

SERTİFİKA PROGRAMLARI

2006	Project Management Certification Program (PMI)
2004	Oracle Database Administrator (Oracle DBA)
2000	Microsoft Certified Systems Engineer (MCSE)
2000	Microsoft Certified Professional (MCP)

PROFESYONEL İŞ DENEYİMLERİ

2002-Devam	TÜBİTAK Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü
	Uzman – Bilgi Sistemleri
2000-2002	ON-NET A.Ş Sistem Mühendisi – Analist Yazılımcı
1999-2000	SHOW TV Ağ Yöneticisi
1998-1999	BP-ATAŞ Sistem Destek Uzmanı