İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

KONU TABANLI ETMEN GÖÇÜ İÇİN GÜVENLİK YAPISI

YÜKSEK LİSANS TEZİ

Ahmet Ali KARZAN

İleri Teknolojiler Anabilim Dalı

Bilgisayar Bilimleri Programı

HAZİRAN 2013

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

KONU TABANLI ETMEN GÖÇÜ İÇİN GÜVENLİK YAPISI

YÜKSEK LİSANS TEZİ

Ahmet Ali KARZAN
(704101002)

İleri Teknolojiler Anabilim Dalı

Bilgisayar Bilimleri Programı

Tez Danışmanı: Nadia ERDOĞAN

HAZİRAN 2013

İTÜ, Bilişim Enstitüsü'nün 704101002 numaralı Yüksek Lisans Öğrencisi **Ahmet Ali KARZAN**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"KONU TABANLI ETMEN GÖÇÜ İÇİN GÜVENLİK YAPISI"** başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Nadia ERDOĞAN** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Prof. Dr. Çoşkun SÖNMEZ** Yıldız Teknik Üniversitesi	.....
	Doç. Dr. Turgay ALTILAR İstanbul Teknik Üniversitesi	.....

Teslim Tarihi : **03 Mayıs 2013**
Savunma Tarihi : **06 Haziran 2013**

Aileme,

ÖNSÖZ

Tezimi hazırlamamda benden yardımlarını esirgemeyen, her zaman yol gösterici ve teşvik edici olan tez danışmanım Prof. Dr. Nadia ERDOĞAN'a teşekkürü bir borç bilirim.

Haziran 2013

Ahmet Ali Karzan
Mühendis

İÇİNDEKİLER

Sayfa

KISALTMALAR

ACL : Agent Communication Language
AES : Advanced Encryption Standart
CA : Certification Authority
CBC : Cipher Block Chaining
CTR : Counter
DEA : Data Encryption Algorithm
DES : Data Encryption Standart
ECB : Electronic Code Book
JCE : JAVA Cryptography Extension
LDAP : Lightweight Directory Access Protocol
FIPA : Foundation of Intelligent Physical Agents
MASIF : Mobile Agent System Interoperability Facility
NIST : National Institue of Standarts and Technology
OMG : Object Management Group
RSA : Rivest Shamir Adelman
SHA : Secure Hash Algorithm
SSL : Secure Sockets Layer
TLS : Transport Layer Security
WEP : Wired Equivalent Privacy
WPA : Wi-fi Protected Access

ŞEKİL LİSTESİ

Sayfa

KONU TABANLI ETMEN GÖÇÜ İÇİN GÜVENLİK YAPISI

ÖZET

Etmenler içinde bulundukları ortamı algılayabilen, kullanıcıları adına otonom, proaktif, işbirlikçi bir şekilde iş yapabilen yazılım varlıklarıdır. Etmenlerin belirli bir amaçları, hedefleri ve inançları vardır ve ortamlarından aldıkları bilgiye göre hedeflerini ve inançlarını değiştirebilir veya güncelleyebililer bunun için hiçbir dış müdahaleye ihtiyaçları yoktur. Etmenler otonom olmalarından ve proaktifliklerinden dolayı yine hiçbir dış müdahale olmadan çevrelerinden topladıları veri ve bilgi üzerine bir işlem yapmaya karar vererek aksiyona geçebilirler. Ayrıca, etmeler sosyal varlıklardır inaçları doğrultusunda hedefleri neyse bu hedefleri gerçekleştirmek için her yolu denemeye çaılışrlar ve tek başlarına yapamayacakları işlerin gerçeklenebilmesi için çevrelerinde bulunan diğer etmenlerle işbiriliği yaparlar. Tüm bunların yanında zorunlu olmamakla birlikte haraketlilik te bir etmen özelliğidir. Hareketli etmenler bilgisayar ağları üzerinde hareket ederek farklı konaklara göç ederek hedeflerini gittikleri yerlerde gerçekleştirmeye çalışırlar.

Sabit veya hareketli etemenlerin işbirliği içerisinde bir arada çalışmalarıyla çoklu etmen sistemleri oluşmaktadır. Çoklu etmen sistemleri etmen özelliklerinin sisteme kazandırdığı yetenekler ile merkezi olmayan ve yüksek oranda dağıtık, özerk, proaktif yazılım sistemlerinin gerçeklenebilmesi açısından oldukça önemlidirler. Özellikle hareketli etmenler üzerinden çalışan etmen sistemleri bilgisayar ağları üzerinde yüksek verimlilik sağlarlar. Bir işin veya hedefin gerçekleşebilmesi için haberleşmek durumunda olan yazlım varlıkları bir birlerine ağ üzerinden bir belirli protokoller kullanarak büyük hacimde veri transferi yapabilmektedirler ve bu işlemler yüksek mikatrda ağ kaynağının alı konulmasını gerektirebilmektedir. Oysa ki hareketli etmenler bu varlıklar arası bahsedilen veri akışını ortadan kaldırarak ağ üzerinde sadece kendileri hareket ederek verinin kaynağına göç edip gerekli işlemlerin bu kaynak üzerinde işlenmesini sağlayarak ağ kaynaklarının kullanımını en aza indirerek bu kaynakların başka işler için kullanılabilmesini sağlayıp ağ verimliliğini arttırlar. Hareketli etmenlerin göç etme yöntemlerine yeni bir yaklaşım olan yayınla/abone ol tekniği ile konu tabanlı göç modeli çoklu etmen sistemlerinin yukarda sayılan özelliklerini bir adım daha ileri taşıyarak göç edilecek konağın adresi bilinmeden yani daha önceden etkileşime geçecek olan varlıklar arsında bir tanışıklık zorunluluğu olmadan etmenlerin özerklik ve prokatifliklerini ön plana çıkaran bu paradigma kullanılarak gerçeklenecek olan sistemler için dinamik ağ topolojisine yüksek esneklik katarak büyük ölçüde ölçeklenebilirkik getirmektedir. Tüm bu özellikleri nedeniyle çoklu etmen sistemleri süreç kontrolünden imalata, imalattan lojistiğe, bilgi ve ağ yönetimine pek geniş bir uygulama alanına sahiptir. Bu alanların yanında internet teknolojilerindeki gelişmeler konuya yapısal uygunluğu nedeni ile özellikle elektronik ticaret alanında çoklu etmen sistemlerinin kullanılmasının gerekliliğini ortaya çıkarmıştır. Tüm uygulama alanlarında özellikle elektronik ticaret alanında çoklu etmen sistemleri kullanarak uygulama geliştirmek ve bu uygulamaları kullanıma sunmak için ortamda ciddi bir güvenlik sağlanması gerekmeketedir.

Güveliğinden emin olunmayan bir ticaret uygulamasının ortaya çıkaracağı zararlar düşünüldüğünde böyle bir uygulamanın gerçek hayatta kullanılması mümkün değildir.

Gerekli güvenlik önlemleri alınmadığı takdirde başta maddi zararlar olmak üzere, itibar kaybı, gizli bilginin sızdırılması ve hizmet sağlayıcıların hizmet kesintileri yaşamaları gibi pek çok sorunla karşılaşılabilmektedir. Çoklu etmen sistemlerinin güvenliğinin sağlanması günümüzde yaygın oalrak kullanılmakta olan dağıtık nesne uygulamalarınn, temel haberleşme ve bilgisayar sistemlerinin güvenliğinin sağlanmasından çok ta farklı değildir. Bu sistemlerin güvenliğini sağlmak için kullanılan tekniklerin çoklu etmen sistemlerine uyarlanması en pratik çözüm olarak karşımıza çıkmaktadır. En başta güven ortamınn sağlanması amacıyla kimlik doğrulama işleminin yapılması kaçınılmazdır. Çünkü bir hizmet talep eden veya bir hizmet vereceğini idda eden etmenin gerçekten de olduğunu idda ettiği etmen olması veya adına hareket ettiği kullanıcının gerçekten de o kullanıcı olması gerekmektedir. Eğer bir etmen olmadığı bir etmen gibi davranırsa diğer etmenlere veya konaklara hizmet vermek yerine veya hizmet sağlamanın yanında bilgi sızdırmak, işlem gücünün sarf edilmesi gibi zararlı işler de yapabilir. Fakat kimlik doğrulama işlemi üst düzeyde yeterli güvenilirlik ortaya koyamayan etmenlerden gelecek olan mesajların kabul edilmemsiyele ortaya çıkacak zararların önün geçilmesinde önemli olmaktadır. Haberleşen etmenlerin gönderdikleri mesajların veya göç eden hareketli etmenlerin bütünlüklerinin kontrol edilmesi de güvenilir bir kaynaktan gelen bir verinin yolda zararlı varlıklar tarafından değiştirilip değiştirilmediğinin saptanmasını sağlayarak zararlı faaliyetlerin önüne geçmek için kullanılması gereken bir önlemdir. Gizli verilerin sızdırlmasını engellmek için verilerin şifrelenmesi gerekmektedir. Çünkü ortam dinlemesi gibi pasif ataklarla gizli veri elde edinilse bile şifreli olduğu ve ilgili gizli anahtara saldırgan tarafından sahip olunmadığı için verinin gizliliği korunmuş olur.

Bu çalışmada yayınla/abonel modeli ile konu tabanlı olarak göç eden etmen sistemlerinin güvenliği sağlanmaya çalışılmıştır. Bunun için yukarda sayılan genel güvenlik gereksinimlerini karşılayan bir mimari ortaya konulmaktadır. Hareketli etmen kavramına uyumluluğu arttırabilmek adına güvenlik sisteminin de etmen gibi hareketli bir yapıya sahip olabilmesi açısından ve güvenlik bağlamında etmenin platform bağımlılığını ortadan kaldırmak amacıyla tüm güvenlik fonksiyonlarının gerçeklenmesi sorumluluğu etmenin kendisine yüklenmiştir. Konakların zararlı hareketli etmenlerden korunabilmesi için konağı adına bu sorumluluğu bir özel yönetim etmeni üstlenmektedir ve konağa bu etmenin kontrolü olmadan girişin gerçekleşmesi mümkün değildir. Tüm yayınla/abone ol haberleşmesi yani hareketli etmenin konağı ile mesajlaşma arakatmanı ve hedef konak arası TLS kullanımı ile taşınım seviyesinde güvenilir hale getirilmesi önerilmektedir. Bu sayede mesajlaşma ara katmanı ile kaynak ve hedef konklar arasında araya girme saldırılarının gerçekleşmesi engellenmiş olup tüm veri tarfiği de dinleme ataklarına karşı korunmuş olmaktadır.Bir konağa göç eden hareketli etmen bu konağın özel yönetim etmeni le karşılandığında yapılan ilk iş konağa gelen etmenin kimlik kontrolünü yapmaktır. Bu işlemin gerçeklenmesi için sayısal imzanın açık anahtar alt yapısıyla birlikte kullanılması önerilmiştir. Bu kontrolün başarılı olmasıyla gelen etmenin yolda bir değişikliğie uğrayıp uğramadığı etmenle birlite gelen imzalı özeti ile burada alınan yeni özetin karşılaştırılmasıyla yapılması önerilmiştir. Eğer bu kontrol de başarılı sonuçlanırsa artık hareketli etmen göç ettiği konak üzerinde çalıştırılmaktadır. Eğer kimlik doğrulama veya bütünlük kontrollerinden biri bile

başarısız olursa etmen güvenilemedeğinden dolayı ihmal edililerek hiç çaılştıtılmaz. Bir kere çalıştırlmaya başlayan etmen artık burada üreteceği verinin gizliliğini sağlamaktan kendisi sorumludur ve bu amaçla taşıdığı yada ürettiği veriyi şifreler ve bu şekilde muhafaza etmye çalışır. Bu çalışmada önerilen güvenlik mekanizmasının en önemli getirisi göç eden etmenin henüz çalıştırlmadan kimlik ve bütünlük kontrollerinin konak adına yönetici etmenle yapılmasıdır. Bu özellik eğer göç eden etmenle ilgili bir güven sorunu olması durumunda erkenden durumun ortaya çıkarılarak ve etmene zararlı bir faaliyet gerçekleştirme fırsatı tanımadan henüz çalışma imkanı vermeden önlemin alınmasını sağlamaktadır. Tüm güvenlik gereksinimlerinin gerçeklenmesinde halihazırda kullanılmakta olan açık standartların ve teknolojilerin kullanılması önerilmiştir. Önerilen mimari modüler yapıda olup mimariyi oluşturan yapısal bloklar bir birlerine gevşek bağlı olup birinde kullanılan teknolojinin değiştirlmesi diğerlerini etkilememektedir. Bu sayede mimariye yeni gelişen teknolojilerin adaptasyonu kolaylığı ile yeni gelişen tehtitlere karşı yüksek güvenilirlik sağlanması amaçlanmıştır.

SECURITY FOR TOPIC BASED AGENT MIGRATION

SUMMARY

Agents are autonomous, proactive, collaborative software entities acting on be half of their respective users. Agents have goals and beliefs and they can change or update their goals and beliefs according to the information that they gather from their environments without any help of outsiders. The autonomy and proactivity properties of agents let them to decide and act on the knowloadge they extract on their own without any intervention. Furthermore, agents are social entities therefore they communicate and collaborate with each other in order to accomplish their goals. Beside the properties listed above agents may have mobility as one of their features. Mobile agents visit some hosts in order to meet their needs by traveling on computer networks.

Multi agent systems consist of collaborating mobile and fixed agents. Multi agent systems are very important in realizing decentric, highly distributed, autonomous and proactive software systems. Especially mobile agents improve computer nektwork efficiency by reducing the total amount of data carried over the network. This utilization is satisfied by migration of the agent itself instead of exhanging high volume of data. In order to accomplish a task some entities may send very much information utilsing some network protocol to each other and this operation along with the data being sent may consume lots of network resources. However, mobile agents travel on the network and can operate locally on the remote platform with minimum network load by eliminating the message exchages. This makes network resources to be available to the others thus efficiency over the network is abstractedly improved. Agent migration via topic based publish/subscribe paradigm brings loosely coupling between communicating entities by eliminating the need of aquintance of the interacting parties before they meet with each other. This birngs great flexsibility over dynamic network topology and high scalability to the applications develpoed using the topic based agent migration paradigm. Due to all the agent properties listed above multi agent systems find very large application space including but not limited to process control, manufacturing, logistics, information and network management areas. Besides, recent improvements in internet technology emerges a new field called electronic commerce. Muli agent systems conceptually is very suitable for the area that is why nowadays the most popular application field of the multi agent systems is electronic commerce. In all application fields but especially in electronic commerce security of the multi agent systems must be esatbilshed in order to let the systems be realized. When the damage arised due to the lack of proper security is considered, it is understood that the fact that such a commerce appliacaiton can not be realized without the security needs are satisfied.

Without the neccessary security measures are in place, problems like monetary losses, reputational losses, data breach and service outages for service providers may emerge. Securing multi agent systems is not much different than that of already

existing fundamental communicaiton and computer systems because the requirements are the same. So, the most practical way of securing multi agent systems is simply adapting the techniques and technologies applied on the already existing systems to the multi agent systems. First of all it is inevitable to perform authenticaiton in order to establish the trust because the agent must be the agent whom it claims it is or the user must be the user whom the agent claims to act on whose behalf. If an agent impersonates another agent it can act maliciously in the name of the other by decieving the one it communicates. However, proper authentication prevents such impersonating activities. The authenticaion may be successful due to trusted origin of the data but the data still can cause harm on the recieving side due to the active attacks like modification applied to the data transfered on the way by malicious entities. That is why data integrity is another important security measure to detect that if a message sent by an agent is modified by a malicious entity on the way or not. Another important precaution that must be considered is data privacy. Confidential data must be kept private and in order to prevent malicious entities to capture sensitive data, encryption must be applied to the data. Because the data is encrypted, it is not important even if it was sniffed by others. As long as the malicious entities do not know the encryption key they cannot reach the real sensitive data.

In this work security for multi agent systems in which the agents migrate via publish/subscribe paradigm is tried to be established. Therefore, a security architecture which meets the aforementioned security requirements is proposed. In order to improve the adaptability of the mobility concept security functionality is embeded into the agent itself. This lets the agent to carry the security mechanism with it and this also decouples an agent from its platform in the security context. This also promotes the inter-platform mobility again in the context of security. In order to secure the hosts visited a special agent called Handler Agent takes the security check responsibility on behalf its platform. It is proposed to use TLS for transport level security. Thanks to TLS, the high level whole communicaiton becomes resistant to man in the middle attacks which causes impersonation for one or all of the communicating parties. TLS also provides data confidentiality by encrypting the all data trafic between communicaiting entities. When an agent migrates to a host it is met by Handler Agent of the platform visited. Fisrt thing performed by the Handler Agent is to authenticate the incoming mobile agent by checking the digital signature appended to the the agent itself. Second, the integrity of the incoming agent is checked by comparing the computed hash of the agent with the hash extracted from the signature. If the two hashes are equal then this means that the incoming agent is integral in other words, it is not modified during its trip to the host. If the two aforementioned operations are successful then the incoming is executed. If one or both of the security measures results in failure then the incoming agent immediately discarded due to lack of trust. According to the security mechanism proposed once an agent gets executed from now on it is responsible of securing the data it produces or carries with itself. In order to secure the sensitive data agent encrypts the data. The most important outcome of the proposed mechanism is that it detects the unturstedness at a very early stage before the incoming agent is executed. This prevents the mobile agent to take any chance to perfom a malicious activity. İt is proposed to use open standarts and technologies to satisfy the security requirements. The proposed architecture is modular and building blocks are loosely coupled to each other and this provides independency between the blocks. That is, the technologies and algoritms used in each block can be changed with no effect onto the others. This

property provides easily adaptation to the new technologies and algorithms against emerging new threats.

1. GİRİŞ

Çoklu etmen sistemleri özellikle hareketli etmen sistemleri dağıtık sistemlerle hesaplama yapılabilmesi için hesaplama gücünün otonom, akıllı ve etkin bir şekilde dağıtılabilmesi açısından çok iyi bir ortam sunarak yüksek performanslı merkezi olmayan yazılımların geliştirilmesinde oldukça önemli bir yere sahiptir. Özellikle haraketli etmenler bilgisayar ağlarında otonom olarak hareket ederek nerede gerekirse orada çalışabildiklerinden dolayı bir problemin çözümünde ortaya çıkabilecek veri trafiğini en aza indirerek haberleşme performansını arttırıp ağ bant genişliğinin daha faydalı kullanılmasını da sağlamaktadır. Haberleşme veya hareketli etmenler için göç alt yapısının konu tabanlı yayınla/abone ol yapısının üstüne kurulması haraketli etmenlerin yukarıda sıralanan özelliklerine ekstra esneklik ve ölçeklenebilirlik getirmektedir (Karzan, Erdoğan, 2013). Bu şekilde ileri derecede ölçeklendirilebir, esnek, otonom ortamların özellikle e-ticaret, bankacılık gibi hasas veri üzerinde işlem yapılan alanlarda kullanılacağı düşünüldüğünde hareketli etmen sistemlerinin belirli bir düzeyde güvenliğinin sağlanması kaçınılmaz olmakatadır. Bu bağlamda kimlik doğrulama, veri bütünlüğü, veri gizliliği fonksiyonlarının yüksek güvenilirlik düzeyinde ortaya konulması gerekmektedir.

1.1 Tezin Amacı

Bu çalışmanın amacı konu tabanlı yayınla/abone ol yöntemi ile göç eden hareketli etmen sistemlerinin güvenlik alt yapsını oluşturmaktır. Bu çalışmada katmanlı bir güvenlik mimarisi önerilerek uygulama seviyesi (etmen seviyesi) ve taşınım seviyesi güvenlik fonksiyonları birbirlerinden ayrılmıştır. Bu noktadaki konuların ayrılığı her iki katmanın birbirlerinin değişimlerinden etkilenmemesi için gereklidir. Taşınım seviyesi güvenliğin TLS (Transport Layer Security) ile sağlanması ön görülmüş olup etmen seviyesi güvenlik için ise hareketli etmen kavramını destekleyici olması açısından hareketli bir yapıya sahip modüler bir mimari önerilmiştir. Güvenlik mekanizmasının hareketliliği gerekli fonksiyonların hareketli etmenin kendisne yüklenmesi ile sağlanmaktadır. Bu sayede etmenler ile etmen platromu arasında

güvenlik bağlamında gevşek bir bağlılık kurularak platformlar arası etmen göçünü destekleyerek bir arada çalışabiliriliği kuvvetlendirmekedir. Etemen seviyesinde ziyaret edilen güvenliği yine etmenler tarafından sağlanmaktadır. Burada bu görev konağı adına yönetimsel işler için özelleşmiş bir etmen olan Yönetici Etemen`e (Karzan, Erdoğan, 2013) verilmektedir. Önerilen sistemde bir konağa göç eden etmen henüz çalışma fırsatı yakalayamadan kimlik ve bütünlük doğrulamasına tabi tutularak güvenilirliği sorgulanmaktadır. Eğer göçeden hareketli etmen bu kontrolleri geçemezse doğrudan ihmal edilerek muhtemel bir saldırının erkenden daha hiç etkisi olamadan önüne geçilmiş olunmaktadır. Önerilen mimaride veri gizliliği etmenler tarafıdan veri şifrelemesi yapılmasıyla sağlanmaktadır. Tüm bu bahsedilen kontrollere ragmen güvenilir bir etmen de konak üzerinde zararlı işlemler yapabilmeketedir. Bu durumda artık zararlı aktivitenin engellenmsei değil inkar edilemeyecek bir şekilde kayıt altına alınması önerilmiş olup bu işlem de sayısal imzaların kullanılmasıyla gerçeklenmiştir.

1.2 Literatür Araştırması

Çoklu etmen sistemlerinin ve özel olarak hareketli etmen sistemlerinin güvenliğinin sağlanması bu kavramların üzerine bina edilen yazlım sistemlerinin gerçeklenmesi ve bu yzalımların internet gibi açık ağlar üzerinde çalıştırılabilmesi açısından büyük bir öneme sahiptir. Bu nedenle bu alanda pek çok çalışma mevcuttur. Bu çalışmada önerilen yapıya benzer bir güvenlik yapısı güvenlik fonksiyonlarının etmenlerin kendilerine yüklendiği bir mimari olarak Novak ve arkadaşları tarafından önerilmiştir (Novak, 2003). Fakat bu çalışmada etmenlerin sağladığı güvenlik sadece etmenler arası haberleşmede kullanılmaktadır. Bu durumda konakları zararlı etmenlere karşı korumak için konak platformlarının da ayrıca bir güvenlik mekanizmasına sahip olması gerekmektedir. Bu durum etmenlerin sağladığı güvenlik yapısı ile konakların sağladığı güvenlik yapılarının uyumluluk göstermesi gerekliliğini ortaya çıkararak etmenler ile etmen platformları arasında bağımlılık oluşturmaktadır. Prem serbest hareket eden yani henüz hareketine başlamadan önce ziyaret edeceği konakların rotasını bilmeyip bu bilgiyi dinamik olarak oluşturan etmenlerin zararlı konakların gerçekleştirebilecekleri dinleme gibi pasik ataklara karşı korunması için üç faktörlü bir güvenlik mimarisi önermiştir (Prem, 2012). Dolayısıyla bu çalışmada veri gizliliğine ve rota bilgisi korumasına konsantre olunmuş olunup kimlik doğrulama ve

ziyaretçi hareketli etmen bütünlüğü kontrolüne kısaca değinilimiştir ve veri bütünlüğü kontrolü sadece belirli bir veri uzunluğuna karşı kıyaslamayla sağlanmaya çalışılmıştır. Bu kontrolden veri sıkıştırma ile kolaylıkla geçmek mümkündür. Zhang ve arkadaşları seçilmiş güvenilir konaklardan oluşturduları bir ağ üzerinde etmen dolaşımınının güvenlik garantili poliçeler ile sağlanması için gayet güçlü bir güvelik alt yapısı önermiştir (Zhang, et al., 2009). Fakat bu yönteme göre ziyaret edilecek konakların henüz hareket başlamadan bilinmesi esasına dayanmakatadır ve bizim çalışmamızda güvenliğini sağlamaktan sorumlu olduğumuz konu tabanlı etmen göçü kavramına uymamakatdır. Çünkü konu tanalı etmen göçü kavramı hedef ve kaynak konakların bir birlerini tanımaması prensibi üzerine kurulmuştur (Karzan, Erdoğan, 2013). Yang ise hareketli etmen ve ziyaret edilen konak arasında kurulan güven üzere rol tabanlı erişim kontrolü mimarisi önermiştir (Yang, 2006). Çalışmanın asıl amacı güçlü bir erişim kontrolü sağlamak olup etmen ile konak arasındaki güven ortamı etmen çalıştırılmaya başladıktan sonra kimlik kontrolü ile yapılmaya çalışılmıştır. Bu durum çalışma fırsatı yakalamış olan kötü niyetli bir etmen için daha henüz zararlı veya güvenilir değil şeklinde bir damga vurulmadan belki de zararlı işler görebilmesi için yeterli miktarda zaman vermekte olabilmektedir.

2. ETMEN TEKNOLOJİSİNE GENEL BAKIŞ

2.1 Etmenler

Etmenler, bir yandan yazılım sistemlerini kavramsallaştıran, tasarlayan ve gerçekleyen var olan yöntemleri geliştiren ve öte yandan eski yazılım birleştirme problemine çözüm olabilecek en önemli paradigmalardan biridir.

"Etmen" veya "yazılım etmeni" terim olarak bir çok teknolojide yer almış ve çokça kullanılmıştır. Örneğin, yapay zeka, veri tabanları, işletim sistemleri ve bilgisayar ağları literatüründe sıkça rastlanır. Etmen için tek bir tanım olmamasına rağmen, tüm tanımlar etmenin herhangi bir sistemle birlikte çalışabilecek bir arayüz sunan otonom olan özel bir yazılım bileşeni olduğu ve kendi gündemine uygun olacak şekilde müşterileri için çalışan bir insan ajan olarak davrandığı yönünde hem fikir olurlar. Etmen sistemi bir çevre içerisinde çalışan ve kullanıcıları ile temasa geçen tek bir etmenden bile oluşabilirken genel olarak çoklu etmenlerden oluşur. Bu çoklu etmen sistemleri karmaşık sistemleri modelleyebilir ve etmenlerin çelişen veya ortak olan hedefleri olma olasılığını gündeme getirir. Bu etmenler diğerleriyle hem iletişim ve anlaşma yoluyla doğrudan hem de ortama müdahale ile dolaylı olarak etkileşebilirler. Etmenler karşılıklı faydalanmak üzere birbirleri ile işbirliği içerisinde olabilirler veya sadece kendi hedeflerini gerçekleştirmek üzere çalışabilirler.

Böylelikle etmenler otonomdurlar (özerktirler) çünkü insanların veya diğerlerinin doğrudan müdahalesi olmaksızın işlerler ve kendi müdahaleleri ve iç durumları üzerinde kontrolü kendileri sağlarlar. Etmenler sosyaldirler çünkü görevlerini tamamlamak üzere diğer etmenlerle veya insanlarla işbirliği yaparlar. Etmenler tepkiseldir çünkü çevrelerini algılarlar ve çevrede oluşan değişikliklere belirli bir zaman içerisinde cevap verirler. Ayrıca etmenler proaktiftirler çünkü sadece çevresinde olup bitenlere tepki göstermenin ötesinde sorumluluk alarak girişim yaparak hedefe yönelik davranış sergilerler.

Üstelik eğer gerekirse etmenler hareketli de olabilirler. Bu özellik ile etmenler ağ üzerinde farklı düğümler arasında seyahat etme yeteneği kazanırlar. Etmenlerden

dürüst olmaları beklenir. Bu yüzden dürüstlükte bir etmen özelliğidir. Bu özelliğe göre etmenler bilerek yanlış bilgi vermeme garantisini sunmalıdırlar. Etmenler yardım sever de olabilirler. Bu özellikleri gereği kendilerinden ne istenirse onu yaparlar. Rasyonel de olabilirler, böylece her zaman hedeflerini gerçekleştirmek üzere çalışırlar ve hedeflerinin gerçekleşmesini engelleyecek şeyleri ortadan kaldırmaya çalışırlar. Öğrenme özelliği ile de etmenler bulundukları çevreye ve kullanıcılarının isteklerine uyum sağlarlar.

2.2 Etmen Sistemlerinin Uygulama Alanları

Çoklu etmen sistemleri kişisel destek ile açılan küçük sistemlerden kritik endüstri uygulamalarına kadar varan çeşitlilikte, artan geniş bir uygulama yelpazesinde kullanılmaktadır (Jennings ve Wooldridge, 1998).

Endüstri uygulamaları çoklu etmen sistemleri için çok önemlidir cünkü bu uygulamalar ilk çoklu etmen sistemleri tekniklerinin denendiği ve ilk potansiyellerinin gösterildiği uygulamalardır. Endüstriyel çerçevede kullanılan çoklu etmen uygulamaları örneğin, süreç kontrolü (Jennings, 1994), sistem teşhisi (Albert ve diğ., 2003), imalat (Parunak, 1987), ulaşım lojistği (Neagu ve diğ,. 2006) ve ağ yönetimi (Greenwood ve diğ.,) alanlarını içerir.

Çoklu etmen sistemlerinin en önemli uygulama alanlarından biri bilgi yönetimidir (Decket ve Sycara, 1997). Özel olarak internet, dağıtık doğasından ve bir sürü veri içermesinden dolayı çoklu etmen sistemleri için ideal bir alan olarak gösterilmiştir. Mesela, etmenler bu yoğun bilgi kümesinde arama ve süzgeçleme yapmak için kullanılabilir (Klusch, 2001). İnternet ayrıca etmen teknolojilerinin ticaret ve iş süreç yönetimi alanlarında kullanılmasını da zorlamıştır. Aslında internet üzerinden ticaret yaygınlaşmadan önce iş süreçleri yönetimi neredeyse tamamen insan etkileşimleri ile ilerlemekteydi. Ne zaman mal alınacağına, ne kadar ödemek istediklerine, …vs hep insanlar karar vermekteydi. Şimdi elektronik ticaret ve otomatikleştirilmiş iş süreçleri bir çok organizasyonda merkezi bir rol oynamakta çünkü bu sistemler iş süreçlerine katılan farklı birimlerin geliştirilmesini sağlayan imkanlar sunmaktadır. Bu senaryoda çoklu etmen sistemlerinin hem iş süreçleri yönetim sistemlerinin modellenmesi ve tasarımına uygun olduğu hem de bu süreçlerdeki bazı ya da tüm adımların otomatikleştirilmesinde anahtar bileşenler olduğu (Jennings ve diğ., 1996) gösterilmiştir.

Trafik ve ulaşım da önemli bir uygulama alanıdır. Trafik ve ulaşım süreçlerinin dağıtık bir doğası olmasından ve bu süreçler içerisindeki birimlerin birbirlerine kuvvetli bağımsızlıkları gerektirmesi çoklu etmen sistemlerini bu alanda etkili ticari çözümler sunma aracı yapmaktadır (Neagu ve diğ., 2006).

Telekomünikasyon sistemleri de çoklu etmen sistemlerinin başarı ile kullanıldığı bir diğer uygulama alanıdır. Esasen, telekomünikasyon sistemleri gerçek zamanda izlenmesi ve yönetilmesi gereken birimlerin birleşmesi ile oluşan büyük, dağıtık ağlardır. Burada telekomünikasyon şirketleri ve hizmet sağlayıcıları kendilerini rakiplerinden farklılaştırmak üzere daha iyi, daha hızlı ve daha güvenilir hizmetler sunarak rekabetçi bir pazar oluştururlar. Böylelikle çoklu etmen sistemleri hem böyle dağıtık ağların yönetiminde hem de ileri telekomünikasyon hizmetlerinin gerçeklenmesinde kullanılırlar (Fricke ve diğ., 2001; Hayzelden ve Bourne, 2001; Greenwood ve diğ., 2006).

Bir çok çoklu robotik sistemleri farklı robotlar arasındaki koordinasyonu sağlamak amacı ile çoklu etmen ve dağıtık planlama tekniklerini kullanmaktadır. Dağıtık planlama ve iş sıralama teknikleri etkili, çoklu gezgin koordinasyon planları, izleme planı üretiminde ve gerektiğinde yeniden planlanmasında kullanılır.

Başka enteresan çoklu etmen sistemleri uygulamaları sağlık hizmetlerinde bulunabilir (Moreno ve Nealon, 2003). Aslında çoklu etmen sistemleri sağlık hizmetleri alanında farklı problemlerle ilgilenmek üzere çoktan önerilmiştir. Bu farklı problemler, hasta düzenleme ve yönetimi, kıdemli ve topluluk bakımı, medikal bilgi erişim ve yönetimi ve karar destek işlerini içerir.

2.3 Etmen Hareketliliği

Hareketli etmenler (White, 1996) iki farklı disiplinden (Brown ve Rossak, 2005) türemiş bir paradigmadır. Bu disiplinlerden bir tanesi yapay zekadır ki etmen kavramı burada ortaya konmuştur (Russel ve Norvig, 1995) ve diğer disiplin de kod hareketliliğini tanımlayan dağıtık sistemlerdir (Picco, 2000).

Standart tanımlara göre hareketli etmenler, hareketli olmayan etmenler ne ise (otonom, tepkisel, proaktif ve sosyal …vs) bu özellikleri aynen içerip ayrıca ek olarak bir de hareketlidirler. Kendilerine atanan görevleri yerine getirmek üzere platformlar arasında göç ederler.

Dağıtık sistemler açısından bakıldığında bir hareketli etmen tek bir kimliği olan ve kendi kod, veri ve durumunu ağ üzerindeki makinalara taşıyan programdır. Bunu başarmak üzere, hareketli etmenler çalışmalarını herhangi bir anda durdurup, başka bir yerde çalışmalarına devam edebilirler. Hareketli etmenleri diğer klasik paradigmalara ilişkin şöyle konumlandırabiliriz (Picco, 2000):

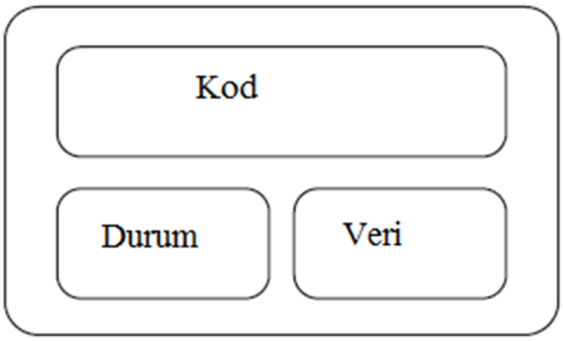
İstemci/Sunucu: En çok kullanılan paradigmadır. Hizmetler bir sunucu tarafından sağlanırken, bir veya daha fazla ve genel olarak uzaktaki istemci(ler) tarafından tüketilirler.

Uzaktan çalışma: Bir bileşen diğer bileşene uzakta çalışmak üzere kod gönderir. Bu işlem ya uzak bileşenden istek gelmesi ile ya da var olan bir kontratın bir parçası olarak gerçekleştirilir. Bir kere çalıştırıldımı çalışan bileşen işlemi başlatan bileşene sonucu döner.

Hareketli etmenler: Bir bileşen kendini (eğer izin verildi ise bir diğerini de olabilir) uzaktaki makinaya orada çalışmak üzere gönderir. Bileşen geçişi kod, veri ve durum bilgileri değişmeden gerçekleşir.

Hareketli bir etmen üç kısımdan oluşur: kod, durum ve veri. Kod bir etmenin bir platforma göç etmesiyle çalıştırılan yönüdür. Durum ise etmen için veri çalıştırma ortamıdır. Bu ortam program sayacı ve çalışma yığınını içerir. Veri, etmenler tarafından kullanılan değişkenlerden oluşur. Mesela bilgi, dosya tanımlayıcıları, … gibi.

Aşağıdaki şekilde etmenin bahsedilen kısımları görülmektedir.



Şekil 2.1 : Hareketli etmen temel yapısı.

2.4 Hareketli Etmenlerin Bazı Avantaj ve Dezavantajları

Etmenlerin avantajları ve dezavantajları üzerine bir çok tartışma gerçekleşmiştir. Genel olarak hareketli olmayan etmenlerle karşılaştırılırlar. Bazı tipik avantaları:

Asenkron ve bağımsız işleme: Yeni platforma göç ettikten sonra etmenler görevlerini yerine getirmek için sahipleri ile iletişime geçmeleri gerekmez. En fazla belki sonuçları geri dönmeleri gerekebilir. Bu özellikle sınırlı kaynaklara sahip olan hareketli cihazları göz önünde bulundurduğumuz zaman yararlıdır. Bir etmen başka bir makinaya karmaşık görevleri yerine getirmek üzere göç edebilir ve periyodik olarak sonuçları dönebilir.

Hata dayanıklılığı: Etmenler, problem tesbit edildiğinde hata durumlarını adreslemek ve çözmek için farklı platformlara göç edebilirler. Aynı şekilde göç edilecek hedef ayakta değilse bir aracı geçici konak olarak seçilebilir. Bu özellik etmenleri düşman ve zarar veren çevrelere karşı uygun bir çözüm olarak getirir.

Veri denizi uygulamaları: Hareketli etmenler uzaktaki çok miktardaki verinin işlenmesi uygulamalarına çok uygundurlar. Hareketli etmenler, verinin kendilerine gelmesindense kendileri veriye doğru hareket ederler ki birçok durumda daha verimli bir seçenektir.

Hareketli etmenlerin bazı dezavantajları da bulunmaktadır. Mir'in tarif ettiklerinden bazıları (Mir, 2004):

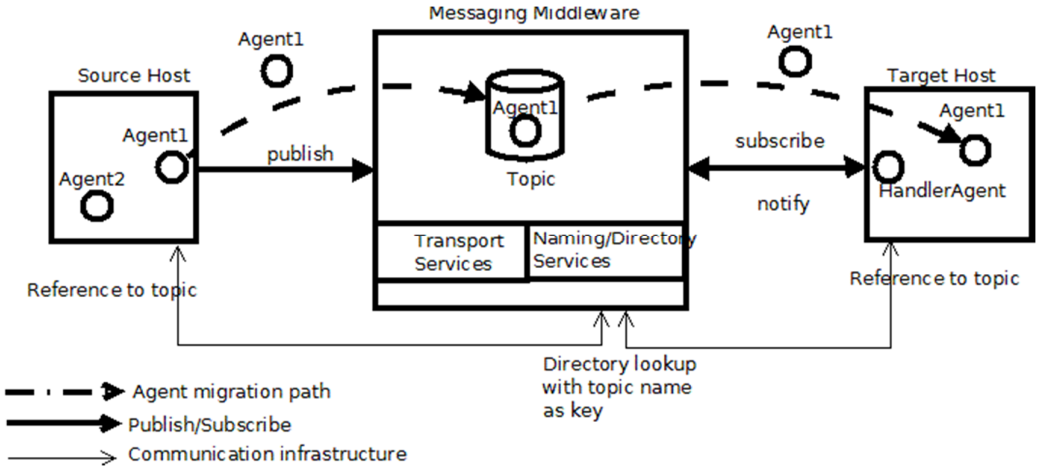
Taşınabilirlik ve standartlaştırma: Eğer etmenler genel haberleşme standartlarını takip etmezlerse birlikte çalışamazlar. OMG MASIF (Mobile Agent System Interoperability Facility) veya FIPA (Foundation of Intelligent Physical Agents) gibi standartların benimsenmesi özellikle platformlar arasi hareketlilik için gereklidir.

Güvenlik: Hareketli etmenlerin kullanımı güvenlik problemlerini de beraberinde getirir. Herhangi hareketli bir kod potansiyel bir tehdit sunar ve başlatılmadan önce dikkatlice kimliği doğrulanmalıdır.

2.5 Yayınla Abone Ol Modeli ile Konu Tabanlı Etmen Göçü

Karzan, M.A. ve Erdoğan, N.'nin çalışmalarında etmenler bir konaktan diğerine yayınla/abone ol haberleşme mekanizmasına göre göç etmektedirler (Karzan ve Erdoğan, 2013). Daha özel olarak etmenler bir kaynak konaktan hedef konaklara

ACL (Agent Communication Language) mesajları içerisinde standart etmen haberleşmesine uygun olarak ve yayınla/abone ol alt yapısı üzerinden standart dağıtık yazılım haberleşmesine uygun olarak hareket ederler. Haberleşme paradigmasından dolayı kaynak ve hedef konaklar bir aracı konak vasıtasıyla birbirlerinden yalıtılırlar. Modelde kullanılan konu tabanlı yayınla/abone ol düzeni sayesinde kaynak ve hedef konaklar önceden birbirlerini tanımak zorunda değildirler. Sadece aralarındaki aracı konağı tanırlar. Böylece sisteme yeni eklenecek konaklar veya sistemden ayrılacak konaklar sistem işleyişine ek yük getirmez. Yayınla/abone ol kalıbı sistemi uzay, zaman ve senkronizasyon bağımlılıklarından kurtarır. Ayrıca sistem ölçeklenebilir ve dinamik ağ topolojisine karşı esnek bir yapı kazanır. Bahsedilen çalışmada etmenler arasında veri göndermektense standartlara uygun biçimde konaklar arasında etmen göndermek ile ağ üzerindeki yük azaltılırken sistem performansı arttırılmış olur. Böylece çalışma yeni bir etmen göçü kavramı tanıtır. Sistem mimarisi aşağıdaki şekildeki gibidir.



Şekil 2.2 : Sistem mimarisine genel bakış (Karzan, Erdoğan, 2013).

Bu mimari ile anlatılan çözümde, kaynak konak hizmet sunucu olarak görev alır ve özel hizmetler sunan etmenleri üretmekle sorumludur. Hizmet kaynağı etmen çalıştırıldıktan sonra kendisini mesajlaşmayı sağlayan ara katmanın önceden belirlenmiş olan konu veya konularına yayınlar. Yine çözümde ifade edilen hedef konak ise kendi amaçları doğrultusunda hizmet sunan etmenin verdiği hizmetlere ihtiyaç duyan bir hizmet tüketici olarak adlandırılır. Hedef konak hizmet veren etmenler ile uyarılabilmek için mesajlaşma ara birimindeki önceden tanımlı konulardan ilgi duyduklarına abone olur. Konak adına abone olma ve bildirim yönetimi işlemlerini yerel bir etmen olan yönetici etmen yapar. Yönetici etmenin

kendi konağının niyetlerini biliyor olması beklenir. Böylelikle konağının ilgi duyacağı konuları bilir ve bunlara abone olur. Abone olunan konuya bir etmen yayınlandığı zaman mesajlaşma ara birimi yayınlanan etmeni ACL mesajının gövdesine sarmalanmış bir biçimde hizmet sunan etmeni hedef konak üzerindeki yönetici etmene bildirir. Yönetici etmen hizmet sunan etmeni ACL mesajından soyup yerel konakta çalışmaya hazır hale gelmesi için gerekli olan işlemleri gerçekleştirir. Bu işlemler genel olarak platform tarafından sunulan etmen yönetim hizmetlerinin sağlanması için gereklidir. Bu işlemlerin başarılı bir biçimde icra edilmesinden sonra hizmet sunan etmen yeni bulunduğu ortamın koşul ve durumunu değerlendirip, konak için gerekli olan hizmeti konağa özgü bir biçimde vermek üzere çalışır ve işi bitince de sonlanır. Bu bağlamda etmen bir konaktan ötekine standart etmen haberleşme mesajı biçiminde gideceği konağın kimliğini önceden bilmeksizin seyahat eder. Kaynak ve hedef konaklar bir ara birim aracılığıyla seçilen konu veya konular üzerinden birleşirler. Bu şekilde ara birim üzerindeki bir konu hizmet sunan ve hizmet tüketen konakların buluşma notasıdır. Konaklar mesajlaşma ara birimi üzerinde verilen konu isimlerini anahtar olarak kullanarak dizin araması gerçekleştirirler. Bu arama işlemini kaynak konak tarafından üretilen hizmet sunan etmenler kendilerini yayınlamadan önce gerçekleştirirken hedef konak üzerindeki yönetici etmenler de abone olma işlemi sırasında gerçekleştirirler. Mesajlaşma ara birimi konakların ilgilendikleri konulara birer referans döner. Bu sayede konaklar için buluşma noktası bina edilmiş olur.

Karzan ve Erdoğan'ın bu çalışmasında yayınla/abone ol paradigması, JMS (Java Message Service) belirtimlerine uygun olarak sunulmuş olmasından dolayı standartlara uygun bir haberleşme çözümü sunmaktadır. Bu yönüyle çalışmada sunulan çözüm sayesinde etmen sistemleri diğer etmen sistemlerine veya hatta öteki yazılım paradigmaları ile hazırlanan sistemlere (özellikle nesneye yönelik sistemlere) kolaylıkla entegre edilebilir.

3. VERİ GÜVENLİĞİ VE DOĞRULAMA

3.1 Simetrik Şifreleme ile Veri Güvenliği

Simetrik şifreleme gizli anahtarlı şifreleme veya tek anahatrlı şifreleme olarak ta bilinmektedir. Simetrik şifreleme beş temel unsurdan oluşmaktadır; açık metin, şifreleme algoritması, gizli anahtar, şifreli metin ve şifre çözme algoritması (Stallings, 2011). Açık metin, şifreleme algoritmasına girişi oluşturan orjinal metindir. Şifreleme algoritması, açık metin üzerinde çeşitli yer değiştirme ve dönüştürme işlemleri uygulanmasını sağlayan algoritmadır. Gizli anahtar da yine şifreleme alogritmasna giriş olarak verilmekte olup şifreleme algoritması tarafından uygulanacak olan değiştirme ve dönüştürme işlemleri bu gizli anahatra dayanmaktadır. Şifreli metin, algoritmanın çıkışındaki döüştürülmüş metindir. Şifre çözme algoritması ise şifreleme algortimasının aynısı olup şifreleme algoritmasının ters sıra ile uygulanmasından ibarettir. Eğer şifre çözme algoritmasına şifreli metin ve bu metni oluşturmak için kullanılan gizli anahtar şifre çözme algoritmasına giriş olarak beslenirse çıkışta orjinal açık metin elde edilmektedir. Simetrik şifrelemenin güvenle kullanılabilmesi için öncelikle gizli anahtarın göderici ve alıcı arasında gizlilik içerisinde paylaşılması gerekmete olup şifreleme algortimasının da yeterince kuvvetli olması gerekmektedir yani şifreleme algoritmasını bilen bir kimsenin bir kaç şifreli metini ele geçerimesiyle açık metinlere veya şifreleme de kullanılan gizli anahtarlara erişememesi gerekmektedir (Stallings, 2011). Bu özellikler genel olarak güvenli hesaplama temel prensipleri olan, şifreyi kırma işleminin maliyetinin şifrelenmiş bilginin değerinden daha yüksek olması gerekliliği ve şifreyi kırabilmek için harcanması gereken zamanın şifrelenmiş bilginin faydalı yaşam süresinden uzun olması gerekliliği ilkelerine dayanmaktadır (Stallings, 2011).

Simetrik şifreleme, blok şifreleme ve akış şifreleme olmak üzere iki genel yöntemle uygulanmakatdır. Bu yöntemlerde şifre kırma (gizli anahtarın elde edilmesi veya şifreli metinden açık orjinal metnin elde edilmesi) işlemini karmaşıklaştırmak için bir takım blok şifreleme modları bulunmaktadır.

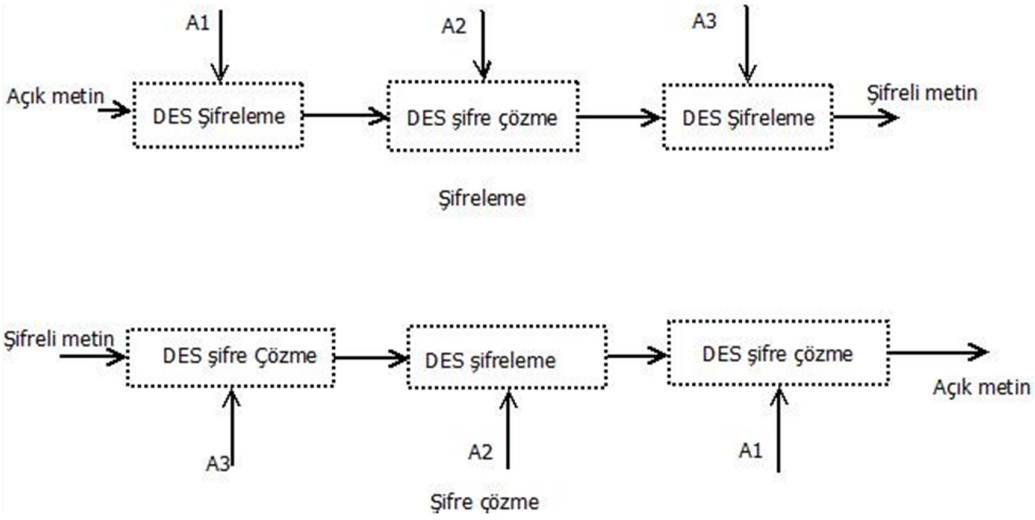
3.1.1 Simetrik blok şifreleme

Simetrik şifreleme algoritmalarından en yaygın olarak kullanıları blok şifreleme algoritmalrıdır. Blok şifreleyiciler şifrelenecek olan original açık metni belirli sabit uzunluktaki bloklara ayırarak bu bloklara şifreleme algoritmalarını gizli anahtarla uygulayarak yine aynı sabit uzunluktaki şifreli metinlere dönüştürürler. Blok şifreleyiciler arasında en önemli olanlar DES (Data Encryption Standart), üçlü DES ve AES (Advaned Encryption Standart) olarak sıralanmaktadır (Stallings, 2011).

DES: Bu simetrik blok şifreleme yönetimi NIST tarafından tanımlanmış olan standart üzerine kurulmuş olup DEA (Data Encryption Algorithm) şifreleme algoritmasını kullanmaktadır (Stallings, 2011). Bu algoritmada 56 bit uzunluğunda bir anahtar kullanılmakta olup orinal açık metin 64 bit uzunluğundaki bloklara ayrılarak her bir blok 16 çevirimden geçirilerek yine 64 bit uzunluğundaki şifreli metin bloğu oluşturulmaktadır. Son çevirim hariç her çevirimde şifrelenecek metin bloğu üzerinde yer değiştirme ve yerleştirme işlemleri yapılmaktadır. Son çevirimde ise 64 bit uzunluğundaki metin bloğunun ilk ve son 32 biti yer değiştirmektedir. Her bir çevirim 56 bit uzunluğundaki ana anahtardan 48 bit uzunluğundaki çevirim anahtarları oluşturulup çevrim içinde bu anahtarlar sırası ile kullanılmaktadır. Şifre çözme işlemi ise şifreleme işlemlerinin birebir aynısı olup kullanılan anahtarlar ters sıra ile işleme alınmkatadır. DES'in kullandığı algoritmanın zayıflıklarının bulunması için uzun zamanlar boyunca çalışılmış olsa da kayda değer bir sonuç üretilememiştir (Stallings, 2011). Fakat algortimanın zayıflığından ziyade kullanılan anahtar uzunluğu bu şifreleme yöntemini günümüz veri işlem kapasitesindeki gelişmler ile tek tek sıralı deneme ataklarına karşı dayanıksız hale getirmektedir (Stallings, 2011).

Üçlü DES (3DES): Bu yöntem DES algoritmasının üç defa ayrı ayrı anahtarlar ile birinin girişinin diğerinin çıkışına Şekil 3.1'de görilüğü gibi uygulanamsından ibarettir. Şifreleme işleminde açık metin önce DES şifreleme algortimasından A1 anahtarı ile muamele edilerek geçirilir ve sonra A2 anahatarnın kullanımı ile ilk şifreleyicinin çıkışına DES şifre çözme işlemi uygulanır son basamak ta ise ikinci basamak çıkışına A3 anahtarı ile bir daha DES şifrelemesi uygulanarak şifreli metin oluşturlmuş olunur. Şifre çözme işleminde ise ilk ve son basamaklarda şifre çözme ve ortadaki basamak ta ise şifreleme işlemi ugulanmaktadır. Üç tane 56 bit

uzunluğunda anahtar kullanılması toplamda 168 bit uzunluğunda bir anahtarın kullanılmasını sağlayarak DES'in en büyük zayıflığı olan anahtar uzunluğu problemi üçlü DES ile giderilmeye çalışılmıştır.



Şekil 3.1 : Üçlü DES şifreleme ve şifre çözme (Stallings, 2011)'den uyarlanmıştır.

Üçlü DES'in şifreleme şifre çözme ve tekrar şifreleme işlemlerinin bir birini takip eden yapısının altında bu yöntemin DES ile karşılıklı kullanılmasının sağlanmasının istenmesidir. Eğer A1 = A2 = A3 seçilirse üüçlü DES yöntemi DES'e indirgenmektedir. Bu özelliği ile üçlü DES ve DES karşılklı olarak yani gödericinin üçlü DES ve alıcının DES veya tam tersi bir durumda kullanılması sağlanmaktadır.

AES: Üçlü DES'in en büyük dezavantajı yazılım olarak gerçeklenmesinde ortaya çıkan yavaşlıktır. Bu dezavantajın da ortadan kaldırabilmesi için NIST tarafından düzenlenen yarışmayı kazanan algoritma AES olarak yine NIST tarafından standartlaştırılmıştır. Bu standart üçlü DES kadar kuvvetli bir yöntem olup bit yerine bayt üzerinde işlem yapması dolayısıyla da yeterince hızlı ve etkin olması sebebiyle yazlım gerçeklemelerine de uygun bulunmuştur. AES'te şifrelenecek açık metin blok uzunluğu 128 bittir. Anahtar uzunluğu ise 128, 192 veya 256 bit olabilmektedir. 128 bit uzunluğa sahip anhatar bu yöntemde en çok kullanılan anahatar çeşidir (Stallings, 2011). Algoritma on çevirimden oluşmaktadır. 128 bit uzunluğundaki açık metin girişi ve anahtar ayrı ayrı matrisler olarak algoritmaya verilmektedir ve bu açık metin matrisi üzerinde bayt değiştirme, satır kaydırma ve kolon karıştırma işlemleri uygulanarak şifreleme gerçekleştirilmektedir.

3.1.2 Simetrik akış şifreleme

Blok şifrelemede şifreleyiciye giriş belirli uzunlukltaki bloklara bölünerek verilmekte ve her bir blok için yine aynı uzunlukta şifreli metin blokları elde edilmekteydi fakat akış şifreyicilerde girişteki açık metin boklara ayrılmayıp sürekli bir şekilde şifreleyiciye beslenmektedir. Simetrik blok şifreleme akış şifrelemeye göre daha yaygın olsa da bazı uygulamalar için akış şifreleme daha uygundur. Bu uygulamalara örnek vermek gerekirse haberleşme kanalları üzerinde veri şifreleme/şifre çözme işlemi için akış şifrelemenin uygunluğu aşikardır. Akış şifreyiciler bir seferde bir bit şifrelemek üzere düzenelebilecek olsalar bile genel olarak bir seferde bir bayt şifreleme yapmaktadırlar. Şifreleyiciler giriş olarak açık metin akışının yanında bir ilk işlem vektörü ve bir de anahtar akışı alır. Anahtar akışı bir rastgele sayı üreticine bir ana ahtarın ilk işlem vekörü ile birlikte beslenmesi ile üretecin çıkışından her bir sefer için sekiz bit olarak elde edilir. Elde edilen bu sekiz bitlik anahtar her seferinde üretece geri beslenmesi ile anahtar akışının devamı üretilir. İlk işlem vektörü sadece bir defa anahtar akışının ilk elemanını oluşturmak için kullanılmaktadır. Her bir sekiz bit'ilk anahtar ile bir bayt açık metinin karşılıklı dışlama işlemine tabi tutulmasıyla bir bayt şifreli metin çıkıştan elde edilir. Anahtar akışı açık metinden bağımsız olduğu için bu akış çevirim dışı olarak hazırlanarak işlemler hızlandıralabilmektdir. En populer simetrik akış şifreyici RC4 olarak karşımıza çıkmaktadır (Stallings, 2011).

RC4: Bu algoritma SSL, WEP ve WPA gibi önemli ve popular güvenlik prokollerinde kullanım alanı bulmuştur. Bu alogritma 256 bayt uzunluğunda bir durum vektörü içerir ve bu vektör her bir anda 8 bit uzunluğundaki 0'dan 255'e kadar olan sayıların bir permütasyonunu içerir. Her bir açık metin baytı bu durum veköründen belirli bir sistemle seçilen bir 8 bit'lik sayı ile karşılıklı dışlanarak bir bayt uzunluğundaki şifreli metin elde edilir.

3.1.3 Blok şifreleme modları

Simetrik şifrelemenin çeşitli uygulamalarda kullanılabilmesi için NIST tarafından beş tane işem modu tanımlanmıştır. Bu modlar DES ve AES'te dahil olmak üzere tüm simetrik blok şifreme algoritmalarıyla kullanılabilmektedir. Bu modlardan en önemli olanları ve en çok kullanılanları ECB, CBC, CTR'dir (Stallings, 2011).

ECB (Electronic Code Book): Bu modda her bir b bitlik açık mein bloğu hep aynı anahtar ile şifrelenir. Bu moda kod kitabı denmesinin sebe ise belirli bir anhatar için her bir b bit açık metin bloğuna karşılık gelen bir tek şifreli metin bloğu olacağından her bir b bitlik açık metne karşılık gelen b bitlik şifreli metin ikililerinden oluşan çok büyük bir kod kitabı akla gelebilir (Stallings, 2011). Aynı b bit uzunluğundaki açık metin her seferinde aynı b bit şifreli metni oluşturacağından yeterince uzun bir açık metin aynı kelimeyi yani b bit uzunluğundaki açık metin parçasını birden fazla defa içerebileceğinden bu da bu kelimenin tahmin edilebilme yani şifrenin kırılma ihtimalini arttırmaktadır.

CBC: ECB'nin bu zayıflığını ortadan kaldırmak için aynı açık metin parçasının birden fazla tekrar ettiği dumlarda her seferinde farklı şifreli metinler oluşturmak üzere bu mod ortaya konmuştur. Bu modda her bir her bir açık metin bloğu bir önce elde edilmiş şifreli metin bloğu ile karşılıklı dışlandıktan (zincir ilşkisi) sonra şifreleme algoritmasına verilmektedir. Bu sebeple şifrelencek açık metin blokları aynı olsa bile şifrelemeden önce bir önceki şifreli metin ile karşılıkı dışlandığı için her seferinde farklı şifreli metinler elde edilecektir. Fakat bu modun da bir zayıf yanı bir şifreli metin oluşturlması sırasında kanal hatalarından ya da işlem hatalarından dolayı bir bitlik bile veri hatası kullanılan zincir ilşkisinden dolayı son bloklara doğru büyüyerek aktarılır ve sonuçta bir bitlik gerçek hata modun çıkışında çok daha fazla bitte hataya neden olmaktadır.

CTR: CBC'deki hata propagasyonunu elimine etmektedir. Açık metin blok uzunluğuna eş uzunlukta sayaç kuallnılmaktadır ve her bir sayaç değeri şifrelenecek olan her bir açık metin bloğu için farklı olmalıdır. Bunun için sayaç belirli bir değere atanır ve her bir açık metin bloğu için bir arttırılır. Şifreleme için sayaç değeri şifrelenerek oluşan veri ile açık metin bloğu karşılıklı dışlanır. Bu yöntem hem paralel çalışmaya imkan sağlayarak etkinliği arttır ve bir bitlik hata zincir ilişkisi kurlmadığı için yine bir bit olarak çıkışa yansır.

3.2 Asimetrik Şifreleme ile Veri Güvenliği

Açık anahtar şifrelemesi olarak ta bilinmektedir. 1976 yılında Diffie ve Hellman tarafından önerilmiştir. Asimetrik şifreleme kriptanalitik ataklara karşı simetrik şifrelemeden daha kuvvetli olduğu doğru olmayıp bu kuvvetlilik kullanılan anahtar uzunluğu ve şifre kırmak için yapılan hesaplama işi ile alaklıdır (Stallings, 2011).

Asimterik şifreleme daha çok mesaj kökeni doğrulama ve anahtar dağıtımı (simterik şifreleme anahtarı) için kullanılmaktadır. Açık anahtar şifreleme altı temel unsurdan oluşmaktadır. Açık metin, şifrelenmek üzere algoritmaya giriş olarak verilen okunabilir orijinal metindir. Şifreleme algoritması, açık metin üzerine bir takım dönüşümler uygulanmasını sağlamaktadır. Açık ve özel anahtarlar, bu anahtar çifiti şifreleme ve şifre çözme işlemlerinde kullanılmaktadır. Bunlardan biri şifreleme için kullanılırsa diğeri de şifre çözmek için kullanılmaktadır. Şifreli metin, şifreleme algoritması çıkışında oluşturulan değiştirlmiş veya dönüştürülmüş metindir. Şifre çözme algoritması, şifreli metin ve gerekli anahtar girişe uygulandığında orijinal metni oluşturan alogitmadır.

Adından da anlaşılacağı gibi açık anahtar herkese açıktır ve üzerinde herhangi bir koruma yoktur fakat özel anahtar her zaman saklı tutulmaktadır.

Bu şifreleme yönteminin genel olarak işleyişi aşağıdaki gibi verilebilmetedir.

Mesajlaşacak olan her iki kişi de şifreleme ve şifre çözme işlemlerinde kullanmak üzere kendi açık ve özel anahtar çiftini üretir. Daha sonra her iki kişi de açık anahtarlarını diğerinin erişebileceği bir alana yerleştirirler. Eğer A B'ye mesaj göndercek ise A B'nin açık anahtarını A'nın koyduğu genele açık olan alandan alarak bu anahtar ile açık anahtar şifreleme algoritmalarından birini kullanarak göndemek istediği mesajı şifreler ve B'ye gönderir. B gelen şifreli mesajı okuyabilmek için önce kendi özel anahtarı ile şifre çözme işlemi yapar ve daha sonra orijinal açık metne ulaşabilir. B'nin özel anahtarı sadece B'de bulunmakta olduğundan A'nın B'ye B'nin açık anahtarı ile şifreleyerek gönderdiği mesajı B'den başkası açamayacak ve gizli bilgiye ulaşamayacaktır.

Asimetrik şifrelemede özel anahtar dağıtımı veya paylaşımı olmadığı için taraflar özel anahtarlarının gizliliğini koruyabildikleri sürece taraflar arası haberleşme güvenli olacaktır. Özel anahtar gizliliğini yitirdiği anda bu anahtar yeni üretilen bir gizli anhatrla ve bunun karşıklık gelen açık anahtarı da eski açık anahtarla değiştirilebilmetedir.

Açık anahtar şifreleme algoritmalarının taşımak zorunda olduğu bir takım gereksinimler vardır. Diffie ve Hellman tarafından ortaya konulan bu gereksinimler aşağıdaki gibi özetlenebilmektedir.

Açık ve özel anahtar çifti oluşturmak hesaplama yönünden kolay olmalıdır.

Mesaj gönderici tarafta karşı tarafın açık anahtarına erişim ve bu anahtar ile metni şifrelemek hesaplama yönünden kolay olmalıdır.

Mesajı alan taraf için kendi özel anahtarı ile mesajın şifresini çözmek hesaplama yönünden kolay olmalıdır.

Bir düşman için aldığı bir açık anahtar için bu anahtara karşılık geln özel anahatrın hesaplanması etkin ve kolay olmamalıdır.

Bir düşman için bildiği bir açık anahtarla şifrelenmiş olan bir metinden açık metni elde etmek kolay olmamlıdır.

Her açık anahtar şifreleme yöntemi için geçerli olmak zordunda olmasa da imzalamaişlemlerinde kullanılabilecek olan algoritmalar aşağıdaki özelliğe de sahip olmalıdırlar (Stallings, 2011).

Açık ve özel anahtar çiftinden herhangi birisi şifreleme diğeri de şifre çözmek için kullanılailmeli ve anahtarların biriyle şifrelenen veri diğerinin kullanımıyla yine açık metin olarak elde edilebilmelidir.

3.2.1 RSA kriptosistemi

RSA kriptosistemi Rivest, Shamir ve Adelman tarafından 1977'de geliştirilmiş olupilk açık anahatar kriptosistemlerinden biri olmakla birlikte en çok gerçeklenenlerinden de biridir (Stallings, 2011). RSA bir blok şifreleyici olup açık metin ve şifreli karşılığı bir n sayısı için her ikiside 0 ile n-1 arasında bir tamsayıdır.

RSA' e göre P açık metni ve C şifreli metni için şifreleme ve şifre çözme işlemleri aşağıda gösterildiği gerçekleşmektedir.

Şifreleme algoritmasının işleyişi aşağıdaki gibi özetlenebilmektedir.

p ve q iki büyük asal sayı (tipik uzunlukları 512 bit) olarak seçilir.

$$C = P^e \bmod n. \qquad \text{(e, n çifti açık anahtar)} \qquad \textbf{(3.1)}$$

$$P = Cd \bmod n. \qquad \text{(d, n çifti gizli anahtar)} \qquad \textbf{(3.2)}$$

e x d = 1 mod z olan e bulunur (Genişletilmiş Euclid algoritması ile).

Şifrelenecek her P bloğu için $0 \leq P < n$ olmalıdır.

n = p x q ve z = (p – 1) x (q – 1) hesaplanır.

z ile en büyük ortak böleni 1 olan bir d sayısı (bağıl asallık) bulunur (Euclid algoritması ile).

Yöntemin gücü çok büyük n sayısını asal çarpanlarına ayırmanın bilinen sistematik bir yolun olmamasından gelmektedir.

3.2.2 Diffie-Hellman anahtar değişim protokolü

Diffie ve Hellman tarafından ortaya konulan algoritmanın amacı iki varlık arasında gizli anahtar paylaşılmasını sağlamaktır. Bu sayede paylaşılan gizli anahtar ile taraflar bir birlerine gönderecekleri bilgiyi simetrik şifreleme ile şifreleyerek güvenle bir birlerine gönderebileceklerdir.

Algoritmanın gücü ve etkinliği ayrık logaritmaların hesaplanma zorluğundan gelmektedir. Algoritmaya göre genel olarak bilinen açık sayılar bir asal sayı olan q ve bunun ilkel kökü olan α'dır. Haberleşmek isteyen A ve B varlıkları için, A $X_A < q$ olmak üzere bir rastgele sayı seçer ve $Y_A = \alpha^{X_A} \text{mod} q$ değerini hesaplar. Aynı şekilde B'de $X_B < q$ olmak üzere bir rastgele sayı seçer ve $Y_B = \alpha^{X_B} \text{mod} q$ değerini hesaplar. Her iki tarafta X değerlerini kendine özel bir şekilde gizli tutar veY değerlerini bir birlerine gönderirler. A varlığı paylaşılan gizli anahtarı $K = {Y_B}^{X_A} \text{mod} q$ ve B varlığıda paylaşılan gizli nahtarı $K = {Y_A}^{X_B} \text{mod} q$ hesaplamaları ile elde ederler.

Tek başına Diffie-Hellman anahtar değişim algoritması ortadaki adam saldırılarına karşı savunmasızdır. Bu olumsuzluğu ortan kaldırmak için karşıklı varlıkların kimlik doğrulamasının sayısal imzalarla yapılması yeterli olmaktadır.

3.2.3 Sayısal imza

Sayısal imza yöntemi açık anahtar şifreleme yönteminin başka bir açıdan uygulamasının bir sonucudur. İki taraf arasında gönderilecek olan bir mesajın gizliliğinden çok gerçekten de bu mesaj oluğunu idda eden varlık tarafından mı gönderiliyor belirlemek için kullanılmaktadır. Burada bir A varlığı B varlığına göndereceği mesajı kendi özel anahtarı ile şifreleyerek göderir ve B varlığı A varlığının bilinen açık anahtarı ile bu mseajın şifresini çözer eğer şifre gerçekten de A varlığının açık anahtarı ile çözülebiliyorsa ve A varlığı özel anahtarı gizliliğini koruyorsa bu mesaj kesinlikle A varlığının özel anahtarı ile şifrelenmiştir demektir. Burada özel anahtar ile şifrelenen metnin tamamı sayısal imza olarak

kullanılmaktadır. Şifrelenecek veri çok büyük olduğu durumda veriyi şifrelemek yerine verinin bir sonraki bölümde anlatıldığı gibi özeti alınarak bu özet şifrelenerek hesaplama yönünden önemli bir avantaj sağlanmış olunur.

3.3 Kriptografik Güvenli Özet Fonkiyonları

Güvenli veya başka bir deyişle tek yönlü özet fonksiyonları sayısal imza, veri bütünlüğü kontrolü ve mesaj kökeni doğrulama işlemlerinde yaygın olarak kullanılmaktadır çünkü işlem yoğunluğu yüksek matematiksel foksiyonların büyük veiler üzerine uygulanması ciddi performans sorunları doğurmaktadır. Bu sorunları ortadan kaldırmak için kimlik doğrulama, bütünlük kontrolü gibi işlemler verinin özeti alındıktan sonra orijinal veriye göre daha kısa olan özet üzeride daha yüksek performans ile gerçeklenebilmektedir. Fakat, yukarda sayılan güvenlik işlemlerinde kullanılabilmesi için bir özet alma fonksiyonun çıktısının gerçekten de üzerine uygulandığı veri yalnızca ve yalnızca kendisinin temsil etmesi yani güvenilir olması gerekmektedir. Bunun için bir özet fonksiyonun sağlamak zorunda olduğu koşular aşağıda sıralanmaktadır. H özet alma fonksiyonu olmak üzere,

H herhangi bir uzunluktaki veriye uygulanabilmelidir.

H sabit uzunluklu bir çıkış üretmelidir.

Herhangi bir x girişi için H(x) kolaylıkla hesaplanabilmelidir.

Herhangi bir H(x) = h için h çıkışından x girişi elde edilememelidir. Bu özelliğe tek yön özelliği veya ön görüntü dayanıklılığı denilmektedir.

Herhangi bir x için H(x) = H(y) olacak şekilde bir $x \neq y$ olmak şartıyla bir y hesaplanabilmesi çok zor olmalıdır. Bu özelliğe ikici ön görüntü dayanıklılık özelliği veya zayıf çakışma dayanıklılığı özelliği denilmektedir.

H(x) = H(y) olacak şekilde bir (x,y) ikilisi hesaplanması mümkün olmamalıdır. Bu özellik güçlü çakışma dayanıklılığı olarak bilinmektedir.

3.3.1 SHA özet fonksiyonu

Son zamanlarda en sık kullanılmakta olan özet fonkiyonu SHA (Secure Hash Algorithm) olarak kaşımıza çıkmaktadır (Stallings, 2011). Çünkü diğer özet fonkiyonu ailelerinin bir takım zayıflıkları bulunarak açıklanmıştır. SHA NIST tarafından ortay konulmuş olup ilk versiyonu SHA-0 1993' standartlaştırılmıştır. Bu

algoritmadaki bir takım zayıflıların giderilmesi amacıyla SHA-1 1995'te ortaya konulmuştur (Stallings, 2011). SAH-1 algoritması 160 bit uzunlunğunda bir çıkış üretmektedir.

3.4 Sayısal Sertifikalar ve Açık Anahtar Altyapısı

3.4.1 Sayısal sertifika

Sayısal sertifikalar güvenilir sertifika ototriteleri tarafından kişiler veya daha genel olarak varlıklar adına düzenlenmektedir. Bu sertifikalar sertifika oteritesiveya kişinin kendisi tarafından belirli dizinlere yerleştirilir. Dizin sunucusu sertifika üretmekten sorumlu olmayıp sadce kolay erişilebilir bir depo görevi görmektedir. Bir X.509 sayısal sertifikası versiyon, seri numarası, imza algoritması tanımlayıcısı, düzenleyen ismi, geçerlilik süresi, adına düzenlenen varlık ismi, adına düzenlenenin açık anahtar bilgisi, dünzeleyen tekil tanımlayıcısı, adına düznlenen tekil tanımlayıcısı, uzantılar ve imza bölümlerinden oluşmaktadır. En sondaki imza sertifika oteritesinin sertifikanın geriye kalan tüm kısmının özeti üzerine attığı sayısal imzasıdır. Sertifikalar bu düzenleyen oterite sayısal imzası sayesinde taklit edilemez olmakta olup güvenle açık dizin sunucularına yerleştirilebilmektedir. Süresi dolduğu için, sertifikada bulunan açık anahtara karşılık gelen özel anahtarın gizliliğini kaybetmesinden dolayı veya sertifikanın kötüye kullanıldığının belilenmesi gibi sebeplerden dolayı sertifikalar iptal edilebilmektedir ve iptal edilen sertifikalar sertifika iptal listelerine eklenmekte ve bu listelerde yine düzenleyen sayısal imzası eklenmiş bir şekilde yine ilgili dizin sunucularına yerleştrilmektedir.

3.4.2 Açık anahtar altyapısı

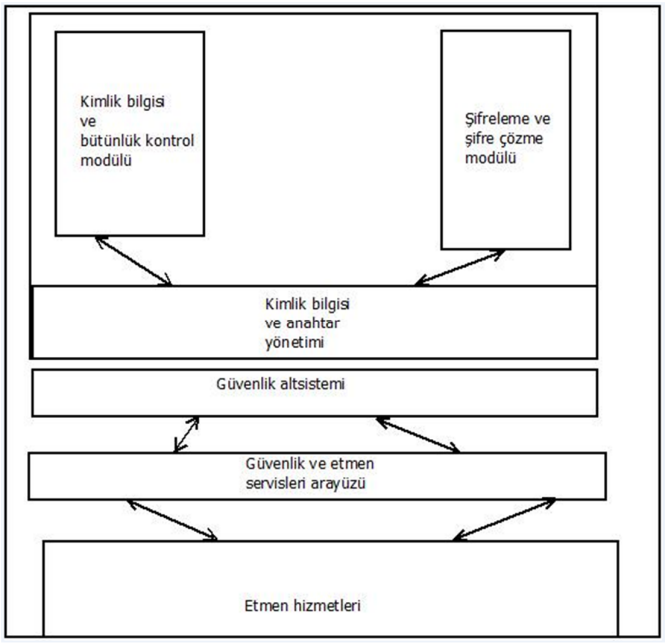
Açık anahtar altyapısı açık anahtar bilgisi içeren sayısal sertifikaların çevirim dışı oluşturulup dağıtılması ile anahtar dağıtım merkezi gibi çözümlerin oluşturduğu dar boğazı ortadan kaldırmaktadır. Altypaıya göre kök sertifika oteriteleri bu oteritelerin yetkilendirdiği bölgesel oteriteler bulunmaktadır. Burada kök oteriteler bölgesel oteritelere sertifika düzenlemekte olup bölgesel oteriteler belki yatayda başka bölgesel oteriteleri veya dikeyde kullanıcıları sertifikalandırmaktadırlar. Bir bölgesel oterite tarafından sertifikalandırılan bir kullanıcının sertifikasının doğru ve güvenilirliğinin tespiti için ilgili bölgesel oteriteyi sertifikalandıran diğer bölgesel oteritelerin ve kök oteritenin de sertifika bilgilerinin edinilmesi gerekmektedir kökten

uca bu sertifika takımına güven zinciri denilmektedir ve her bir sertifikanın imzası bir üst seviyedeki oteritenin sertifika bilgisiyle doğrulanmaktadır. Bu sertifikalar çevirim dışı oluşturulup dizin sunucularına yerleştirildikleri için anahtar (açık anahtar) dağıtımı etkin bir şekilde gerçekleştirilmektedir.

4. KONU TABANLI ETMEN GÖÇÜ İÇİN GÜVENLİK YAPISI

4.1 Güvenlik Mimarisi

Bölüm 2.5'te anlatılan konu tabanlı olarak göç eden etmen sistemlerinin güvenliği haberleşme güvenliği temel prensipleri olan kimlik doğrulama, veri gizliliği ve veri bütünlüğü genel ilkelerinin karşılanması ile sağlanmaya çalışılmıştır. Bu çalışmada yukarda sayılan genel güvenlik ihtiyaçlarının sağlamak amacıyla hareketli etmenlerin hareketleri boyunca kendileri ile birlikte taşıyarak gidecekleri her yere kendileri ile birlikte götürebilecekleri bir güvenlik mimarisi önerilmektedir. Güvenlik işlevlerine hareketlilik kazandırılabilmesi için tüm fonksiyonalite etmen içerisine gömülmüştür. Bu özellik güvenlik bağlamında etmenlerin üzerinde çalıştıkları platform bağımlılığını azaltarak platformlar arası etmen göçünün desteklenmesini sağlamaktadır. Önerilen mimari kimlik doğrulama ve bütünlük kotrolü, şifreleme ve kimlik bilgisi ve ahatar yönetimi olmak üzere her biri diğerine gevşek bağlı üç modülden oluşmaktadır. Etmen içerisine gömülmüş güvenlik mimarisi Şekil 4.1'de görülmektedir.



Şekil 4.1 : Güvenli etmen mimarisi.

4.1.1 Kimlik doğrulama ve bütünlük kontrol modülü

Bu modül göç edecek etmen için bütünlüğü ispatlayan bütünlük kodunun ve kimlik bilgilerinin oluşturulması ve bu bilgilerin göç edecek olan etmene eklenmesinden sorumlu iken, ziyaret edilecek olan konak üzerinde çalışmakta olan ve bu konağa etmen girişini kontrol eden yönetici etmen tarafında ise kimlik biligierinin ve bütünlük verisinin doğruluğunun kontrol edilmesinden sorumludur. Bu kimlik doğrulama ve bütünlük verilerinin oluşturulması ve kontrollerinin yapılabilmesi için sayısal imza X.509 saysal sertifika ve açık anahtar alt yapısı ile birlikte kullanılmaktadır. Sayısal imza yöntemi Bölüm 3.2.3'te anlatılan özelliklerinden dolayı araya girme, veri değiştirme ve ekleme tipi aktif saldıralara karşı oldukça dayanıklı olması ve inkar edilememezlik gereksiniminde de kullanılabilmesi sebebi ile ve açık anahtar altyapısı ve sayısal sertifikalar da Bölüm 3.4'te anlatılan anahtar dağıtımı özelliklerinden yararlanabilmek amacıyla bu modülde kullanılmaktadır. Veri bütünlüğü kodu oluşturmak ve kontrolünü yapabilmek amacıyla Bölüm 3.3'te anlatılan özelliklerinden yararlanmak amacıyla kriptografik güvenli özet fonksiyonları kullanılmaktadır. Bu modülde sayısal imza ve kriptografik güvenli çırpı fonksiyonlarının kullanımı ile önerilen mimarinin karşılamakla yükümlü olduğu üç temel güvenlik gereksinimi olan gelen hareketli etmen için kimlik doğrulama, bütünlük ve inkar edilememezlik ihtiyaçları karşılanmaktadır.

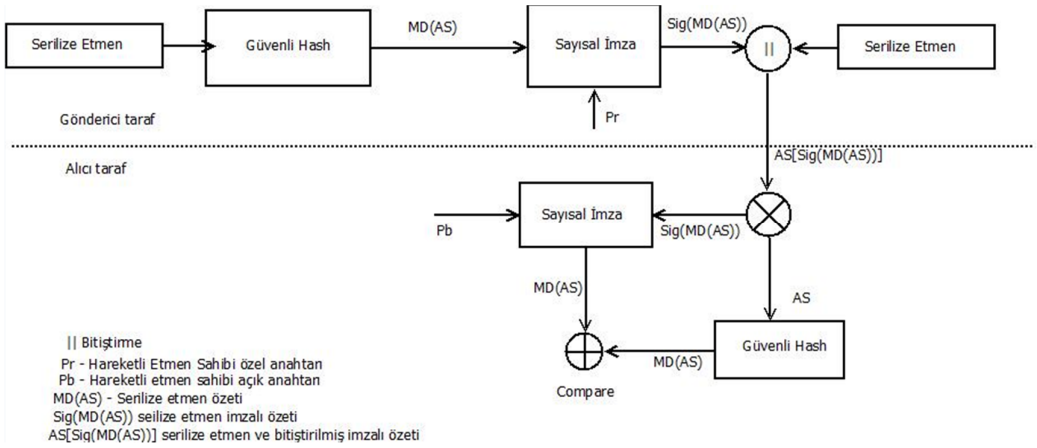
Kimlik ve bütünlük işlemleri göç edecek olan yayıncı etmen tarafında etmenin kendisini serilize etmesini takiben bu serilize etmen ve etmen sahibi kimlik bilgisinin ard arda eklenmesi ile oluşan yapının kriptografik güvenli bir çırpı fonkiyonu olan SHA-1 algoritmasından geçirilip özetinin oluşturması ile başlamaktadır. Daha sonra çırpı fonksiyonu çıktısı olan serilize etmen özeti RSA açık anahtar şifreleme yöntemi kullanılarak yayıncı etmenin sahibinin (yayıncı etmenin adına hareket ettiği varlık) gizli anahtarı ile yine yayıncı etmenin kendisi tarafından imzalanır ve oluşturulan imzalı özet serilize etmenin sonuna eklenerek yayıncı etmen göçe hazır hale gelmiş olur. Artık mesajlaşma ara katmanına sadece serilize yayıncı etmen değil serilize yayınıcı etmenin sonuna etmen sahibi kimlik bilgisi ve imzalı serilize etmen özeti eklenmiş hali gönderilmektedir. Oluşturulan imzalı özet artık hareket halindeki yayıncı etmenin veya sahibine ait kimlik bilgisinin değiştirilmesine veya yerine geçilmesine yönelik aktif saldıralara karşı bir koruma ortaya koymaktadır çünkü eğer etmen yolda değiştirilse bile etmenin sonuna eklenmiş olan güvenlik verisi

değiştirilemeyecektir çünkü bu veri imzalıdır ve Bölüm 3.2.3'te anlatıldığı gibi bu imzanın oluşturulmasında kullanılan gizli anahtar açığa çıkartılmadıkça bu imza taklit edilemezdir. Yönetici etmen yani ziyaret edilen konak tarafında ise öncelikle gelen hareketli etmen henüz serilize haldeyken kendisine eklenmiş olan kimlik doğrulama bilgisi olan sayısal imza yine RSA açık anahtar şifreleme yönteminin yayıncı etmen sahibine ait açık anahtarı ile işleme tabi tutularak veri kaynağı doğrulama işlemi gerçekleştirilmektedir. Gelen yayıncı etmen sahibi kimlik bilgisi serilize etmen verisi sonuna eklendiği için buradan alınmaktadır ve bu kimliğe ait açık anahtarı barındıran X.509 sertifikası önce yönetici etmen yerel deposunda aranmaktadır eğer bulunabilir ise bu açık anahtar kullanılır aksi takdirde ilgili sertifika Bölüm 3.4'te anlatıldığı gibi açık anahtar alt yapısı kullanılarak elde edilip yönetici etmen yerel deposunda saklanmaktadır. Eğer kimlik doğrulama işlemi başarısız olursa gelen etmen doğrudan ihmal edilerek zararlı aktiviteler gerçekleştirecek olması ihtimaline karşılık kendisine hiç çalışma imkanı verilmemektedir. Kimlik doğrulamasının başarılı olması durumunda ise imzanın açılmasıyla edilen doğruluk ispatlı serilize etmen özetinin serilize etmen olarak gönderilen verinin yine SAH-1 özeti alınarak oluşturulan özeti ile karşılaştırılması yapılmaktadır. Eğer her iki özet aynı ise gelen yayıncı haraketli etmenin bütünlüğü de ispatlanmış olmaktadır. Eğer bütünlük kontrol işlemi başarısız yani imzanın açılmasıyla elde edilen özet ile serilize etmenden elde edilen özet aynı değilse etmen henüz bu aşamada daha hiç çalıştırılma fırsatı yakalayamadan ihmal edilerek gerçekleştirebileceği zararlı faaliyetlere karşı koruma sağlanmış olmaktadır. Bu şekilde yönetici etmen gayet erken bir safhada genel geçer güven bilgisini kullanarak platformunu ziyaret eden zararlı etmenlerin aktif saldırılara karşı dayanıklı bir hale getirmektedir.

Hem yayıncı etmen hem de yönetici etmen taraflarında kimlikdoğrulama ve bütünlük kontrolü işlemlerinin akışı Şekil 4.2'de görülmektedir.

4.1.2 Şifreleme ve şifre çözme modülü

Bu modül göç eden yayıncı etmeninin kendisiyle birlikte taşıdığı ya da ziyaret ettiği konak (platform) üzerinde çalışması sırasında ürettiği önemli ve hasas veriyi aynıkonak üzerinde çalışmakta olan diğer (zararlı) hareketli yada sabit etmenlerin ya da (zararlı) konağın kendisinin gerçekleştireceği dinleme/izleme gibi pasif saldırılara

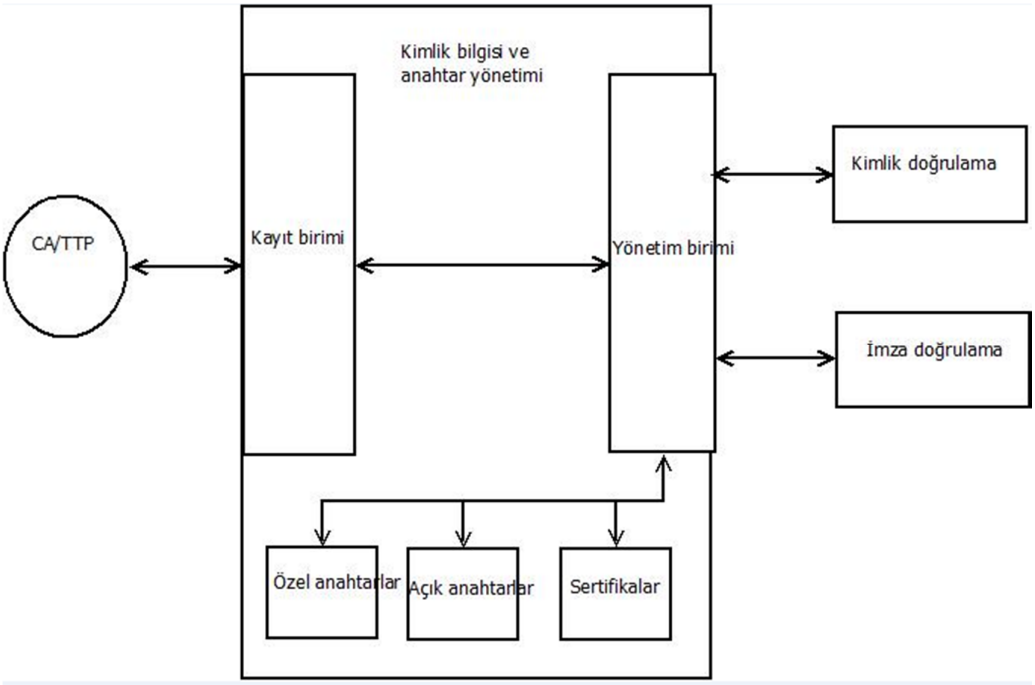


Şekil 4.2 : Kimlik doğrulama ve bütünlük kontrol operasyonları.

karşı korumaktan sorumludur. Bu modül tarafından sağlanan koruma uygulama katmanında etmen seviyesindedir. Şifreleme modülü veri şifreleme işlemlerinde Bölüm 3.1.1'de anlatılan simetrik blok şifreleme yöntemlerinden AES algoritmasını kullanmaktadır. Bu yöntemin kullanılmasının başlıca sebebi ilgili bölümde de anlatıldığı gibi simetrik şifrelemenin asimetrik şifrelemeye göre ve AES'in de diğer simetrik blok şifreleme yöntemlerine göre hesaplanma performansının daha yüksek olmasıdır. Bu özelliğinin yanı sıra simetrik şifrelemenin en önemli problemlerinden biri olan anahtar paylaşımı işinin göç eden etmenlerin bilginin uzmanı yayıncı etmenler olması ve bu nedenle diğer etmenler ile haberleşme ihtiyaçlarının en az olacağı için anahtar paylaşımı yapma ihtiyacı da en az olacaktır ki bu durum da yeteri kadar güçlü simetrik blok şifreleme yöntemlerinin burada kullanılmasını uygun kılmaktadır. Eğer göç eden yayıncı etmen ürettiği ya da taşıdığı hassas veri üzerinden başka etmenler ile haberleşmek durumunda kalırsa burada haberleşeceği etmene şifrelemede kullanacağı geçici simetrik anahtarı göndermesi gerekmektedir. Bu anahtar paylaşımının güvenle yapılabilmesi için Bölüm 3.2.2'de detayları anlatılan Diffie-Hellman anahtar değişim protokolü araya girme ataklarına karşı dayanlıklı olabilmesi açısndan kimlik doğrulama mekanizmalarıyla birlikte kullanılmaktadır. Göç eden yayıncı etmen haberleşeceği her bir diğer etmenle haberleşmede yeni geçici simetrik anhatar kullanmaktadır ve kendi üretteği yada taşıdığı veriyi saklamak üzere kullandığı simetrik anahtarı kimseyle paylaşmamaktadır.

4.1.3 Kimlik bilgisi ve anahtar yönetim modülü

Bu modül temel güvenlik gereksinimleri olan kimlik doğrulama, gizlilik, inkar edilemezlik ve erişim kontrolü işlevlerinin hiç birini doğrudan karşılamamaktadır. Modülün görevi bu sıralanan gereksinimlerin karşılanabilmesi için önceki bölümlerde anlatılan modüllerin kriptografik işlemlerinde kullanılacak olan anahtarların ve kimlik bilgilerinin yönetimini sağlamaktır. Anahtar ve kimlik bilgisinin kontorlünün ayrı bir modül tarafından gerçeklenmesi diğer modüllerin bir birlerine olan bağımlılıklarını azaltarak modüllerin bir birlerinin değişimlerinden etkilenmelerini en aza indirmek amaçlanmaktadır. Modüle ait mimari Şekil 4.3'te gösterilmektedir.



Şekil 4.3 : Kimlik bilgisi ve anahtar yönetimi modülü.

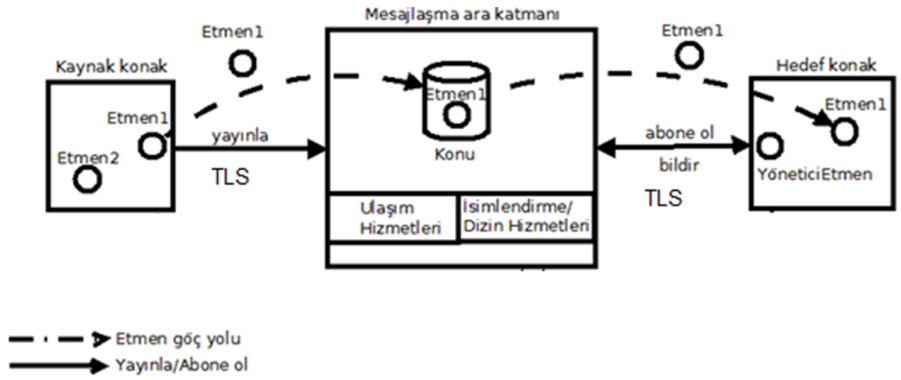
Kayıt ve yönetim birimleri modülü meydana getiren temel birimlerdir ve iki temel görev bu iki birime yüklenerek görevlerin ayrılığı ile modül kendi içerisinde de değişimlere karşı korunaklı bir hale getirilmektedir. Kayıt birimi modülün dış dünyaya yani etmen platform dışına açılan kapısı olmaktadır. Bu nedenle sertifika otoritesi (CA) ile tüm etkileşim kayıt biriminin sorumluluğundadır. Sertifika otoritesi ile etkileşim açık yeni anahtar sertifikası isteğini, var olan bir sertifikanın yenilenmesi isteğini, sertifika iptal isteğini, kapsamaktadır. Kayıt birimi sertifika otoritesinin yanında yine dış dünyada LDAP sunucuları ile de temas halindedir.

Modül bu teması kimlik doğrulama modülünde kullanılmak üzere göç eden yayıncı etmenin sahibine ait sayısal sertifika bilgisini alabilmek için Bölüm 3.4'te anlatılan açık anahtar alt yapısı uyarınca kurmaktadır. Sertifika otoritesi ile anahtar yönetim modülü arasındaki güvenliği kurabilmek için yine TLS kullanılmaktadır. Yönetim birimi ise modülün iç işlerini yürütmekten ve güvenlik mimarisini oluşturan diğer modüller ile haberleşmekten sorumludur. Simetrik şifrelemede kullanılacak olan anahtarların üretilmesi, saklanması, şifreleme modülüne iletilmesi, sayısal sertifikaların saklanması ve kimlik doğrulama modülüne iletilmesi yönetim biriminin başlıca görevleridir. Bu modülün kotrolünde olan ve etmenle birlikte haraket eden asimetrik şifreleme gizli anahtarı etmen sahibine ait olan ve kimlik doğrulama verisinin oluşturulmasında kullanılan gizli anahtar değildir. Buradaki, geçici olarak üretilmiş ve yine geçici olan simetrik şifreleme anahtarı paylaşımında kullanılacak olan asimetrik gizli anahtardır. Etmen sahibine ait asimetrik gizli anahtar sadece yönetici etmen anahtar yönetim modülünde bulunmaktadır.

4.1.4 Taşıma seviyesi güvenlik

Bu çalışmada taşınım seviyesinde güvenliği tesis edebilmek için TLS (Transport Layer Security) kullanılmaktadır. Bu sayede kaynak konak ile mesajlaşma ara katmanı ve mesajlaşma ara katmanı ile hedef konak arasında hareket halinde olan etmenin aktif ve pasif ataklara karşı korunması sağlanmakla birlikte haberleşmede yer alan herhangi bir varlığın, kaynak konak, hedef konak ve mesajlaşma ara katmanı, yerine geçme ve araya girme saldırılarına karşı güvenliği sağlanmaktadır. TLS araya girme saldırılarına karşı haberleşemekte olan varlıklar için X.509 sayısal setifikalarının açık anahtar alt yapısı ile birlikte kullanımı ile karşılıklı kimlik doğrulama işlemi sağlamakta iken hareket halindeki etmenin pasif ataklara karşı korunmasını ise kimlikleri doğrulanmış iki varlık arasındaki tüm veri trafiğinin simetrik bir yöntem ile şifrelenmesi ile sağlamaktadır (Dierks, T. ve Rescorla, E.), (Dierks, T. ve Allen, C).

Bölüm 2.5'te anlatılan yayınla abone ol modeli ile konu tabanlı etmen göçü mimarisine TLS entegrasyonu Şekil 4.4'te görülmektedir.



Şekil 4.4 : TLS korumalı etmen göçü (Karzan, Erdoğan,2013)'ten uyarlanmıştır.

4.2 Güvenlik Mimarisinin Gerçeklenmesi

Bu çalışmada önerilen güvenlik mimarisi Bölüm 2.5'te anlatılan yayınla/abone ol modeli ile konu tabanlı etmen göçü yapısının güvenliğini sağlamak amacıyla ortaya konmaktadır. Dolayısıyla ilgili etmen göçü modelinin gerçeklendiği dil olan JAVA dili güvenlik mimarisinin de gerçeklenmesinde kullanılmıştır. Yine aynı nedenle etmen platformu olarak JADE kullanılmıştır. Güvenlik fonkisyonlarının gerçeklenmesi için JCE (Java Cryptogrphic Extensions) kütüphanesi kripto sağlayıcı olarak kullanılmıştır. Kimlik ve bütünlük doğrulama modülünde sayısal imzanın oluşturulması ve gelen imzanın kontrol edilmesinde JCE tarafından sağlanan sayısal imza sınıf seti mesaj özeti ve imzalama işlemlerinin bir arada uygulanmasına imkan veren "SHA1withRSA" algoritması ile kullanılmıştır. Şifreleme ve şifre çözme modülünde yine JCE tarafından sağlanan simetrik şifreleme sınıf seti "AES192" algoritması ile kullanılarak üretilen hassas verinin şifrelenerek dinleme gibi pasif ataklardan korunması sağlanmıştır. Simetrik anahtarın paylaşılması için kullanılan Diffie-Hellman algoritması yine JCE tarafından sağlanan algoritma parametreleri yardımcı sınıfları kullanılarak gerçeklenmiştir. Kimlik bilgisi ve anahtar yönetim modülü kendi içinde modüler olarak tasarlanmış olup yönetim ve kayıt birimleri birer JAVA sınıfı olarak gerçeklenmiş olup simetrik ve asimetrik şifreleme anahtarları ve açık anahtar sertfikaları parola korumalı anahtar depolarında yine parola korumalı olarak saklanmaktadır.

Yukarıda genel olarak verilen gerçekleme detayları aşağıdaki gibidir.

Kimlik doğrulama modülünde kendini serilize eden yayıncı etmen serilize halinin sonuna kendisinin SHA-1özetini RSA imzalı bir şekilde ekler. Bunun için öncelikle RSA özel anahtarı anahtar deposundan alınır ve sonra özet üzerine imzalama gerçeklenir. Anahtarın depodan alınması işlemi:

```
KeyStore ks = KeyStore.getInstance("JKS");
FileInputStream ksfis = new FileInputStream("C:\\keystore.jks");
BufferedInputStream ksbufin = new BufferedInputStream(ksfis);
```

Şeklinde gerçeklenebilmektedir. Özet alma ve imzalama işlemi aşağıda gösterildiği gibi yapılabilmektedir:

```
Signature rsa = Signature.getInstance("SHA1withRSA", "SunRsaSign");
rsa.initSign(priv);
rsa.update(new sun.misc.BASE64Decoder().decodeBuffer(payload));
byte[] signatureData = rsa.sign();
String asciEncodedSignature = new BASE64Encoder().encode(signatureData);
String payloadSignatureAdded = payload+";"+ asciEncodedSignature;
```

Yönetici etmen tarafında ilgili sertifikanın depodan alınması işlemi aşağıda gösterildiği gibi yapılabilir:

```
KeyStore ks = KeyStore.getInstance("JKS");
FileInputStream ksfis = new FileInputStream("C: \\keystore.jks");
BufferedInputStream ksbufin = new BufferedInputStream(ksfis);
ks.load(ksbufin, "123456".toCharArray());
Certificate cert = (Certificate) ks.getCertificate("ali");
```

Sayısal imzanın doğrulanması işlemi ise aşagıdaki gibi yapılabilmektedir:

```
Signature sig = Signature.getInstance("SHA1withRSA", "SunRsaSign");
sig.initVerify(cert);
sig.update(new sun.misc.BASE64Decoder().decodeBuffer(payloadParts[0]));
boolean verifies = sig.verify(new
sun.misc.BASE64Decoder().decodeBuffer(signature));
```

Veri gizliliğinin sağlanması amacıyla AES simetrik şifrelemesi JCE kütüphanesi kullanılarak aşağıdaki gibi gerçeklenmesi mümkündür.

```
char[] password = "123456".toCharArray();
```

```
java.io.FileInputStream fis =new
java.io.FileInputStream("C:\\symmetricKeyStore.jceks");
ks.load(fis, password);
fis.close();
SecretKey key = (SecretKey)ks.getKey("secretKeyAlias", "123456".toCharArray());
Cipher c = Cipher.getInstance("AES");
SecretKeySpec k = new SecretKeySpec(key.getEncoded(), "AES");
c.init(Cipher.ENCRYPT_MODE, k);
byte[] encryptedData = c.doFinal(dataToSend.getBytes());
```

Gizli anahtarın paylaşımında kullanılan Diffie-Hellman anahtar değişim algoritmasının yayıncı etmen tarafında JCE kütüphanesi fonksiyonlarıyla gerçeklenmesi aşağıdaki gibi yapılabilmektedir.

```
KeyPairGenerator yayıncıKpairGen = KeyPairGenerator.getInstance("DH");
yayıncıKpairGen.initialize(dhSkipParamSpec);
KeyPair yayıncıKpair = yayıncıKpairGen.generateKeyPair();
// Yayıncı etmen kendi DH KeyAgreement nesnesini oluşturur
KeyAgreement yayıncıKeyAgree = KeyAgreement.getInstance("DH");
yayıncıKeyAgree.init(yayıncıKpair.getPrivate());
// Yayıncı etmen kendi açık anahtarını Yönetici etmene verir.
byte[] yayıncıPubKeyEnc = yayıncıKpair.getPublic().getEncoded();
byte[] yoneticiPubKeyEnc = {};
KeyFactory yayıncıKeyFac = KeyFactory.getInstance("DH");
X509EncodedKeySpec x509KeySpec = new
X509EncodedKeySpec(yoneticiPubKeyEnc);
PublicKey yoneticiPubKey = yayıncıKeyFac.generatePublic(x509KeySpec);
yayıncıKeyAgree.doPhase(yoneticiPubKey, true);
/* Bu asamada yonetici ve yayıncı etmenler DH anlsma protokolunu
*tamalamıstır.*/
byte[] yaynıcıSharedSecret = yayıncıKeyAgree.generateSecret();
yayıncıKeyAgree.doPhase(yoneticiPubKey, true);
SecretKey yayıncıDesKey = yayıncıKeyAgree.generateSecret("DES");
Cipher yoneticiCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
yoneticiCipher.init(Cipher.ENCRYPT_MODE, yayıncıDesKey);
```

```
byte[] cleartext = "This is just an example".getBytes();
byte[] ciphertext = yoneticiCipher.doFinal(cleartext);
```

Yönetici etmen tarafında Diffie-Hellman protokolü gerçeklemesini aşağıdaki gibi yapmak mümkündür.

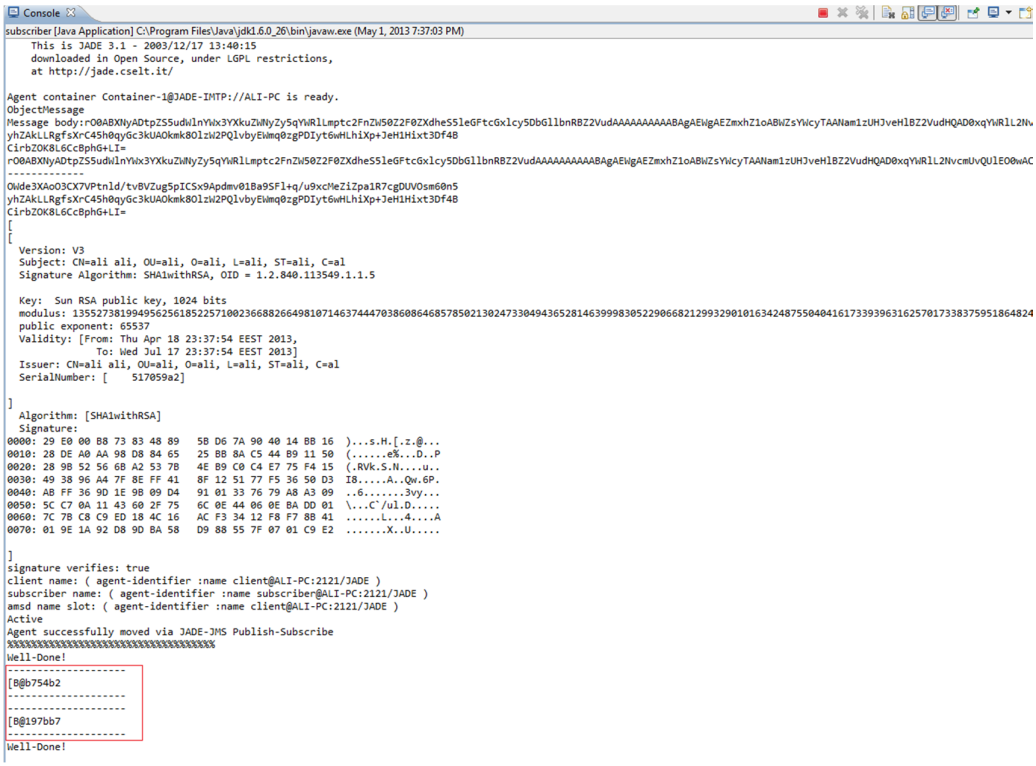
```
//yayıncıpubkeyenc yayıncı etmen taarfından gonderilir
 byte[] yayıncıPubKeyEnc = {};
 KeyFactory yoneticiKeyFac = KeyFactory.getInstance("DH");
 X509EncodedKeySpec x509KeySpec = new
X509EncodedKeySpec(yayıncıPubKeyEnc);
 PublicKey yayıncıPubKey = yoneticiKeyFac.generatePublic(x509KeySpec);
 DHParameterSpec dhParamSpec = ((DHPublicKey) yayıncıPubKey).getParams();
/ /yonetici DH anahtar çiftini oluşturur
KeyPairGenerator yoneticiKpairGen = KeyPairGenerator.getInstance("DH");
yoneticiKpairGen.initialize(dhParamSpec);
KeyPair yoneticiKpair = yoneticiKpairGen.generateKeyPair();
// yonetici kendi DH anahtar anlaşmasını olusturur
 KeyAgreement yoneticiKeyAgree = KeyAgreement.getInstance("DH");
yoneticiKeyAgree.init(yoneticiKpair.getPrivate());
 byte[] yoneticiPubKeyEnc = yoneticiKpair.getPublic().getEncoded();
//Anahtar anlasmasi
 yoneticiKeyAgree.doPhase(yayıncıPubKey, true);
 byte[] yoneticiSharedSecret = new byte[yayıncıLen];
 int yoneticiLen;
yoneticiLen = yoneticiKeyAgree.generateSecret(yoneticiSharedSecret, 0);
byte[] yoneticiSharedSecret = yoneticiKeyAgree.generateSecret();
yoneticiKeyAgree.doPhase(yayıncıPubKey, true);
SecretKey yoneticiDesKey = yoneticiKeyAgree.generateSecret("DES");
Cipher yayıncıCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
yayıncıCipher.init(Cipher.DECRYPT_MODE, yoneticiDesKey);
 //yayıncı (publisher agent) etmen tarafından gonderilen acl mesajı içerisindeki sifreli
metin
 byte[] ciphertext ={};
byte[] recovered = yayıncıCipher.doFinal(ciphertext);
```

Yukardaki gerçeklemeler önerilen güvenlik mimarisini gerçeklemek için örnek teşkil etmekte olan yapıları temsil etmekte olup bu mimari farklı şekillerde veya farklı kütüphaneler kullanılarak ta gerçeklenebilir.

Çalışan sistemde yayıncı etmenin kendini serilize ettikten sonra imzalı özetini kendi sonuna eklediği aşağıdaki şekilde görülmektedir.

Şekil 4.5 : RSA imzalı yayıncı etmen özeti.

Çalışan sistemde yönetici etmen tarafından imza, veri bütünlüğü kontrolü ve kırmızı kutu içinde göç eden yayıncı etmen şifreleme, şifre çözme işlemleri aşağıdaki şekilde görülmektedir.



Şekil 4.6 : RSA imzalı yayıncı etmen özeti imza doğrulama.

5. SONUÇLAR

Bu çalışmada genel olarak hareketli etmen sistemlerinin özel olarak ise yayınla/abone ol modeli ile konu tabanlı olarak göç eden etmen sistemlerinin güvenli hale getirilebilmesi için bir mimari önerilmektedir. Sistemin güvenli hale getirilmesi problemi kimlik doğrulama, bütünlük ve veri gizliliği temel gereksinimlerinin karşılanması durumuna indirgenmiştir. Önerilen mimaride temel gereksinimleri karşılayacak olan güvenlik fonksiyonları etmenin kendisine yüklenerek güvenlik bağlamında etmenin üzerinde çalıştığı platforma olan bağımlılığı en aza indirmektedir. Bu özelliğiyle mimari platformlar arası etmen göçünü desteklemektedir. Mimarinin gerçeklenmesinde halihazırdaki açık standartlar ve teknolojiler kullanılarak farklı platform ve sistemler arasında birlikte çalışabilirlik yüksek güvenilirlikle sağlanmaya çalışılmıştır. Mimari gevşek bağlı ama kendi içinde yüksek uyumlu modüllerin bir araya getrilmesiyle oluşturulmuştur. Bu özellik kullanılan standartların ya da teknolojilerin gelişmesi veya değişmesi durumunda en az gayretle en çok faydanın sağlanmasını tesis etmektedir. Bu da demek oluyor ki bir modülün gerçeklenmesinde kullanılan bir teknoloji yada algoritmanın değişmesi diğer bir modülü hiç etkilemeyecek ya da en az etkileyecektir. Genel olarak güvenlik etmen seviyesi ve taşınım seviyesi olmak üzere iki ayrı katman olarak sunulmaktadır. Bu da etmen seviyesini taşınım seviyesine gevşek bağlamaktadır ve herhangi bir seviyede yapılan bir değişiklik diğerini en az etkilemektedir ve bu sayede her iki katmanda kullanılan teknolojiler ve standartlar bir birinden bağımsız bir şekilde değiştirilebilmektedir. Bu bahsedilen gevşek bağlılıklar ve konuların ayrılığı önerilen mimariye esneklik ve genişletilebilirlik katmaktadır. Bu esneklik ve genişletilebilirlik ortaya konulan güvenlik mekanizmasının yeni gelişen tehditler karşısında halihazırda kullanılmakta olan teknoloji veya algortimaların yenileriyle kolaylıkla değiştirilmesiyle rahatlıkla güçlendirilebilir olmasını sağlamaktadır.

KAYNAKLAR

Albert, M., Laengle, T., Woern, H., Capobianco, M. and BrighentiA. (2003). "Multi-agent Systems for Industrial Diagnostics", Proceedings of 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, 483–488, Washington, DC.

Brown, P. and RossakW. (2005). "Mobile Agents", Morgan Kaufmann Publishers and dpunkt.verlag.

Decker, K. and Sycara, K.(1997). "Intelligent Adaptive Information Agents", Journal of Intelligent Information Systems9(3): 239–260.

Dierks T. ve AllenC. (1999). "https://ietf.org/rfc/rfc2246.txt", IETF Request for Comment.

Dierks T. ve Rescorla E. (2008). "http://tools.ietf.org/html/rfc5246", IETF Request for Comment.

Fricke, S. Bsufka, K., Keiser, J., Schmidt, T., Sesseler, R. and Albayrak, S.(2001). Communications of the ACM, 44(4): 43–48.

Greenwood, D., Vitaglione, G., Keller, L. and Calisti, M.(2006). "Service Level Agreement Management with Adaptive Coordination", Proceedings of theInternational Conference on Networking and Services (ICNS'06), SiliconValley,USA.

Hayzelden, A.L. and Bourne, R.A.(2001). "Agent Technology for Communication", John Wiley & Sons,Ltd, London, UK.

Jennings, N. (1994). "The Archon System and its Applications", Proceedings of the2nd International Working Conferenceon Cooperating Knowledge Based Systems (CKBS-94), 13–29, Dake Centre, University of Keele, UK.

Jennings, N., Faratin, P., Johnson, M.J., Norman, T.J., O'Brien, P. and Wiegand, **M.E.** (1996). "Agent-based Business Process Management", International: Journal of Cooperative Information Systems 5(2-3): 105–130.

Jennings, N. and Wooldridge, M. (1998). "Applications of Intelligent Agents", Agent Technology: Foundations, Applications, and Markets, 3–28, Secaucus, NJ, Springer-Verlag, Berlin.

Karzan M. A., Erdoğan N. (2013). "Topic Based Agent Migration Scheme via Publish/Subscribe Paradigm", International Conference on Computer Engineering and Technology.

Klusch, M.(2001). "Information Agent Technology for the Internet: a Survey", DataKnowledge Engineering 36(3): 337–372.

Mir, J. (2004). “Protocolos criptogr´aficos para canales de comunicaci´on an´onimos”, PhD thesis, Escola de postgrau.

Moreno, A. and Nealon, J. (2003). “Applications of Software Agent Technology” Whitestein.

Neagu, N., Dorer, K., Greenwood, D. and Calisti, M. (2006). “LS/ATN: Reporting on a Successful Agent-Based Solution for Transport Logistics Optimization”Proceedings of the IEEE 2006 Workshop on Distributed Intelligent Systems(WDIS’06), Prague, 2006.

Novak P., Rollo M., Hodik J., Vlcek T. (2003). “Communication Security in Multi-agent Systems”, CEEMAS 2003, LNAI 2691, sf. 454-463, Springer-Verlag Berlin Heidelberg.

Parunak, H. (1987). “Manufacturing Experience with the Contract Net”, Huhns, M.(ed.), Distributed Artificial Intelligence, 285–310, Pitman, London.

Picco, G.P. (2000). “Understanding Code Mobility”, ICSE ’00: Proceedings of the 22nd International Conference on Software Engineering, 834, ACM Press, New York.

Prem M. V. (2012). “Securing Mobile Agent and Its Platform from Passive attack of Malicious Mobile agents”, IEEE, International Conference on Advances in Engineering, Science and Management (ICAESM).

Russell, S.J. and Norvig, P. (2003). “Artificial Intelligence: a Modern Approach”, 2nd edn. Prentice Hall,.

Stallings,W. (2011). “Network Security Essentials: Applications and Standards”, 4th Edition, Pearson Education.

White, J.E. (1996). “Telescript Technology: Mobile Agents”, Bradshaw Jeffrey, (ed), *Software Agents*, AAAI Press/MIT Press.

Yang,C. (2006). “Secure Internet Applicaitons Based on Mobile Agents”, International Journal of Network Security, Vol.2, No.3, 228–237.

Zhang,Q. (2009). “Secure Mobile Agents with Designated Hosts”, IEEE, International Conferene on Network and system Security (NSS).

ÖZGEÇMİŞ

Ad Soyad: Ahmet Ali Karzan

Doğum Yeri ve Tarihi: İstanbul, 02.10.1984

Adres: Yasemen sokak no: 14/2 Yeniköy – Sarıyer/İstanbul

E-Posta: alikarzan@yahoo.com

Lisans: İstanbul Üniversitesi

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- Karzan A. A., Erdoğan N. 2013 "Securing Mobile Agent Systems in which The Agents Migrate via Publish/Subscribe Paradigm", International Conference on Computing Communications and Networking Technologies, July 4-6 2013, Tamil Nadu, India.