



T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ  
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

160044

LDAP ERİŞİMİNDE KULLANILAN  
GÜVENLİK PROTOKOLLERİ VE ÖZGÜN  
BİR GÜVENLİK PROTOKOLÜ (ELIAC)  
TASARIMI

F. Koray ATSAN

YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ  
ANABİLİM DALI

160044

TEZ DANIŞMANI

Yrd.Doç.Dr. F. Erdoğan SEVİLGİN

GEBZE

2003



Koray ATSAN'ın tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 16-06-2003 tarih ve .2003-18 sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

### JÜRİ

ÜYE : Yrd. Doç. Dr. F. Erdoğan SEVİNGEN  
(Tez Danışmanı)

ÜYE : Yrd. Doç. Dr. İbrahim SOĞUKPINAR

ÜYE : Yrd. Doç. Dr. İlyas KANDEMİR

### ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..30..10..2003.. tarih ve ..2003../35.. sayılı kararı.

İmza/Mühür  
Prof. Dr. Adem KILIC  
GYTE Mühendislik ve Fen Bil.  
Enstitüsü Müdürü

## ÖZET



Dizin sistemleri özellikle LDAP sistemleri günümüzde gerek internette gerekse çok kullanıcıli yerel ağlarda, bilgi paylaşımına yönelik, çok sık kullanılan sistemlerdir. Dizin sistemlerinde tasnif dışı bilgiler saklandığı kadar gizlilik dereceli bilgiler de saklanabilmektedir. Bu nedenle dizin sistemlerine ağ üzerinden yapılan erişimlerde güçlü kimlik doğrulama ve gizlilik servislerini sağlamak amacıyla kriptografik yöntemlerin kullanılması kaçınılmaz olmuştur. Bu amaçla kullanılan bazı standartlaşmış protokoller arasında SASL, TLS, Kerberos ve IPSEC sayılabilir.

Bu çalışmada, uygulama katmanında çalışan ve LDAP protokolüne, güçlü kimlik doğrulama, bütünlük ve gizlilik gibi güvenlik servisleri sağlamaya yönelik bir protokol (**ELIAC – Easy LDAP Interface for Authentication and Confidentiality**) tasarlanmıştır. ELIAC'ın yapılacak saldırılara karşı sağlam ve güvenli olduğu, saldırı senaryoları tanımlanarak ve protokolün bu senaryolara karşı davranışı açıklanarak ispata çalışılmıştır. ELIAC protokolünün fark ve özelliklerin net bir şekilde anlaşılabilmesi açısından, diğer güvenlik protokolleri ile birebir karşılaştırılması yapılmıştır.

ELIAC protokolü gerçekleştirme, uygulama ve kullanım kolaylığı sağlamak amacıyla mümkün olduğunca basit ve sadedir.



## SUMMARY

Directory Systems especially LDAP Directories, are widely used on Internet and heavy user networks for sharing data. As well as unclassified information, classified information may also be stored in directories. For this reason, while gaining access to directories over networks, it is unavoidable to deploy cryptographic mechanisms in order to support strong authentication and confidentiality services. Some standard protocols providing these services are SASL, TLS, Kerberos and IPSEC.

In this study we aim at designing an application layer protocol (**ELIAC – Easy LDAP Interface for Authentication and Confidentiality**) which provides strong authentication, confidentiality and integrity services. To emphasize reliability and robustness of ELIAC, various attack scenarios are defined and how ELIAC resists is shown. For the sake of clarifying the differences between ELIAC and above listed protocols and their properties, a cross comparison is provided.

ELIAC is intended to be kept as simple as possible taken into consideration the ease of implementation, application and use factors.

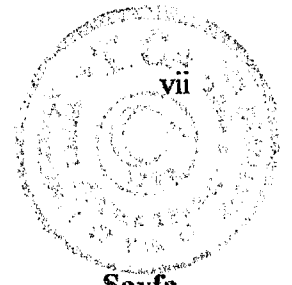


## TEŞEKKÜR

Bu tez çalışmasının tamamlanması yönünde teknik olarak titizlikle incelemeleri ve yaptığımız tartışmalar neticesinde çalışmamın tez halini alması yönündeki katkıları nedeniyle sayın hocam Yard.Doç.Dr. Fatih Erdoğan SEVİLGEN'e

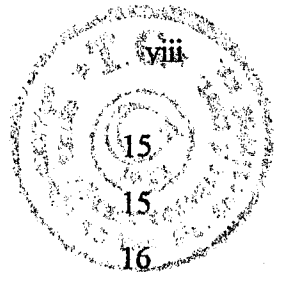
Tez çalışmamın gerek teknik gerekse yazım konusunda eksikliklerini belirterek en az hata ile yayınlanması yönündeki katkıları nedeniyle sayın hocam Yard.Doç.Dr. İbrahim SOĞUKPINAR ve Yard.Doç.Dr. İlyas KANDEMİR'e

Bana Yüksek Lisans yapma imkanını tanıyan TÜBİTAK – UEKAE (TÜBİTAK Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü)'ne ve Manevi desteklerini daima hissettiğim aileme teşekkürlerimi sunarım.

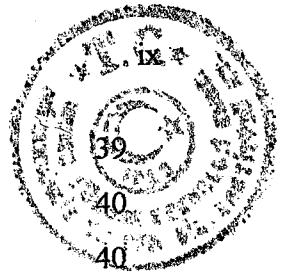


# İÇİNDEKİLER DİZİNİ

|                                                           | <u>Sayfa</u> |
|-----------------------------------------------------------|--------------|
| ÖZET                                                      | iv           |
| SUMMARY                                                   | v            |
| TEŞEKKÜR                                                  | vi           |
| İÇİNDEKİLER DİZİNİ                                        | vii          |
| SİMGELER VE KISALTMALAR DİZİNİ                            | x            |
| ŞEKİLLER DİZİNİ                                           | xi           |
| TABLolar DİZİNİ                                           | xii          |
| <br>                                                      |              |
| 1. GİRİŞ                                                  | 1            |
| 2. DİZİN SİSTEMLERİ VE ERİŞİM PROTOKOLLERİ                | 4            |
| 2.1. Dizin Nedir ?                                        | 4            |
| 2.2. Çevrimiçi Dizinler Dinamiktir                        | 4            |
| 2.3. Çevrimiçi Dizinler Esnektir                          | 5            |
| 2.4. Çevrimiçi Dizinler Güvenli Hale Getirilebilirler     | 5            |
| 2.5. Çevrimiçi Dizinler Kişiselleştirilebilirler          | 6            |
| 2.6. Dizinlerle Veritabanları Arasındaki Farklar          | 6            |
| 2.6.1. Okuma/Yazma Oranı                                  | 6            |
| 2.6.2. Bilginin Çeşitlendirilebilirliği                   | 6            |
| 2.6.3. Dağıtık Yapı                                       | 7            |
| 2.6.4. Çoklama (Replikasyon)                              | 7            |
| 2.6.5. Standartlar ve Ortak Çalışabilirlik                | 7            |
| 2.7. X.500 Dizin Hizmetleri Standardı (DAP)               | 8            |
| 2.8. LDAP (Lightweight Directory Access Protocol)         | 8            |
| 2.9. LDIF (LDAP Data Interchange Format)                  | 9            |
| 2.10. LDAP İsimlendirme Modeli                            | 10           |
| 2.11. LDAP İşlevsel Modeli                                | 12           |
| 2.11.1. Sorgulama İşlemleri                               | 12           |
| 2.11.2. Güncelleme İşlemleri                              | 12           |
| 2.11.3. Doğrulama ve Denetim İşlemleri                    | 12           |
| 2.12. LDAP da Erişim Denetimi (Erişim Denetim Listeleri ) | 13           |

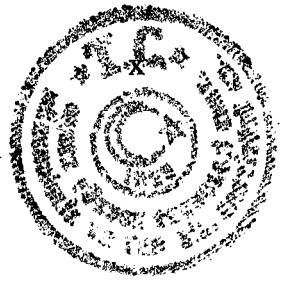


|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| 2.13. LDAP API ve Programlama                                                  | 15 |
| 2.14. OpenLDAP                                                                 | 15 |
| 2.15. Dizinlerin Güvenlik Hizmetlerindeki Yeri                                 | 16 |
| 3. LDAP DOĞRULAMA VE GİZLİLİK TEKNİKLERİ                                       | 18 |
| 3.1. Doğrulama, Doğrulama Bilgileri , Kimlik                                   | 18 |
| 3.1.1. Anonim Doğrulama                                                        | 18 |
| 3.1.2. Parola Tabanlı Doğrulama                                                | 19 |
| 3.1.3. Sertifika Tabanlı Doğrulama                                             | 19 |
| 3.1.3.1. TLS 1.0 Protokolü (Transport Layer Security)                          | 20 |
| 3.1.3.2. SASL (Simple Authentication and Security Layer)                       | 22 |
| 3.1.3.3. Kerberos v4, v5                                                       | 23 |
| 4. INTERNET KATMANLARI ve ELIAC PROTOKOLÜ'NÜN YERLEŞİMİ                        | 26 |
| 4.1. TCP/IP Güvenliği ve Internet Katmanları                                   | 26 |
| 4.2. ELIAC'ın Katman Modelinde Yerleşimi                                       | 26 |
| 5. ELIAC VE ALT PROTOKOLLERİ                                                   | 28 |
| 5.1. İklendirme ve El Sıkışma Protokolü                                        | 28 |
| 5.2. İletişim Protokolü                                                        | 30 |
| 5.3. Ortak Bilinen Güncelleme Protokolü                                        | 31 |
| 5.4. Kullanılan Algoritmalar ve Anahtar Ömürleri                               | 32 |
| 5.5. Protokol Değişkenleri                                                     | 33 |
| 6. ELIAC – TLS – KERBEROS – IPSEC PROTOKOLLERİNİN KARŞILAŞTIRILMASI            | 34 |
| 6.1. Güvenlik Servisleri Karşılaştırma Tablosu                                 | 34 |
| 6.2. Protokol Özellikleri karşılaştırma Tablosu                                | 35 |
| 7. SALDIRILAR VE GÜVENLİK ANALİZİ                                              | 37 |
| 7.1. LDAP'a yönelik Olası Saldırıları                                          | 37 |
| 7.1.1. Hizmet Aksatma                                                          | 38 |
| 7.1.2. LDAP Sunucuların Virüs Dağıtım Sistemi Olarak Kullanılması              | 39 |
| 7.1.3 Yoğun Saldırıları (Brute Force Attacks)                                  | 39 |
| 7.2. ELIAC'a Yönelik Olası Saldırıları ve ELIAC'ın Saldırıları Karşı Davranışı | 39 |



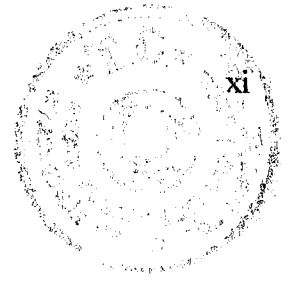
|                                                |    |
|------------------------------------------------|----|
| 7.2.1. Aradaki Adam Saldırısı                  |    |
| 7.2.1.1. Araya Girme                           | 40 |
| 7.2.1.2. Engelleme                             | 40 |
| 7.2.1.3. IP Taklit Etme                        | 41 |
| 7.2.2. Tekrarlama Saldırısı                    | 42 |
| 7.3. Güvenlik Analizi                          | 42 |
| 7.3.1. ELIAC'ın Zafiyetleri                    | 43 |
| 7.3.1.1. Ortak Bilinen Güvenliği               | 43 |
| 7.3.1.2. Ortak Bilinen Ele Geçirilme Tehlikesi | 43 |
| 7.3.1.3. ELIAC Hata Kodlaması                  | 45 |
| 8. SONUÇLAR                                    | 46 |
| KAYNAKLAR                                      | 48 |
| ÖZGEÇMİŞ                                       | 49 |





## SİMGELER VE KISALTMALAR

| <u>Simgeler</u> | <u>Açıklama</u>                                                                       |
|-----------------|---------------------------------------------------------------------------------------|
| DAP             | Directory Access Protocol (Dizin Erişim Protokolü)                                    |
| LDAP            | Lightweight Directory Access Protocol (Basit Dizin Erişim Protokolü)                  |
| SASL            | Simple Authentication and Security Layer (Basit Kimlik Doğrulama ve Güvenlik Katmanı) |
| TLS             | Transport Layer Security (Ulaşım Katmanı Güvenliği)                                   |
| IPSEC           | IP Security (IP Güvenliği)                                                            |
| Authentication  | Kimlik Doğrulama                                                                      |
| Confidentiality | Gizlilik                                                                              |
| Non repudiation | İnkâr Edememe                                                                         |
| Integrity       | Bütünlük                                                                              |
| TCP             | Transmission Control Protocol (İletim Denetleme Protokolü)                            |
| DN              | Distinguished Name (TKA)                                                              |
| TKA             | Tekil Kayıt Adı                                                                       |
| ACL             | Access Control List (Erişim Denetim Listesi)                                          |
| DS              | Doğrulama Sunucusu                                                                    |
| BTKA            | Bağlı TKA                                                                             |
| Nonce           | Tek kullanımlık sayı                                                                  |
| RFC             | Request for Comments (Yorum İstekleri)                                                |
| IANA            | Internet Assigned Numbers Authority (İnternet Atanmış Numaralar Makamı)               |
| OSI             | Open Systems Interconnection (Açık Sistem Arabağlantısı)                              |
| PKI             | Public Key Infrastructure (AAA)                                                       |
| AAA             | Açık Anahtar Altyapısı                                                                |
| CA              | Certification Authority (SM)                                                          |
| SM              | Sertifika Makamı                                                                      |



## ŞEKİLLER DİZİNİ

| <u>Şekil</u>                                  | <u>Sayfa</u> |
|-----------------------------------------------|--------------|
| 2.1. LDAP isimlendirme modeli                 | 11           |
| 3.1. TLS El Sıkışma Protokolü                 | 21           |
| 3.2. Kerberos Protokolü                       | 25           |
| 4.1. Internet Katmanları                      | 26           |
| 4.2. ELIAC Katman Yerleşimi                   | 27           |
| 5.1. ELIAC Genel Yapısı                       | 28           |
| 5.2. ELIAC İlkendirme ve El Sıkışma Protokolü | 30           |
| 5.3. ELIAC İletişim Protokolü                 | 31           |
| 5.4. Ortak Bilinen Güncelleme Protokolü       | 32           |
| 7.1. Araya girme saldırısı                    | 40           |
| 7.2. Engelleme saldırısı                      | 41           |
| 7.3. IP Taklit etme saldırısı                 | 42           |



## TABLÖLAR DİZİNİ

| <b><u>Tablo</u></b>                   | <b><u>Sayfa</u></b> |
|---------------------------------------|---------------------|
| 5.1. Algoritmalar ve Anahtar Ömürleri | 32                  |
| 5.2. Protokol Değişkenleri            | 33                  |
| 6.1. Güvenlik Servisleri              | 35                  |
| 6.2. Protokol Özellikleri             | 36                  |
| 7.1. Ortak Bilinen Kriterleri         | 43                  |





# 1. GİRİŞ

İnternet ve çok kullanıcıli ağların yaygınlaşması ile, bu ağların üzerinde çalışacak sistemlerin çok fonksiyonellik, ölçekleme ve performans gibi özellikleri önem kazanmıştır. Farklı amaçlarla kullanıma açık olması, çok sayıda kullanıcı ve cihazın ölçeklenebilmesi ve aynı zamanda sistem performansının belli bir kalite seviyesinin üzerinde tutulmasını sağlaması dizin sistemlerinin yaygın kullanımını sağlamıştır.

Dizin sistemlerini diğer pek çok veri tabanı sisteminden ayıran en önemli özellik çok okunan az yazılan (güncellenen) uygulamalar için tasarlanmış olmasıdır. Erişim protokolü ve ağaç veri modeli kullanılan iç yapısı çok okumaya göre optimize edilmiştir. Tipik bir dizin sistemi yazıldığından 1000, hatta 10000 defa daha fazla okunur. Dizin sistemlerinin genel kullanım alanları aşağıda sıralanmıştır :

- Adres bilgileri saklanması (e-posta, telefon, faks numaraları, posta adresleri)
- Ağ bilgileri (login kullanıcı adı, durum, öncelikler)
- İnsan Kaynakları Bilgileri (isimler, bölümler, durum, roller)
- Kaynak Bilgileri (kaynaklar, yetenekler)
- Güvenlik bilgileri (parolalar, tek giriş (Single Sign On), X.509 sertifikalar, erişim yetkileri )

LDAP (Lightweight Directory Access Protocol), verimli ve gerçekleştirilmesi kolay olmasının yanında, kolay kullanılabilir bir dizin erişim protokolüdür [1]. LDAP protokolünün ilk sürümü 1993, ikinci sürümü 1995, üçüncü ve halihazırdaki sürümü ise 1997 yıllarında yayınlanmıştır.

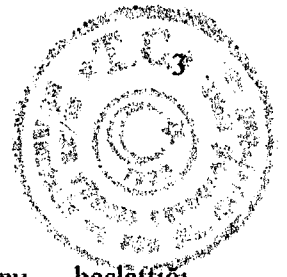
Dizin sistemlerinde saklanan örnekleri yukarıda sıralanan bilgiler, şifrelenmeden, açık bir halde saklanmakta, bu nedenle bu bilgilere erişimin güvenli

2

bir şekilde yapılması gerekmektedir [2]. Özellikle de güvenlik bilgileri gibi hassas verilerin dizin sistemlerinde saklanması durumunda saldırganların hedefi haline gelme ihtimallerinden dolayı güvenli erişim gereksinimi kaçınılmazdır. Dizin sunucularına yönelebilecek tipik saldırılar Bölüm 7.1 de sıralanmıştır. Bu nedenle dizin sistemlerine ağ üzerinden yapılan erişimlerde güçlü kimlik doğrulama ve gizlilik servislerini sağlamak amacıyla kriptografik yöntemlerin kullanılması kaçınılmaz olmuştur.

Dizin sistemlerine yapılabilecek saldırıların engellenmesi amacıyla SASL, IPSEC, SSL/TLS ve Kerberos gibi iletişimde kimlik doğrulama ve/veya gizlilik sağlayan protokoller dizin sistemlerine erişiminde güvenlik sağlanmasına destek vermektedirler [3]. SASL (Simple Authentication and Security Layer), bağlantı temelli protokollere kimlik doğrulama desteği sağlayan bir kimlik doğrulama çerçevesidir. IPSEC, IP v6 ile kullanılmak üzere tasarlanmış karmaşık bir güvenlik protokolüdür. IPSEC, Diffie – Hellman, IKE vb. temel protokolleri geliştirerek IP iletişimine makine bazlı kimlik doğrulama, bütünlük, gizlilik servislerini sağlar [4]. SSL ilk olarak Netscape şirketi tarafından 1995 de geliştirilmiş, Oturum katmanında çalışan karmaşık bir güvenlik protokolüdür. Son sürümü SSL v3 haline gelmiş ve bunu müteakiben IETF tarafından TLS v1 olarak standartlaştırılmıştır. TLS, güvenlik sağlamak amacıyla açık anahtar sertifikaları kullanarak, kimlik doğrulama, bütünlük, gizlilik ve inkar edememe servislerini sağlar [5]. Kerberos ise ilk olarak MIT de Athena projesi kapsamında geliştirilmiş uygulama katmanında çalışan bir güvenlik protokolüdür. Kimlik doğrulama, bütünlük, gizlilik servislerini sağlar [4]. Kerberos v4 den sonra, IETF tarafından Kerberos v5 standartlaştırılmıştır.

Bu çalışmada, yukarıda sıralanan protokoller ile aynı doğrultuda, LDAP dizin sunucusuna erişimde kimlik doğrulama, bütünlük ve gizlilik servislerini sağlayan bir güvenlik protokolü (ELIAC – Easy LDAP Interface for Authentication and Confidentiality) önerilmektedir. ELIAC'ın bu protokollerden en önemli farklı yüksek seviyede güvenlik sağlamanın rağmen anahtar yönetimine ihtiyaç duymaması ve LDAP için özel tasarlanması nedeniyle sade ve gerçekleştirilmesi kolay bir protokol olmasıdır. ELIAC üç adet alt protokolden oluşur:



## 1. İklendirme ve El Sıkışma Protokolü

ELIAC istemcisinin, ELIAC sunucusu ile iletişim oturumunu başlattığı protokoldür. Bu aşamada, istemci ve sunucu karşılıklı olarak açık anahtarlarını değiştirirler ve sunucu tarafından üretilen oturum anahtarı iletişim protokolünde kullanılmak üzere istemciye gönderilir.

## 2. İletişim Protokolü

ELIAC istemcisi ile sunucusu arasında düzenli veri iletişiminin gerçekleştiği protokoldür. Bu aşamadaki iletişim kimlik doğrulamalı, gizli ve bütünlük kontrollü bir şekilde gerçekleşir.

## 3. Ortak Bilinen Güncelleme Protokolü

ELIAC istemcisi ile sunucusu arasında açık anahtar değiş tokuşunda kullanılan ortak bilinenin güvenlik amacıyla güncellendiği protokoldür.

Bölüm 2’de, dizin sistemleri ile ilgili bilgiler, Bölüm 3’de, güvenlik protokolleri’nin detaylı açıklamaları, Bölüm 6’da ELIAC alt protokollerinin detaylı tasarım ve açıklamaları, kullanılan algoritmalar, protokol değişkenleri’nin tanımları, Bölüm 7’de ELIAC’ın diğer güvenlik protokolleri ile karşılaştırılması ve Bölüm 8’de ELIAC’ a yönelik saldırılar ve güvenlik analizi verilmiştir.



## 2. DİZİN SİSTEMLERİ VE ERİŞİM PROTOKOLLERİ

### 2.1. Dizin Nedir ?

En geniş anlamda dizin özelleşmiş bir veritabanıdır. Çeşitli bilgileri organize bir halde tutarak bunlara özellikle okuma yapmak amacıyla rahat ve hızlı ulaşımı sağlayan bir yapıdır. Dizinler aslında günlük hayatımızın bir parçasıdır. Her gün karşılaştığımız dizinlere örnek olarak telefon rehberleri, sarı sayfalar, televizyon rehberlerini ve alışveriş kataloglarını verebiliriz.

Bilgisayar ve ağ ortamlarındaki dizinler de, kağıda basılı dizinlere birçok yönden benzerler fakat en önemli fark bu tip dizinlerin “çevrimiçi” dizinler olmasıdır. Çevrimiçi dizinler diğerlerinden aşağıda sıralanan özellikleriyle ayrılabilirler.

### 2.2. Çevrimiçi Dizinler Dinamiktir

Günlük hayatta karşılaştığımız matbu dizinler statiktir, yani çok sık değişmezler. Mesela bir telefon rehberi yılda bir defa güncellenir. Yeni taşındığınız bir şehrin telefon rehberinde yer alabilmek için aylarca beklemeniz gerekebilir. Mağazaların dağıttığı alışveriş katalogları yılda birkaç kez düzenlenir ama onlar bile şu anda mağazanın elinde hangi boyalar, renkler var gibi önemli sorulara cevap veremezler çünkü bu bilgiler o kadar sık değişirler ki bazen katalog elinize ulaştığı anda bile içindeki bazı bilgiler güncelliğini yitirmiş olabilir.

Diğer yandan Çevrimiçi dizinler dinamiktir: Yani istenilen sıklıkta güncellenebilirler. Burada istenilen sıklık aslında dizin yöneticilerinin (admin) istediği sıklıktır. Bazen güncelleme işini idari prosedürlere yükleyip güncelleme işi otomatige bağlanabilir. Bunların dışında çevrimiçi dizinler, matbu dizinlerde hayal bile edilemeyecek bazı faydalı uygulamalara kaynak olabilirler. Mesela : Bir şirketin binalarına giriş otomatik kart okuyucuyla sağlanır. Eğer bu okuyucuların giriş bilgilerini dizinden alması sağlanabilirse, şirketin bazı politikaları çok rahat

uygulanabilir. İçeri girmesinde mahsur görülen bir kişinin dizindeki kaydı silinerek ve bu kayda “Giremez” ibaresi koyularak, tek bir işlemle şirketin politikası uygulanabilir.

Diğer bir güzel örneğe yine bir çalışma ortamından verilebilir. Bir şirkette çalışan insanların konum bilgisi (hangi odada..), kişilerin odalara kart okuyuculardan geçtikten sonra girmesine dayanarak, dizinde saklanabilir. Bu durumda kişiye gelen telefonlar, faksler veya mesajlar, dizindeki konum bilgisine başvurularak kişinin o anda bulunduğu odaya yönlendirilebilir. Böyle bir uygulamayı matbu dizinlerle düşünmek mümkün değildir.

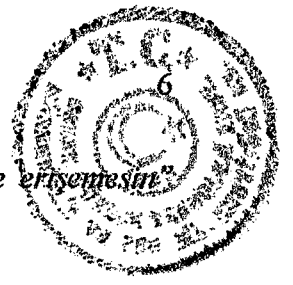
### 2.3. Çevrimiçi Dizinler Esnektir

Çevrimiçi dizinler içerik yönünden çok daha esnektirler. Matbu dizinlerde saklanabilecek bilgi sınırlıdır. Bir kişinin adresi ve telefon numarası dışında ihtiyacınız olan birçok bilgi matbu dizinlerde bulunmaz. Fakat kişinin faks numarası, e-posta adresi hatta küçük bir özgeçmişi gibi bilgilere de ihtiyaç duyulabilir. İşte bu noktada çevrimiçi dizinler çok kullanışlıdır. Çevrimiçi dizinlere yeni bilgi(ler) eklemek çok ucuz ve kolaydır fakat matbu dizinlere yeni bir bilgi eklemek için tüm dizinin yeniden tasarlanıp, basılıp, dağıtılması gereklidir. Çevrimiçi dizinlerde değişiklik yapmak çok kolaydır ve dağıtıma da gerek yoktur çünkü kişiler kendileri dizine erişip sorgulama yaparlar, yani kişilerin kendi özel kopyaları yoktur.

Çevrimiçi dizinler veri saklama ve saklanmış veriler üzerinde arama yapma yönünden de çok esnektirler. Matbu dizinler genelde tek bir öznitelik (ad, soyad) kullanılarak indekslenirler ve arama bu indeksle yapılabilir. Ama çevrimiçi dizinler bu yönden de esnektirler, her özniteliğe göre arama yapılabilir.

### 2.4. Çevrimiçi Dizinler Güvenli Hale Getirilebilirler

Matbu dizinlerin güvenlik konusunda fazla bir yetenekleri yoktur. Pazarlamacılar kurtulmak için dizinde yer almazsınız ama gerçekten size ulaşmak isteyen arkadaşlarınız vs size ulaşamazlar. Çevrimiçi dizinlerde Erişim Denetim Listeleri (ACL) vasıtasıyla size ait kayıda kimlerin erişeceğini



belirleyebilirsiniz. Mesela “Benim sitemde yaşayanların dışında kimse erişemez” gibi faydalı bir erişim kontrolü sağlayabilirsiniz.

## 2.5. Çevrimiçi Dizinler Kişiselleştirilebilirler

Matbu dizinler yerel olarak basılabilir ama bu yerellik kişisel baza kadar indirilemez. Oysa çevrimiçi dizinler her kişiye özel hizmet verebilirler. Mesela bir mağaza katalogu her müşterisine kendine özel bilgiler sunabilir: Kadınlara kadın elbiseleri, o müşterinin sevdiği renkler, modeller vs.

## 2.6. Dizinlerle Veritabanları Arasındaki Farklar

### 2.6.1. Okuma/Yazma Oranı

Dizinler çok okunan az yazılan (güncellenen) uygulamalar için tasarlanmıştır. Erişim protokolü ve iç yapısı çok okumaya göre optimize edilmiştir. Tipik dizin yazıldığından 1000, hatta 10000 defa daha fazla okunur. Veritabanları işlem (Transaction) bilgilerini saklamak, veya izleme (auditing) bilgilerini saklamak amacıyla kullanılabilir. Bu tip bilgiler ya yılda birkaç kez okunurlar(auditing) yada çok sık güncellenirler (Transaction). Sık okunan, seyrek güncellenen veri tiplerine örnek olarak katalog bilgileri, ağ yönlendirme tabloları, e-posta, telefon numarası gibi bilgiler verilebilir. Sık okunma seyrek güncellenme özelliği dizin tasarımında bir temel oluşturur. Bir okunma işlemi yazmaya göre 1000 kez daha fazla yapıyorsa, dizin tasarımında okuma işlemine daha çok kaynak ve emek harcanıp bu işlem daha çok hızlandırılabilir.

### 2.6.2. Bilginin Çeşitlendirilebilirliği

Dizinlerde ne tip bilgi saklanabilir, bu bilgileri saklarken hangi kurallara uyulmalı gibi soruların cevabı o dizinin “Şema” sında saklıdır. Bir dizin genel olarak önceden tanımlanmış standart bazı öznitelik tipleriyle beraber gelir. Ama ihtiyaca paralel olarak yeni öznitelik tipleri tanımlanabilir. Veritabanları çok değişik veri tipleri saklamaya uygun olmakla beraber, yeni veri tipi tanımlamaya açık değildir.



### 2.6.3. Dağıtık Yapı

Dizinlerin veritabanlarından ayrıldığı en önemli noktalardan biri de dağıtık yapılanmaya verdiği destektir. Veritabanlarıyla da dağıtık bir yapı kurulabilir ama en fazla bir tablo A makinesinde, başka bir tablo B makinesinde tipi bir dağıtım yapılabilir. İki tabloyu da ilgilendiren bir arama yapılırsa performans problemleriyle karşılaşılır. Dizinlerin tasarım amaçlarından biride dağıtık veri depolanmasına imkan sağlamaktır. Uluslararası bir şirketin her bir birimi, dizinin kendi ülkesine ait kısmında yetkili olabilir ve kendine ait bilgiyi kendi bünyesinde saklayabilir ve buna rağmen dışarıdan arama yapan birisi için şirketin farklı ülkelerdeki tüm birimleri tek bir ağaç yapısı altında toplanıp, konum olarak olmasa da mantık olarak birleştirilebilir.

### 2.6.4. Çoklama (Replikasyon)

Çoklama, bir veri grubunun birden fazla kopyasını saklama olayına verilen isimdir. Çoklama sayesinde, bir dizin sunucusu bozulursa başka bir kopyası hizmet vermeye devam eder. Ayrıca her bir istemci dizinin kendisine coğrafi olarak yakın olanına bağlanıp daha hızlı hizmet alabilir ve toplamda tüm sistemin performansı artar çünkü aynı anda daha fazla arama isteğine cevap verilebilir. Veritabanlarıyla dizinler burada da farklılık gösterirler. Veritabanlarında genel olarak tutarlılık göstermesi gereken bilgiler tutulur, yani veritabanı çoklandığında bir sunucuda yapılan bir değişiklik anında diğer çoklama sunucularına da yansıtılmalıdır ama dizin bu konuda daha az hassastır. Dizinin çoklama sunucuları arasında oluşabilecek geçici tutarsızlıklar kabul edilebilir. Ayrıca, senkronize çoklamanın veritabanları için performans sorunlarına sebep olduğu belirlenmiştir [6].

### 2.6.5. Standartlar ve Ortak Çalışabilirlik

Veritabanı dünyasında erişim için SQL (Structured Query Language) kullanılır, ama bu dil değişik veritabanlarının ortak kullanımı için tam bir birliktelik sağlamaz. Bir firmanın veritabanı ürününü kullanmış olmak, başka bir firmanın ürününü öğrenmeyi kolaylaştırır ama yine de tam bir ortak çalışabilirlik yoktur. Dizinler bu noktada farklıdır. LDAP (Lightweight Directory Access Protocol) günümüz

dizinlerinde kullanılan standart model ve protokolleri tanımlar. Bu standartlar sayesinde Netscape Dizin Sunucusuna, Microsoft tarafından geliştirilen bir uygulama ile erişim mümkündür. Ama veritabanları için bunu söylemek zordur. Bir Oracle uygulaması, Informix veya Sybase veritabanı ile kullanılamaz. Standartlar ayrıca :

- Dizin istemcilerini, sunuculara bağlı olmaktan kurtarır
- Uygulama geliştirme sürecini, Dizin Sunucu firmalarından bağımsız kılar.

LDAP olmasaydı her bir uygulama kendi dizin sunucusuyla birlikte gelecekti. 24 farklı uygulama kullanan bir sistem yöneticisi bazı bilgileri 24 ayrı dizine girmek zorunda kalacaktı ama LDAP'la tek bir dizin sunucusuna 24 ayrı uygulama, standartlar vasıtasıyla erişebilir ve tek bir noktadan veri girişi yeterli olur.

## 2.7. X.500 Dizin Hizmetleri Standardı (DAP)

Dizin adacıklarının birbirleriyle ortak çalışabilmeleri için tanımlanmış, uluslararası (ISO 9594), standart bir erişim protokolü (DAP) ve hiyerarşik bir isimlendirme yöntemidir. X.500 standardı, dizin bünyesinde fazla miktarda veri saklayabilmek için gerekli olan veri organizasyonunu bir standarda bağlar. Veriyi hiyerarşik, ters ağaç yapısı şeklinde organize eder. Ayrıca güçlü arama mekanizmaları tanımlar ve bilgi erişimini kolaylaştırır.

X.500 standardı, istemci ile sunucu arasındaki iletişimin DAP (Directory Access Protocol) üzerinden gerçekleşmesini gerektirir. DAP bir uygulama katmanı protokolü olarak bütün OSI katmanlarının altta çalışır vaziyette bulunmasını şart koşar. Burada sorun olan nokta OSI katmanlarının çok kaynak gerektirmesi ve basit kullanıcı makinelerine göre ağır bir iletişim protokolü olmasıdır. X.500 standardı bu çalışmanın kapsamı dışındadır.

## 2.8. LDAP (Lightweight Directory Access Protocol)

LDAP (Lightweight Directory Access Protocol), verimli, doğrudan ve gerçekleştirilmesi kolay olmasının yanında, yüksek derecede kullanışlı bir dizin erişim protokolüdür. Esasen LDAP, DAP'a alternatif olarak geliştirilen, DAP'ın

basitleştirilmiş ve getirdiği yük hafifletirilmiş şeklidir. DAP belirli bir süre kullanıldıktan sonra masaüstü (PC) uygulamaları için uygun olmadığına karar verildi. Bu sıralarda Internet yaygınlaşmaya ve OSI katmanları güncelliğini yitirmeye başlamıştı. Bu şartlar LDAP'ın geliştirilmesine yol açtı. LDAP DAP'tan farklı olarak TCP/IP katmanları üzerinde çalışır. DAP'ın nadir kullanılan karmaşık uygulamaları devre dışı bırakılmıştır ve daha basit ama hızlı bir protokol ortaya çıkmıştır. Bu protokol herhangi bir işletim sistemine veya platforma bağımlı değildir. Tek gerekli olan TCP/IP iletişim katmanının kurulu olmasıdır [7].

LDAP, mesajlaşma kaynaklı bir protokoldür. İstemci istek içeren bir mesajı oluşturur ve sunucuya gönderir. Sunucu isteği işledikten sonra sonuç veya sonuçları istemciye bir dizi mesaj olarak gönderir.

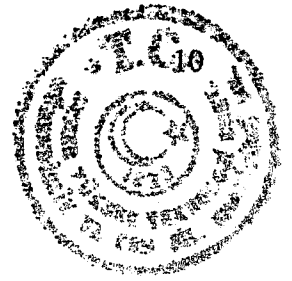
## 2.9. LDIF (LDAP Data Interchange Format)

Dizin bilgilerinin başka yerlere iletimi veya bir kayıt üzerinde yapılacak değişiklikleri belirtmek amacıyla kullanılan dosya şablonuna LDIF (LDAP Data Interchange Format) denir. Bu dosya yapısı, dizin ilk defa yüklenirken yada birçok kayıt bir anda değiştirilirken çok kullanışlıdır. Ayrıca LDAP uyumlu 2 dizin sunucusu arasında yoğun (bulk) bilgi değişimi yapılacaksa LDIF kullanılabilir.

LDIF formatı ilk olarak Mişgin üniversitesi LDAP uygulamasında geliştirilmiş ve kullanılmıştır [8]. LDIF'in ilk kullanım alanı dizin kayıtlarını tanımlamada olmuştur. Daha sonra, bu format dizin kayıtlarındaki değişiklikleri temsil etmek amacıyla genişletilmiştir. LDIF formatı, dizin bilgisinin taşınması veya dizin kayıtlarındaki bazı değişikliklerin tanımlanması için kullanılır. Aşağıda LDAP LDIF dosyası örnekleri görülmektedir.

Örnek 1: LDIF formatında basit iki girişli bir LDIF dosyası

```
version: 1
dn: cn=Koray Atsan, ou=Ürün geliştirme, dc=UEKAE, dc=gov
objectclass: top
objectclass: person
```



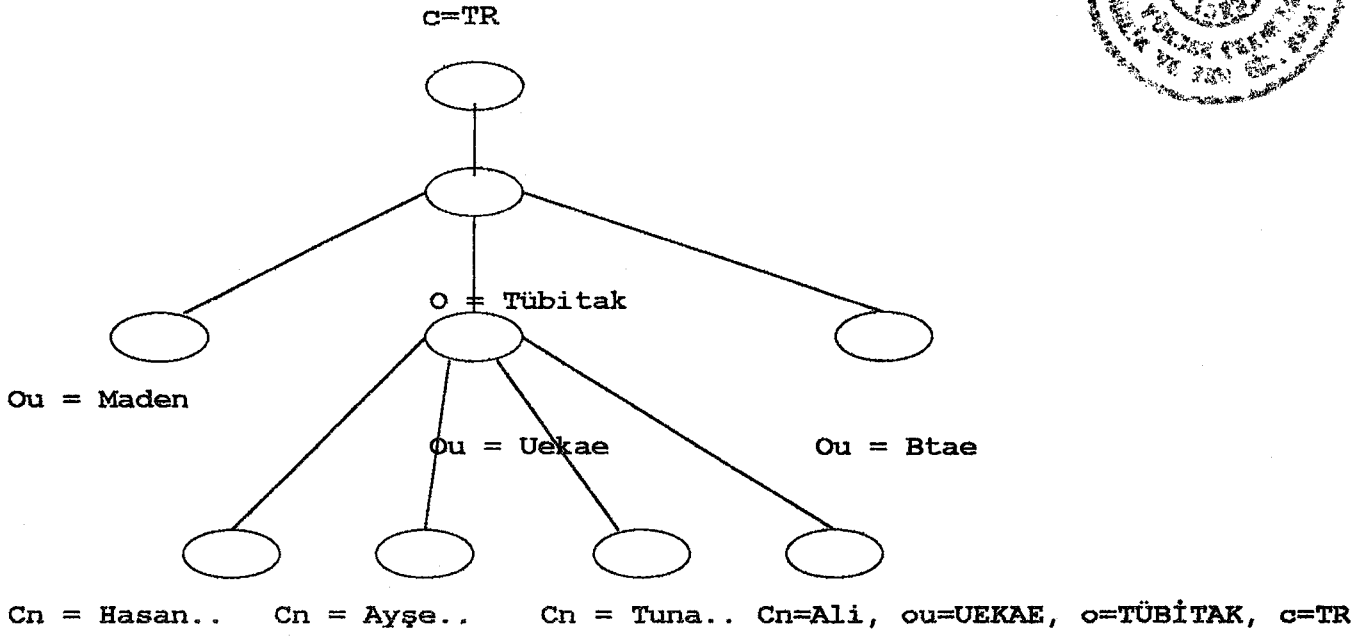
```
objectclass: organizationalPerson
cn: Koray Atsan
cn: Koray F. Atsan
cn: K.A.
sn: Atsan
uid: katsan
telephonenumber: +90 408 555 1212
description: Gemicilik hayranı
```

Örnek 2: Bir kontrol ile beraber bir kayıt değişikliği içeren bir LDIF dosyası

```
version: 1
# Kayıt Silme. İşlem LDAP v3 [5] da tanımlanan ağaç silme
kontrolü
# uyarınca gerçekleştirilecek. Kritiklik alanı
# "true" ve kontroldeğeri alanı
# yok,[5]gereğince.
dn: ou=Ürün geliştirme, dc=UEKAE, dc=gov
control: 1.2.840.113556.1.4.805 true
changetype: delete
```

## 2.10. LDAP İsimlendirme Modeli

LDAP isimlendirme Modeli, elinizdeki veriyi nasıl düzenleyip, ona nasıl erişebileceğinizi tanımlar. Ayrıca elinizdeki kayıtlardan (entry) nasıl yeni yapılar kurabileceğinizi de belirler. LDAP İsimlendirme Modeli en kısa olarak, elinizdeki veriyi ters ağaç yapısında düzenlemenizi gerektirir. Bu yapıda elinizdeki tüm veriyi tek bir kutu içinde toplayabileceğiniz gibi, kurumunuzun hiyerarşik yapısını yansıtan iç içe geçmiş kutular şeklinde de toplanabilir.

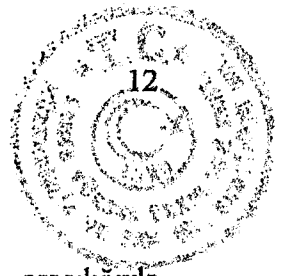


Şekil.2.1. LDAP isimlendirme modeli

Şekil.2.1’de LDAP isimlendirme modeline uygun, TÜBİTAK için tasarlanmış bir ters ağaç yapısı vardır. Bu yapıda ki kısaltmalara LDAP yapılarında çok sık rastlanır, kısaca:

|                         |                                                     |
|-------------------------|-----------------------------------------------------|
| C (Country)             | : İki haneli ülke kodu                              |
| O (Organization)        | : Kurumunuzun adı                                   |
| Ou(Organizational Unit) | : Kurumunuzun alt birim(ler)inin ad(lar)ı           |
| CN (Common Name)        | : Ağacın en ucunda yer alan kullanıcıların isimleri |
| DN (Distinguished Name) | : Tekil Kayıt Adı                                   |
| RDN(Relative DN )       | : Bağlı DN                                          |

Bir kaydın kesin konumu o kaydın Tekil Kayıt Adı ile bulunur. Her bir TKA tekil (unique)’dir, yani tüm dizinde aynı TKA’lı 2 kayıt olamaz. Bir kaydın en soldaki isimlendirme birimi onun Bağlı TKA’dır. Bir ebeveyn noktanın (node), altındaki tüm kayıtlar farklı BTKA’lara sahip olmalıdır. Bu kural yukarıdan aşağı tüm ağaca uygulanırsa her TKA’nın tekil olması sağlanmış olur.



## 2.11. LDAP İşlevsel Modeli

LDAP İşlevsel Modeli, Dizin üzerinde LDAP protokolü aracılığıyla gerçekleştirilebilecek işlemleri belirler. 3 tip işlem vardır

### 2.11.1. Sorgulama İşlemleri

İstemcilerin dizinde arama tarama yapmalarını sağlayacak işlemleri belirler. İki farklı işlem vardır

- **Ldap\_search()** : Ldap\_search() işlemi dizinde arama yapıp, kriterlere uyan kayıtları getirir.
- **Ldap\_compare()** : Ldap\_compare() işlemi dizindeki bir kayda ait bir özniteliğin değerinin elimizdeki değerle aynı olup olmadığını belirler.

### 2.11.2. Güncelleme İşlemleri

İstemcilerin dizinde ekleme, silme ve değişiklik yapabilmelerini sağlar. 4 ayrı işlem vardır:

- **Ldap\_add()** : Ldap\_add() dizine yeni bir kayıt eklemek için kullanılır.
- **Ldap\_delete()** : Ldap\_delete() dizinden bir kayıt silmek için kullanılır.
- **Ldap\_modifyDN()** : Ldap\_modify() dizindeki bir kaydın bazı özniteliklerini değiştirmek için kullanılır.
- **Ldap\_modify()** : Ldap\_modifyDN() bir kaydın dizindeki yerini değiştirmek için kullanılır.

### 2.11.3. Doğrulama ve Denetim İşlemleri

Bağlanma, bırakma vazgeçme. Bağlanma işlemi istemcinin, kimlik ve doğrulama kriterlerini sağlayarak, dizin'e kendini tanıttığını sağlar. Bırakma ise istemcinin bir oturumu sonlandırmasını sağlar. Vazgeçme işlemi ise, istemci'nin artık, daha önce yapmış olduğu sorgulama sonuçları ile ilgilenmediği'ni belirtmesini sağlar.

## 2.12. LDAP da Erişim Denetimi (Erişim Denetim Listeleri)

İstemcilerin kimlik doğrulaması, sadece kimliklerini tanımlama amacına yönelik değil aynı zamanda dizin sunucusundaki kayıtları görme, değiştirme, ekleme veya silmede hangi yetkilere sahip olduklarının da belirlenmesi amacını taşır [13]. Kullanıcı yetkileri atamak ve erişim denetimi kullanmak, dizin sunucusunun kullanıcıların sadece yetkileri olan bilgileri görmelerinden emin olmasını sağlar. Varsayılan olarak ağaç yapısındaki bütün veri erişime kısıtlanmış ve bütün veriyi görme, karşılaştırma, değiştirme ve okuma hakkı Dizin sunucusu yöneticisine (LDAP sunucusu süper kullanıcı hesabı) verilmiştir.

LDAP'ta Erişim Denetim Listeleri (ACL), kayıtların içindeki "ACI" özniteliği kullanılarak oluşturulur. Asıl sözdizimi, herbir dizin üreticisinin kendi erişim denetimi modelini kullanmasından dolayı değişmektedir. Örneğin Netscape Dizin sunucusunda ACI (ACL) kayıtları LDAP sunucusunda genellikle denetimi sağlanacak ağaç yapısının kökünde saklanırken, Michigan Üniversitesi/OpenLDAP sunucusunda genellikle LDAP sunucusu başlangıç dosyasında yer alan harici bir dosyada saklanırlar. Temel LDAP ACL tanımı aşağıda görülmektedir.

```
<access directive> ::= access to <what>
                        [by <who> <access>] +
```

```
<what> ::= * | [dn=<regex>]
           [filter=<ldapfilter>]
           [attrs=<attrlist>]
```

```
<who> ::= * | self | dn=<regex> |
           addr=<regex> |
           domain=<regex> |
           dnattr=<dn attribute>
```

```
<access> ::= [self]none | [self]compare |
              [self]search | [self]read |
              [self]write
```

Örnek iki ACL kaydı aşağıdaki şekildedir :

```

Access to filter = "cn =<haber_grub ismi>"
  Attr = yetkiliBirim, yetkiliKişi
by kendi write
by dnattr = sahip write
by dn = "cn = Yönetici, ou = Kişiler,
  o = <alan>, c= TR" write
by dn = "cn = <Haber Sunucusu IP konak adı>, ou =
Ağ Konakları,
  o = <alan>, c= TR" read
by addr = <Haber Sunucusu IP adresi> read
by *none ACL - 1

access to attr = YParola
by kendi write
by dn = "cn = Yönetici, ou = Kişiler,
o = <alan>, c= TR" write
by addr = <Samba Sunucusu IP adresi> read
by * none ACL - 2

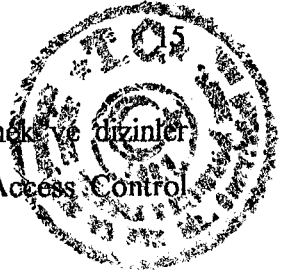
```

Birinci ACL, TKA (cn) değeri <haber\_grubu ismi> ile başlayan özel haber grubunun LDAP kaydında tanımlanan *yetkiliBirim* ve *yetkiliKişi* LDAP özniteliklerine bütün erişimleri sınırlar. Yazma ve okuma erişimi *sahip* özniteliği ile tanımlanan ve diğer bir LDAP kayıt tka sına işaret eden kişiye atanmıştır. Yönetici tka sı ile ilişkili kişiye aynı erişim izni verilmiştir.

İkinci ACL, herhangi bir LDAP kaydında tanımlanan Yparola özniteliğine erişimi sınırlar. Bir önceki ACL sadece bir kayıt ile sınırlı idi. Burada kayıt tka'sı ile tanımlanan kullanıcı (kendi) ve LDAP yöneticisi yazma ve okuma erişim yetkisine sahiptir. Parola verisini kullanan Samba sunucusuna sadece okuma erişimi izni verilmiş ve IP adresi byaddr ile belirtilen IP adresi ile uyusmak zorundadır. Diğer bütün erişim girişimleri reddedilir.

Sınırlandırılmış LDAP kaydının sahibi, ACL sürecine alıştıktan ve tanıdıktan sonra, LDAP yöneticisi ilgili ACL den çıkarılır veya çıkarılmayabilir. Genellikle yeni bir servisin çalıştırılması ve eğer kayıt sahibi sınırlandırılmış verisini güncellemekte sorunlar yaşıyorsa güncellemeyi geri alma için eklenir.

Bu konuda LDAP için erişim denetim gereksinimlerini belirlemek ve dizinler arasında birlikte çalışabilmeyi sağlamak amacıyla IETF tarafından "Access Control Requirements for LDAP" RFC si [9] yayınlanmıştır.



## 2.13. LDAP API ve Programlama

LDAP'ı ilk geliştiren uzmanlar, eğer LDAP Sunucularına erişim için standart bir API (Application Protocol Interface) olursa LDAP'ın daha da yaygınlaşacağını düşündüler ve University of Michigan'da çalışan bu uzmanlar bir "*C dili programlama kütüphanesi*" oluşturdular. LDAP yaygınlaştıkça değişik dillerde kütüphaneler ve class'lar oluşturuldu ve programcılar artık standart haline gelen bu kütüphaneleri kullanarak Rehber erişimlerini sağlamaya başladılar. Gittikçe yaygınlaşan bu API sonunda standart oldu. LDAP'ın birçok programlama dilinde API'si gerçekleşmiştir. (C, JAVA, Perl...). Bu dokümanın sonunda ek olarak "*LDAP C API*" kullanılarak yazılmış bir C erişim kodu bulunabilir.

## 2.14. OpenLDAP

OpenLDAP projesi, sağlam, ticari seviyede olgun, tüm beklenen LDAP özelliklerine sahip ve açık kodlu LDAP uygulamaları ve geliştirme araçlarının geliştirilmesine yönelik ortak bir girişimdir. Proje kar amacı olmayan ve temel amacı açık kodlu LDAP yazılım ve bilgilerinin geliştirilmesi olan OpenLDAP kuruluşu tarafından yürütülmektedir.

Geliştirilen yazılım ve dokümanlar arasında ;

- OpenLDAP Sürüm 2.1.2
  - Slapd (Stand alone LDAP server)
  - Slurpd (Stand alone LDAP replication Server)
  - Libraries (implementing the LDAP protocol, and utilities, tools, and sample clients. )
- JLDAP (LDAP Class Libraries for Java)
- JLDAP Dokümantasyonu
- OpenLDAP 2.0, 2.1 Administrators guide

OpenLDAP'ın avantajı açık kod olması ve veritabanı erişimine "Backend Module" desteği sayesinde izin veriyor olmasıdır. Bu destek OpenLDAP 2.0 sürümünden itibaren verilmektedir. Backend Modüller, slapd'nin fonksiyonelliklerini geliştirmek üzere koşma zamanında (runtime) slapd'ye entegre olan ortak nesne veya dll ler olarak tanımlanabilirler.

Backend Modüller statik olarak derlenirler ve konfigürasyonları yapılarak slapd'ye eklenirler.

Örnek olarak :

```
./configure -- enable module --enable password -- with passwd -- module =  
dynamic
```

komutu yürütülerek modül <prefix>/libexec/openldap ın altına yüklenir.

Aynı zamanda slapd.conf dosyasında modülün tanıtımı yapılmalıdır.

```
Modulepath/opt/openldap/libexec/openldapmoduleload back_passwd.la
```

## **2.15. Dizinlerin Güvenlik Hizmetlerindeki Yeri**

Günümüzde, veri ve ağ güvenliği hizmetlerinde şifreleme ve imzalama işlemleri sıkça kullanılmaktadır. Şifreleme ve İmzalama işlemleri bir bit dizisi olan anahtarlar vasıtasıyla gerçekleştirilir. İki tip şifreleme vardır: Simetrik ve Asimetrik. Simetrik şifrelemede, hem şifreleme hem de şifre çözme için tek anahtar kullanılır. Asimetrik şifrelemede ise 2 anahtar vardır. Açık anahtarla şifrelenen mesaj özel anahtarla açılır, özel anahtarla şifrelenen mesaj da açık anahtarla açılır. Bu anahtarlardan açık olanlar sertifika denilen bir yapı içerisinde bütünlüğü korunarak saklanabilir. Sertifikaları, Sertifikasyon Makamları (SM) imzalar ve herkesin ulaşip çekebileceği bir yere bu sertifikaları yerleştirir. İşte bu noktada Dizin Hizmetleri ve LDAP devreye girer. SM tarafından imzalanan sertifikalar Dizine yerleştirilir ve kullanıcılar ihtiyaç duydukları sertifikaları dizinden rahatça çekerler. Bu yapıya ve anahtar ve sertifikaların idaresi işine genel olarak PKI (Public Key Infrastructure) denir. Şimdi şu soru sorulabilir: Sertifikalar neden dizinde saklanır?

Sertifikalar SM tarafından bir kere üretilir ve ömürleri boyunca yüzlerce, binlerce defa kullanıcılar tarafından dizinden çekilip okunurlar. Yani çok okunup az güncellenen sertifikalar dizine uygun bir uygulamadır.

Birçok farklı platformlarda çalışan farklı coğrafi konumlarda bulunan kullanıcıların sertifikalara erişimleri için standart bir protokolü aracı olarak kullanmaları gereklidir. LDAP bu iş için biçilmiş kaftandır.

Dizinlerin dağıtık yapıda kurulabilmesi ve rahatça çoklanabilmesi ayrı bir avantajdır çünkü sertifika içeren bir dizin coğrafi olarak dağıtılıp çoklanır ve kullanıcılara daha yakın olup performans artışı sağlayabilir.

Sertifikaların rahat çekilebilir ve standart bir protokol aracılığıyla erişilebilir olması çok önemli bir olaydır çünkü güvenli iletişimde sayısal sertifikalar ve PKI dan faydalanan şirketler için sertifikaların idaresi çok zordur. PGP (Pretty Good Privacy) sisteminde de sertifikalar kullanılır ama genelde güven kişilerin ikili olarak birbirlerini tanımaları üzerine kuruludur. Eğer PKI'ı daha büyük kurum ve kuruluşlarda kullanmak istiyorsak, sertifika değişimini basitleştirmek ve bir standarda bağlamak lazımdır. İşte dizinler bu yönleriyle PKI ve sertifika depolanmasını gerektiren diğer güvenlik uygulamaları için çok elverişli ortamlardır.

### 3. LDAP DOĞRULAMA VE GİZLİLİK TEKNİKLERİ

LDAP dizin sunucusuna erişimde çeşitli doğrulama ve gizlilik teknikleri kullanılabilir. LDAP v2 protokolünde sadece 3 çeşit doğrulama tekniği desteklenmişti : Anonim, Basit(Açık) ve Kerberos v4. LDAP v3, değişik doğrulama tekniklerini'nin kullanımını desteklemek amacıyla RFC 2222 de tanımlanmış SASL doğrulama çerçevesini kullanır [14]

#### 3.1. Doğrulama, Doğrulama Bilgileri , Kimlik

Doğrulama bilgileri, bağlantı kurmak isteyen bir tarafın (bir kullanıcı) diğer bir tarafa (bir sunucu) kendi kimliğini ispatlamak amacıyla gönderdiği bilgilerdir. Doğrulama, bu bilgilerin üretilmesi, iletilmesi ve ait olduğu kimliğin belirlenmesidir. Doğrulama kimliği, doğrulama bilgilerinde bulunan isimdir. Doğrulama bilgileri kullanılan bir doğrulama mekanizmasına özel olabilir. Örneğin : X.509 sertifikaları, Kerberos biletleri, basit parolalar vb.

LDAP protokol işlemleri :

- Sorgulama işlemleri
- Güncelleme işlemleri
- Doğrulama ve Denetim

Biz bu protocol işlemleri içinde Doğrulama işlemi tekniklerini inceleyeceğiz :

##### 3.1.1. Anonim Doğrulama

Dizin kayıtlarını veya erişim denetimli öznitelikleri veya kayıtları değiştiren, dizin işlemleri için genelde istemci doğrulaması gerektirir. Bu işlemlerden herhangi birisini yapma ihtiyacı olmayan istemciler anonim doğrulama kullanırlar.

Anonim tabiri sözlük anlamıyla kimliği belirsiz demektir. Bazı durumlarda LDAP 'ta saklanan bazı bilgilerin tüm kullanıcılara açık olması söz konusu olabilir. Bu durumlarda kullanıcıların doğrulanmış olmasına bakılmaksızın bilgiler sunulur. Doğrulanmayan veya herhangi bir yetki olmaksızın doğrulanan kullanıcılara Anonim kullanıcı diyoruz. Bu tip kullanıcılar LDAP sunucusu tarafından doğrulanmamış varsayılırlar ve yetkileri buna göre düzenlenir.

### 3.1.2. Parola Tabanlı Doğrulama

Bir kullanıcı parolası özniteliğine sahip dizin kaydı olan bir kullanıcı, basit parola bağlantı isteğinde bulunarak kimliğini doğruladır. Sunucu, kullanıcı kaydındaki parola özniteliği ile kullanıcının göndermiş olduğu parolayı karşılaştırır. Eğer parolalar uyuşuyorsa sunucu sonuç kodu olarak başarılı tersi durumda ise başarısız cevabını döner. LDAP basit doğrulama seçeneği Internet gibi ağ veya iletim katmanı gizliliğinin olmadığı ortamlarda kullanılması uygun değildir.

### 3.1.3. Sertifika Tabanlı Doğrulama

Açık anahtar güvenli bir üçüncü taraf tarafından imzalanmış, Açık/Özel anahtar çiftine sahip bir istemci, bu anahtar çiftini, sunucu tarafından istendiği takdirde, dizin sunucusuna kimlik doğrulama amacıyla kullanabilir. Bu doğrulama işlemi için, kullanıcı sertifikası nesne alanı, kullanıcının dizin ismi ile aynı olmak durumundadır.

TLS yönteminde, sunucu istemciden bir sertifika ister. İstemci sunucuya sertifikasını ulaştırır. İstemci, bu sertifikayla ilintili olan özel anahtara sahip olduğunu ispatlamak için bir şifreleme işlemi gerçekleştirir. İstemci ve sunucu karşılıklı olarak bir şifreleme algoritması üzerinde anlaşmalıdırlar. Sunucu, istemcinin sertifikasının geçerli olduğunu onaylar. Sunucu istemcinin sertifikasının bilinen bir SM tarafından üretildiğini ve sertifika zincirinde herhangi bir SM'nin geçersiz veya iptal edilmiş olmadığını kontrol eder.

### 3.1.3.1. TLS 1.0 Protokolü (Transport Layer Security)

TLS, Internet üzerinde iletişim gizliliği sağlayan bir protokoldür [4]. İstemci ve sunucu arasında, dinleme, değiştirme ve mesaj sahteciliği ni önleyecek bir şekilde iletişim sağlar. 1996 da TLS çalışma grubu SSL v 3.0 I çıkardılar. Daha sonra 1999 da RFC 2246 ile beraber TLS Sürüm 1.0 çıkarıldı ve standartlaştı. TLS 1.0 ile SSL 3.0 arasında çok büyük farklar olmamasına rağmen, ikisi beraber çalışamayacak kadar farklılıklar vardır.

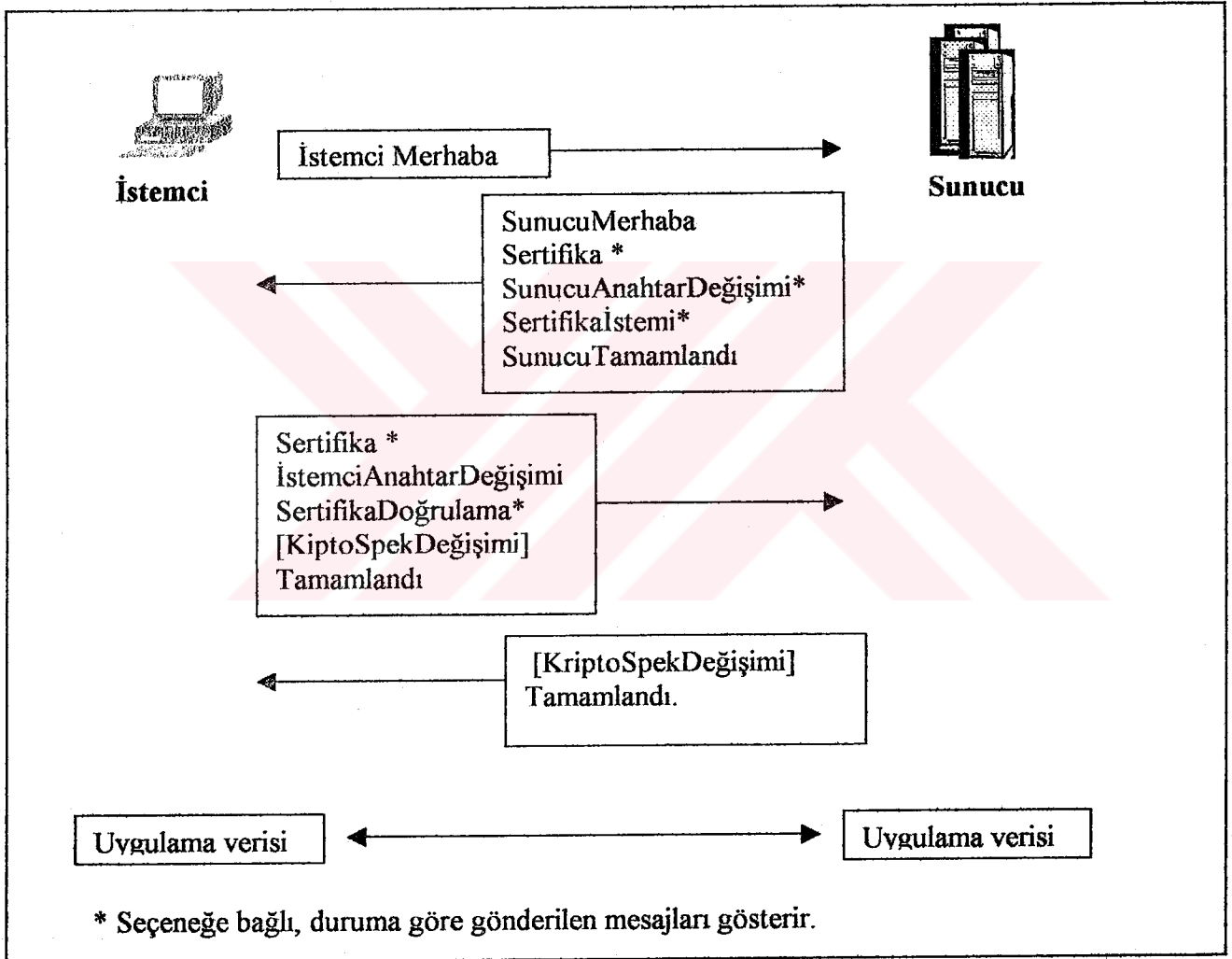
#### 1-TLS El Sıkışma Protokolü

Bir TLS istemci ve sunucusu iletişime ilk başladıklarında, protokol sürümü üzerinde anlaşılır, kriptografik algoritmaları seçerler, seçeneğe bağlı olarak karşılıklı doğrulama yaparlar ve açık anahtar şifreleme tekniklerini kullanarak paylaşılan oturum anahtarını oluştururlar.

TLS El sıkışma protokolü aşağıdaki adımlardan oluşur :

1. İstemci, sunucuya “İstemci merhaba” mesajı, kendi rastgele değeri ve desteklediği kriptotakımlarını gönderir.
2. Sunucu, “Sunucu merhaba” mesajı ve kendi rastgele değerini göndererek cevap verir. Sunucu sertifikasını istemci’ye doğrulaması için gönderir ve istemciden sertifikasını isteyebilir.
3. Sunucu “sunucu tamamlandı” mesajını gönderir. Eğer sunucu istemciden sertifikasını istedi ise, istemci sertifikasını gönderir.
4. İstemci, rastgele bir ön gizli anahtar üretir ve sunucunun sertifikasından aldığı sunucu açık anahtarı ile bunu şifreleyerek sunucuya gönderir.
5. Sunucu ön gizli anahtarı alır. Sunucu ve istemci, asıl gizli anahtarı ve ön gizli anahtara dayanan oturum anahtarlarını üretirler.
6. İstemci sunucuya, “kripto spesifikasyonunu değiştir” mesajı göndererek, mesajları özetleme ve şifrelemede yeni üretilen oturum anahtarlarını kullanacağını belirtir. İstemci “istemci tamamlandı” mesajını gönderir.

7. Sunucu “kripto spesifikasyonunu deęiřtir” mesajını alır ve kayıt katmanı güvenlik durumunu oturum anahtarlarını kullanarak “simetrik řifreleme” ye çevirir. Sunucu istemci’ye “sunucu tamamlandı” mesajını gönderir.
8. İstemci ve Sunucu kurdukları güvenli kanaldan veri deęiřimi yapabilirler. İstemciden sunucuya ve tersi istikamette giden bütün mesajlar oturum anahtarı ile řifrelenir.



Şekil.3.1. TLS El Sıkışma Protokolü

## 2-TLS Güvenli Oturumunun Yeniden Başlatılması

İstemci, yeniden başlatılacak oturumun oturum numarasını kullanarak « İstemci merhaba » mesajını gönderir. Sunucu, oturum önbelleginden uyan bir oturum numarası olup olmadığını kontrol eder. Eşit bir numara bulur ve sunucu

oturumu tekrar başlatabilecek durumda ise, oturum numarası ile bir « sunucu merhaba » mesajı gönderir. Eğer uyan bir oturum numarası bulunamazsa, sunucu yeni bir oturum numarası üretir ve TLS istemcisi ve sunucusu el sıkışma protokolünü uygularlar. İstemci ve sunucu, «kripto takımları değiştir» mesaj değişimini yaparlar ve «istemci tamamlandı», sunucu tamamlandı» mesajlarını gönderirler. İstemci ve sunucu bu aşamadan sonra uygulama veri değişimini güvenli kanal üzerinden yeniden başlatabilirler.

### 3.1.3.2. SASL (Simple Authentication and Security Layer)

SASL (Simple Authentication and Security Layer), bağlantı temelli protokollere doğrulama desteği sağlayan bir yöntemdir [8]. Bu yöntemi kullanmak için, bir protokol kullanıcıyı sunucuya tanıtacak ve doğrulatacak bir komut içerir. Seçeneğe bağlı olarak da diğer alt protokoller için bir güvenlik düzeyi seçilebilir. Bu komut SASL yöntemini tanıtmak için gerekli olan bir parametre içerir. SASL yöntemleri uzunluğu 1 den 20 ye kadar olan, büyük harfler, tireler veya alt çizgilerden oluşan karakter dizileriyle isimlendirilirler. Bu isimler IANA'ya kayıt ettirilmelidir.

Bir sunucu istenilen mekanizmayı destekliyorsa, bir doğrulama protokolü değişimini başlatır. Bu istenilen mekanizmaya özel bir dizi sunucu sorgulaması ve buna karşı istemci cevaplarını içerir. Bu sorgulamalar ve cevapları değişik uzunlukta ikilik sayı dizilerinden oluşur. Daha sonra, protokol bu ikilik sayı dizilerinin bağlantı üzerinden iletimi için nasıl kodlanacağını belirler.

Sunucu, doğrulama komutunu veya herhangi bir istemci cevabını aldıktan sonra , bir sorgu yapabilir ve başarısızlık veya başarı belirtebilir. Bunun nasıl yapılacağını protokol belirler. İstemci, sorgulamayı aldıktan sonra, bir cevap verir veya değişimden vazgeçebilir.

Doğrulama protokol değişimi sırasında, mekanizma istemciden sunucuya yetkilendirme kimliği göndererek (kullanıcı kodu) doğrulamayı gerçekleştirir ve mekanizmaya özel güvenlik seviyesi kullanımı konusunda görüşür. Eğer güvenlik



seviyesi kullanımı üzerinde anlaşılırsa, mekanizma her iki tarafın alabileceği maksimum kriptolu metin belleğini belirtir.

İletilen yetkilendirme kimliği, istemci nin yetkilendirme bilgilerinden farklı olabilir. Bu vekil sunucular gibi bazı ara elemanların kendi bilgilerine göre yetkilendirmelerine izin verirken filitreledikleri kimlik adına kendi istekleri doğrultusunda sunucudan yetki almalarını sağlar. Güvenlik seviyesi kullanımı üzerinde anlaşılırsa, bu bağlantı üzerinden aktarılan tüm müteakip verilere uygulanır.

Güvenlik seviyesi etkinleştğinde, protokol akışı güvenlik seviyesi tarafından işlenerek kriptolu belleğine koyulur. Her bir kriptolu bellek alanı verisi, bağlantı üzerinden 8 lik dizisi şeklinde aktarılır. Kriptolu bellek alanı büyüklüğü anlaşılan maksimum büyüklüğü geçmemelidir.

Bazı mekanizmalar, doğrulama protokol değişiminde ilk veri aktarımı istemciden sunucuya olduğunu belirtir. Eğer ilk istemci cevap parametresi verilmezse veya bir protokol doğrulama protokol değişimini başlatan komutun bir istemci ilk cevabı içermesine izin vermiyorsa, o zaman sunucu veri içermeyen bir sorgulama yapar. İstemcinin bu sorgulamaya cevabı ilk istemci cevabı olarak kullanılır.

Yine bazı mekanizmalar, sunucu sorgulama verisinin, değişimin başarılı olduğuna dair veri ile beraber gönderilmesini belirtir. Bu veri örneğin sunucunun kimliğini istemciye ispatlar.

Bu yöntem diğer bazı protokolleri kapsayan bir çerçeve yapısı olduğundan ELIAC ile karşılaştırması yapılmamıştır.

### **3.1.3.3. Kerberos v4, v5**

Kerberos v4, MIT’de geliştirilmiş ağ yönelimli bir kimlik doğrulama protokolüdür. Kerberos v4, kimlik doğrulama hizmetini « bilet » adı verilen veri yapısı kullanarak gerçekleştirir. Ağ üzerinde istemci/sunucu uygulamalarına simetrik



anahtar kriptografisi kullanarak kuvvetli doğrulama sağlamak üzere tasarlanmıştır. Internet, güvenliği olmayan bir ortamdır. Internet üzerinde kullanılan birçok protokol güvenlik sağlamamaktadır. Parolaları çalmaya yarayan bazı araçlar korsanlar tarafından kullanılmakta, bu nedenle ağ üzerinden açık halde parola ileten uygulamalar güvenlik açısından zayıf duruma düşmektedirler. Daha kötüsü ise diğer istemci/sunucu uygulamaları, istemci programının kullancısının doğrulanması amacıyla istemci programına güvenmektedirler.

Bazı siteler, ağ güvenliklerini sağlamak amacıyla güvenlik duvarı kullanmaktadırlar. Ne yazık ki güvenlik duvarları, "kötü adamların" dışarıda olduklarını varsayarlar ki bu çoğu zaman kötü bir varsayımdır. Çoğu zarar verici bilgisayar suçları içeridekiler tarafından gerçekleştirilirler. Güvenlik duvarları aynı zamanda kullanıcıların Intenete bağlantılarını da kısıtladıkları için dezavantajları vardır. Birçok yerde bu kısıtlamalar gerçek dışı ve kabul edilemezdir.

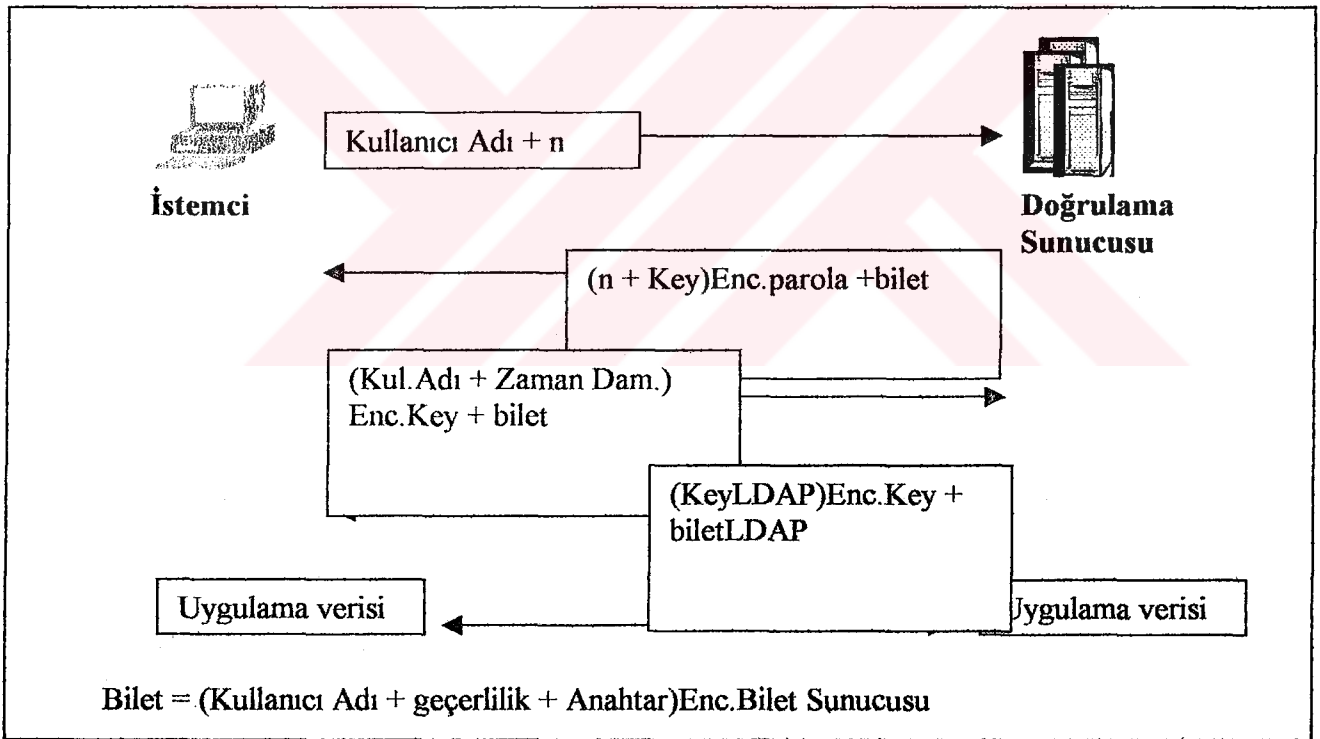
Kerberos v4, MIT tarafından bu ağ güvenliği sorunlarına çözüm üretmek amacıyla geliştirilmiştir. Kerberos protokolü, kuvvetli kriptografi kullanarak, güvenli bir ağ bağlantısı üzerinde istemci'nin sunucuya kimliğini doğrulamasını (ve tersi) sağlar [5]. İstemci ve sunucu kimliklerini doğrulamak amacıyla Kerberos'u kullandıktan sonra gizlilik sağlamak amacıyla bütün iletişimlerini şifreleyebilirler. Kerberos doğrulama hizmeti çeşitli üst protokoller (LDAP,FTP vs.) için kullanılabilir. Kerberos v4'ün ücretsiz bir gerçekleştirilmesi MIT'de bulunmakla birlikte birçok ticari üründe de bulunabilmektedir.

## 1-Kerberos Protokolü

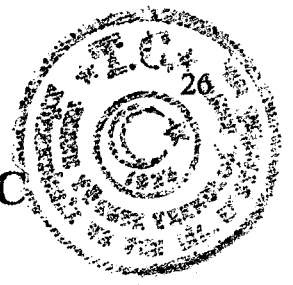
1. Kullanıcı ismini ve rasgele ürettiği bir sayıyı doğrulama sunucusuna gönderir. Burada dikkat edilmesi gereken nokta, kullanıcının karşı tarafa parolasını göndermediğidir. Rasgele sayı tekrarlamayı tespit ve önlemede kullanılır.
2. Doğrulama sunucusu kullanıcının parolasının bir kopyasına sahiptir. Bu parolanın özetini alarak DES tarafından kullanılmak üzere 56 bitlik simetrik anahtar oluşturur. Sunucu kullanıcıya döndürmek üzere bir oturum anahtarı üreterek DES anahtarı ile bunu şifreler. Kullanıcı oturum anahtarına sadece

parolası ile erişebilir. Doğrulama sunucusu, aynı zamanda kullanıcıya bilet sunucusuna erişimi için bir bilet gönderir. Bu bilet aynı oturum anahtarını, kullanıcı adını ve geçerlilik zamanını içerir ve bilet ve doğrulama sunucuları tarafından bilinen bir anahtarla şifrelenir.

3. Kullanıcı bilet sunucusuna, LDAP sunucusuna erişmek için bir bilet istediğini belirten bir mesaj yollar. Bu mesaj oturum anahtarı ile şifrelenmiştir. Kullanıcı daha sonra bu bileti, bilet sunucusuna gönderir ve bilet sunucusu oturum anahtarı ve kullanıcı mesajını çözer.
4. Bilet sunucusu, kullanıcı ismi ve yeni ürettiği LDAP anahtarını içeren bir bilet üretir ve LDAP sunucusunun özel anahtarı ile şifreler. Yeni oturum anahtarı eski oturum anahtarı ile şifrelenerek kullanıcıya gönderilir.



Şekil.3.2. Kerberos Protokolü



## 4. INTERNET KATMANLARI ve ELIAC PROTOKOLÜ'NÜN YERLEŞİMİ

### 4.1 TCP/IP Güvenliği ve Internet Katmanları

Ağ güvenliğinin pratikte nasıl çalıştığının anlaşılabilmesi açısından Internet katmanlarının incelenmesi yararlı olacaktır. Internet modelinde, ISO/OSI model protokolünün bazı katmanları elenerek katman sayısı 4 e indirilmiştir.

- Uygulama katmanında çalışan protokoller;  
Telnet, FTP, HTTP, SMTP gibi protokollerdir.
- Ulaşım katmanında çalışan protokoller ;  
TCP ve UDP
- Internet katmanında IP protokolü
- Fiziksel Arabirim katmanında kullanılan  
protokoller ise kullanılan ağ teknolojisine göre değişmektedir.

|                  |
|------------------|
| Uygulama         |
| Ulaşım/Oturum    |
| Internet         |
| FizikselArabirim |

Şekil.4.1. Internet Katmanları

### 4.2 ELIAC'ın Katman Modelinde Yerleşimi

ELIAC Internet Katmanları modelinde Uygulama katmanında çalışır. (Şekil 4.2) Diğer uygulama protokollerine eş katman seviyesinde güvenlik hizmeti sağlar. Şekilde 4.2 de görüldüğü gibi SMTP ve HTTP gibi protokoller TCP katmanının üstünde çalışmaktadır ve SMIME ve PGP gibi protokoller de bunlara Uygulama katmanında bir üst seviyede güvenlik hizmeti sağlamaktadırlar.



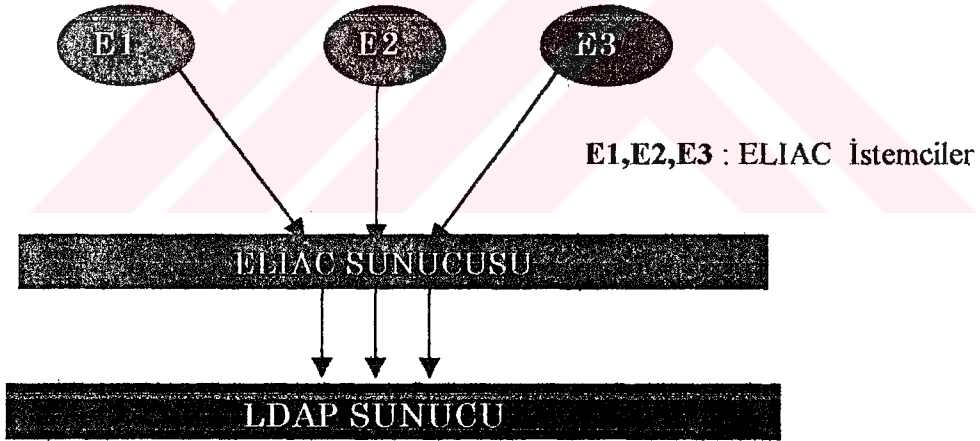
|          |            |     |       |      |
|----------|------------|-----|-------|------|
| Uygulama | SMIME      | PGP | ELIAC |      |
|          | SMTP       |     | LDAP  | HTTP |
| Internet | TCP (SSL)  |     |       | UDP  |
|          | IP (IPSEC) |     |       |      |

Şekil.4.2. ELIAC Katman Yerleşimi

## 5. ELIAC VE ALT PROTOKOLLERİ



ELIAC protokolü ile güvenli bir LDAP bağlantısının kurulabilmesi için Şekil 5.1 de görüldüğü gibi ELIAC protokolünü destekleyen bir istemci ve sunucuya ihtiyaç vardır. ELIAC istemcisinin ilklendirme ve El sıkışma alt protokolünü çalıştırması ile karşılıklı olarak Açık anahtarların ve Oturum anahtarının değiştirilmesini sağlar. Bir sonraki aşamada iletişim alt protokolü çalıştırılır ve iletişim bu şekilde kimlik doğrulanmış ve gizli bir şekilde devam eder. Kullanıcı istediği zaman ortak bilineni, ortak bilinen güncelleme alt protokolü ile güncelleyebilir. Bir ELIAC oturumu sonlandırılmadığı sürece belirli bir süre (örneğin 9 saat) ile geçerli kalır. Bu esnada istemci istediği takdirde daha önceki bir oturumu ilklendirme/el sıkışma protokolünün çalıştırılmasına gerek kalmadan tekrar başlatabilir. ELIAC protokolünde hata kotarma işlemi protokolün sadelik prensibine binaen oldukça basit bir şekilde gerçekleşmiştir. Protokolün herhangi bir aşamasında hata durumu, karşı tarafa bir hata kodu göndermek yoluyla kotarılır.



Şekil.5.1. ELIAC Genel Yapısı

Alt protokollerde kullanılan değişken isimlendirmeleri ve açıklamaları tablo 5.2 “Protokol Değişkenleri”nde bulunabilir.

### 5.1. İlklendirme ve El Sıkışma Protokolü

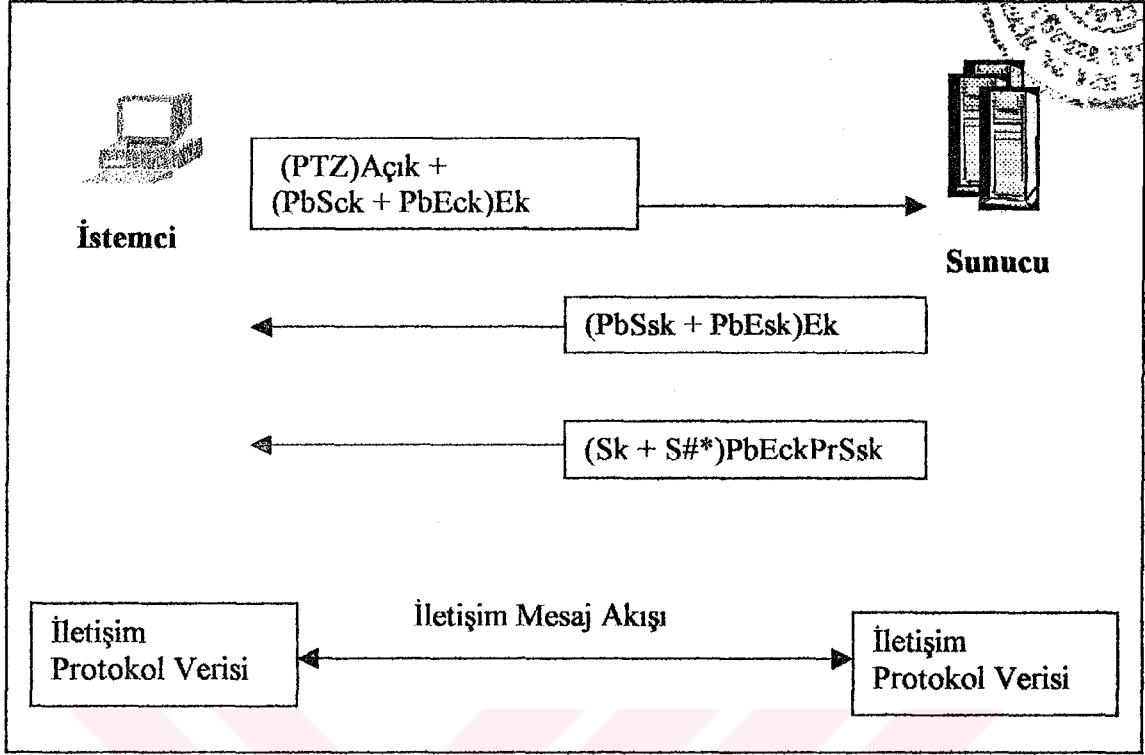
İlklendirme ve El Sıkışma protokolü, istemci'nin Doğrulama Sunucusuna (DS) oturum açma amacıyla bağlantı kurduğu ilk aşamada çalışan protokoldür.

Protokolün **ilkendirme** aşamasında istemci yeni ürettiği açık anahtarlarını, sunucu ise varolan açık anahtarlarını karşılıklı olarak birbirlerine iletirler. **El Sıkışma** aşamasında ise kimlik doğrulamalı ve gizli veri iletişimi başlatılmadan önce oturum anahtarının değişimi gerçekleştirilir.

Bu protokolde istemci ve DS'nin karşılıklı olarak ortak bir gizlilik parametresine (Parola vb.) sahip oldukları varsayılmaktadır. Protokolün akışı Şekil.5.2 de görülmektedir :

1. İstemci, Protokol numarasını (P), kendisini tekil olarak tanımlayan Tekil Kayıt Adını (T) ve Zaman Damgasını (Z) açık bir şekilde, İmzalama ve Şifreleme Açık Anahtarlarını ise parolasından üretilen gizli anahtar ile şifreleyerek DS'ye gönderir.
2. DS, kullanıcı TKA sını dikkate alarak, kullanıcının parolasından aynı gizli anahtarı üretir ve bu anahtarla kullanıcı açık anahtarlarına erişir.
3. DS, aynı gizli anahtarı kullanarak kendi açık anahtarlarını şifreler ve istemciye iletir.
4. istemci açık anahtarların şifresini çözerek DS'nin açık anahtarlarına erişir.
5. İstemci bu aşamdan sonra ilkendirme tamamlandı mesajını DS'e gönderir.
6. DS, daha önceden belirlenmiş olan kriptu algoritması ile kendisine gelen TKA adına oturum anahtarını üretir. Bu oturum anahtarını ve seçeneğe bağlı olarak oturum numarasını imzalama özel anahtarı ile imzalar ve TKA ile tanımlanan kullanıcının açık anahtarı ile şifreleyerek istemciye gönderir.

Eğer DS Oturum numarasını istemciye göndermemişse, Sunucu oturumun tekrar başlatılmasını desteklemiyor demektir.



Şekil.5.2. ELIAC İklendirme ve El Sıkışma Protokolü

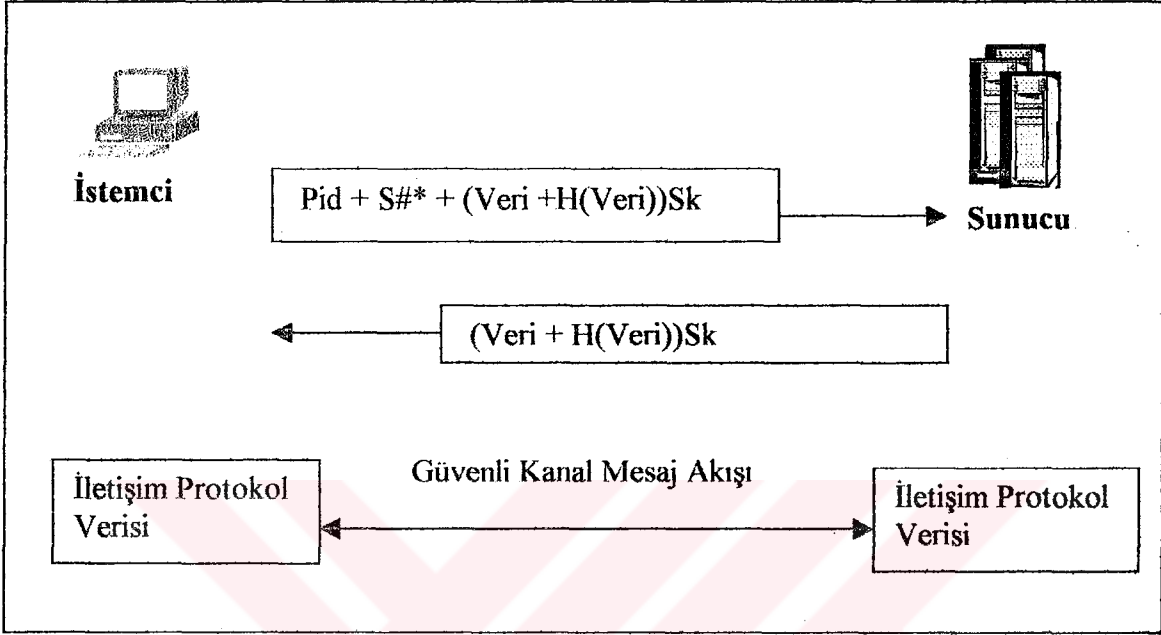
**S# (Oturum Numarası) :** İstemci, eğer sunucu ile olan iletişimine daha önceki oturumu canlandırıp, devam etmek isterse, iletişim protokolünün başında oturum numarasını sunucuya gönderir. Sunucu bu isteği kabul ederse iletişim eski oturum anahtarı üzerinden devam eder.

## 5.2. İletişim Protokolü

İletişim protokolü, bir oturum için el sıkışma protokolü tamamlandıktan sonra güvenli iletişim kanalının oluşturulduğu protokoldür. Protokol akışı Şekil 5.3 de görülmektedir :

1. İstemci, protokol numarasını ve daha önceki bir oturumu tekrar başlatacaksa oturum numarasını açık olarak, veri'yi ve verinin özetini oturum anahtarı ile şifreleyerek DS'ye gönderir.

2. DS, oturum anahtarı ile gelen veriyi çözer, verinin bütünlüğünün kontrolü amacıyla özetini kontrol eder.
3. DS'den istemciye veri aktarımı da, aynı şekilde gerçekleşir.

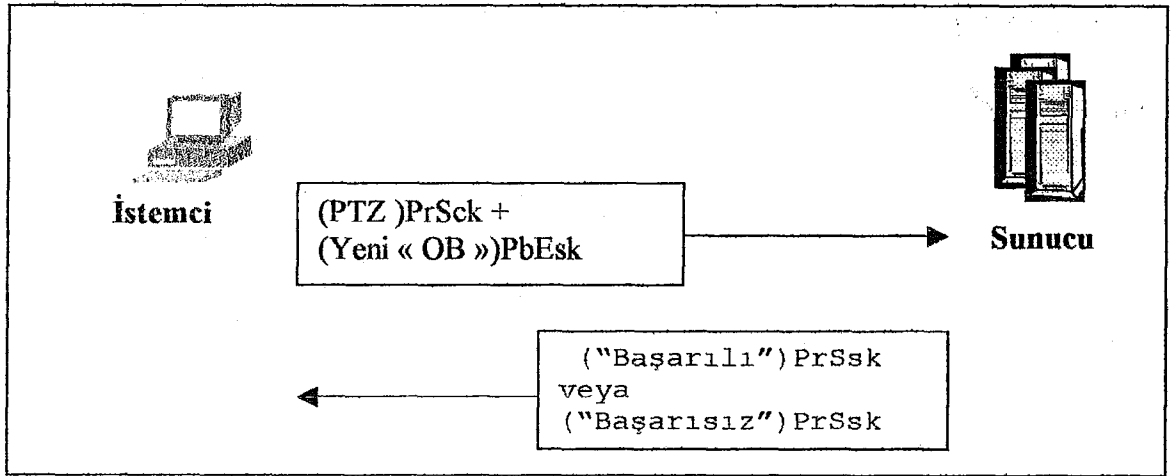


Şekil.5.3. ELIAC İletişim Protokolü

### 5.3. Ortak Bilinen Güncelleme Protokolü

Ortak bilinen güncelleme protokolü, istemci ile LDAP sunucusu arasındaki ortak bilinenin istemcinin isteğine bağlı olarak güncellenmesini sağlayan protokoldür. Her bir ayrı iletişim için üretilen anahtar çiftlerinin karşılıklı değişimi, ortak bilineneye dayanan simetrik şifreleme ile sağlandığından, ortak bilinenin belli aralıklarla güncellenmesi güvenlik açısından gereklidir. Protokol akışı Şekil 5.4 de görülmektedir :

1. İstemci, Protokol numarasını (P), kendisini tekil olarak tanımlayan Tekil Kayıt Adını (T) ve Zaman Damgasını (Z) özel imzalama anahtarı ile imzalayarak , yeni "Ortak bilinen" i ise DS'in şifreleme açık anahtarı ile şifreleyerek DS'e gönderir.
2. DS işlem sonucunu bir yanıt kodu ile istemciye bildirir.



Şekil.5.4. Ortak Bilinen Güncelleme Protokolü

## 5.4. Kullanılan Algoritmalar ve Anahtar Ömürleri

ELIAC protokolünün temelinde gerçekleştirme ve uygulama kolaylığı ve sadelik olması nedeniyle kullanılan kriptolojiler sabit tutulmuştur. Tablo 5.1 de çeşitli güvenlik fonksiyonları için kullanılan algoritmalar listelenmiştir.

Tablo.5.1. Algoritmalar ve Anahtar Ömürleri

| Fonksiyon      | Algoritma |
|----------------|-----------|
| Özetleme       | ▪ SHA –1  |
|                | ▪ 160 bit |
| Şifreleme      | ▪ RC4     |
|                | ▪ 128 bit |
| İmzalama       | ▪ RSA     |
|                | ▪ 2048    |
| Anahtar İletim | ▪ RSA     |
|                | ▪ 2048    |

Oturum Anahtarları her bir yeni oturum için yeniden üretilmektedir. Tekrar başlatılan oturumlarda ise önceki oturum anahtarı kullanılabilir. Oturum anahtarları en çok 9 saat süre ile geçerlidir. DS Özel/Açık anahtar çiftleri üretildikten sonra en geç 12 ay sonra yenilenmelidir.

## 5.5. Protokol Değişkenleri

ELIAC protokolünde kullanılan değişkenler ve tanımları aşağıdaki Tabloda görülmektedir :

Tablo.5.2. Protokol Değişkenleri

| Değişken                           | Açıklaması                                                                                                                                                                                                                                                                  |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>{Gp's} – Genel Parametreler</b> |                                                                                                                                                                                                                                                                             |
| <b>PTZ</b>                         | Pid+TKA+Z den oluşan veri yapısı                                                                                                                                                                                                                                            |
| <b>Pid(Protocol id)</b>            | Protokollere atanan tekil bir numarayı tanımlar.                                                                                                                                                                                                                            |
| <b>TKA (Tekil Kayıt Adı)</b>       | Kullanıcıyı tekil olarak tanımlayan herhangi bir kayıt adını tanımlar (e-mail, isim soyadı vb.)                                                                                                                                                                             |
| <b>Zaman damgası (Z)</b>           | Zaman damgası. Protokolün tekrarlama saldırılarına karşı korunması amacıyla kullanılan bir veri yapısını tanımlar. Bu değer istemci tarafından üretilir ve sunucuya gönderilir.<br>$Z = \text{Zaman} + H(\text{Zaman})$                                                     |
| <b>Yanıt Kodu</b>                  | Açık Anahtar güncelleme/iptal protokollerinde işlemin durumunu gösteren kod. Başarısız olduğu durumda protokol tekrarlanır.                                                                                                                                                 |
| <b>S# (Oturum Numarası)</b>        | Bir ELIAC oturumunun tekrar başlatılabilmesi için kullanılır. Sunucu taraf eğer oturumun tekrar başlatılabilmesini istiyorsa bu değeri istemciye döndürür. Her iki taraf ta bu değeri daha sonra kullanılmak üzere saklarlar. Oturum numaraları en fazla 9 saat geçerlidir. |
| <b>OB (Ortak Bilinen)</b>          | İstemci ile LDAP Sunucusu arasında tanımlı bir ortak bilineni tanımlar.                                                                                                                                                                                                     |
| <b>{Pb's} - Açık Anahtarlar</b>    |                                                                                                                                                                                                                                                                             |
| <b>PbSck</b>                       | Kullanıcı İmzalama Açık Anahtarı (Client Public Signing Key)                                                                                                                                                                                                                |
| <b>PbEck</b>                       | Kullanıcı Şifreleme Açık Anahtarı (Client Public Encryption Key)                                                                                                                                                                                                            |
| <b>PbSsk</b>                       | Sunucu İmzalama Açık Anahtarı (Server Public Signing Key)                                                                                                                                                                                                                   |
| <b>PbEsk</b>                       | Sunucu Şifreleme Açık Anahtarı (Server Public Encryption Key)                                                                                                                                                                                                               |
| <b>{Pr's} – Özel Anahtarlar</b>    |                                                                                                                                                                                                                                                                             |
| <b>PrSck</b>                       | Kullanıcı İmzalama Özel Anahtarı (Client Private Signing Key)                                                                                                                                                                                                               |
| <b>PrEck</b>                       | Kullanıcı Şifreleme Özel Anahtarı (Client Private Encryption Key)                                                                                                                                                                                                           |
| <b>PrSsk</b>                       | Sunucu İmzalama Özel Anahtarı (Server Private Signing Key)                                                                                                                                                                                                                  |
| <b>PrEsk</b>                       | Sunucu Şifreleme Özel Anahtarı (Server Private Encryption Key)                                                                                                                                                                                                              |
| <b>{S's} – Gizli Anahtarlar</b>    |                                                                                                                                                                                                                                                                             |
| <b>Ek</b>                          | Şifreleme (Simetrik) Anahtarı (Encryption Key)                                                                                                                                                                                                                              |
| <b>Sk</b>                          | Oturum (Simetrik) Anahtarı (Session Key)                                                                                                                                                                                                                                    |

## 6. ELIAC – TLS – KERBEROS – IPSEC PROTOKOLLERİNİN KARŞILAŞTIRILMASI

Güvenlik protokolleri incelenirken akla daha alt bir katmanda (IPSEC gibi) güvenlik hizmeti sağlayan bir protokol varken diğer protokollere ne gerek var sorusu gelebilir. Esasen bu soru, C++ gibi çok iyi bir programlama dili varken neden Java ya gerek var sorusuna benzetilebilir. Uygulamalar alt katman protokollerinin sağladığı güvenlik servislerini yeterli veya isteklerine tam cevap verecek şekilde bulmayabilirler. Buna örnek olarak IPSEC protokolünde kullanıcı bazında kimlik doğrulama servisi olmamasını ve diğer protokollerin bu özelliği sağlayabilmesini gösterebiliriz. Daha üst katmanlarda güvenlik hizmeti sağlayan protokoller ise uygulamalar için yeterince esnek veya basit olmayabilir. ELIAC protokolü bu bağlamda güvenli protokollere güvenlik sağlayan farklı bir seçenektir. Aşağıda ELIAC protokol özelliklerinin diğer protokoller ile karşılaştırılması maddeler halinde sıralanmıştır.

ELIAC protokolünün temel farklılıklarından biri sertifika yerine açık anahtar kullanmasıdır. Bu şekilde sertifika kullanımının getirmiş olduğu, sertifika yolu doğrulanması, sertifika geçerlilik durumunun belirlenmesi ve 3. taraflara güvenme gibi protokole karmaşıklık ve performans problemleri getiren unsurlardan kaçınılmıştır. Bunun yerine her iletişimin başlangıcında güvenli olarak istemci ve DS arasında açık anahtarlar değiştirilir.

Tablo 6.1 te sağladıkları güvenlik servisleri, Tablo 6.2 de ise özelliklerinin özet bir karşılaştırması görülmektedir.

### 6.1. Güvenlik Servisleri Karşılaştırma Tablosu

Aşağıdaki tablo'da tüm protokoller in bütün hizmetleri sağladıkları görülmektedir. Buna bir istisna olarak IPSEC protokolü alt katmanda çalıştığından dolayı makine bazında kimlik doğrulama sağlayabilmesine rağmen, kullanıcı bazında bu servisi sağlayamamaktadır.

ELIAC protokolünde inkar edememezlik güvenlik hizmeti sağlanmamaktadır. Bu hizmetin sağlanabilmesi için iletişim protokolünde imzalama desteğinin verilmesi gerekir. Bu destek, protokolü oldukça yavaşlatacağı için verilmesi uygun görülmemiştir. Ayrıca LDAP erişimlerinin yüksek bir oranda okuma erişimleri olduğunu düşünürsek, inkar edememezlik hizmetinin LDAP erişiminde çok önemli bir hizmet olmadığı aşıkardır. Yine LDAP'a erişimde Erişim Denetim Listeleri kullanımı vasıtasıyla bu eksiklik giderilebilir.

ELIAC protokolü ile diğer protokollerin sağladıkları güvenlik servislerinin karşılaştırması Tablo 6.1.' de görölmektedir :

Tablo. 6.1. Güvenlik Servisleri

| Güvenlik Servisi           | TLS | ELIAC | IPSEC | KERBEROS |
|----------------------------|-----|-------|-------|----------|
| Kullanıcı Kimlik Doğrulama | ✓   | ✓     | X     | ✓        |
| Gizlilik                   | ✓   | ✓     | ✓     | ✓        |
| Bütünlük                   | ✓   | ✓     | ✓     | ✓        |

Tablo 6.1.' de dikkat edilirse, inkar edememe güvenlik servisinin yeralmadığı göröülecektir. ELIAC protokolünde inkar edememezlik güvenlik hizmeti sağlanmamaktadır. Bu hizmetin sağlanabilmesi için iletişim protokolünde imzalama desteğinin verilmesi gerekir. Bu destek, protokolü oldukça yavaşlatacağı için verilmesi uygun görülmemiştir. Ayrıca LDAP erişimleri'nin yüksek bir oranda okuma erişimleri olduğunu düşünürsek, inkar edememezlik hizmeti'nin LDAP erişiminde çok önemli bir hizmet olmadığı aşıkardır. Yine LDAP'a erişimde Erişim Denetim Listeleri kullanımı vasıtasıyla bu eksiklik giderilebilir.

## 6.2. Protokol Özellikleri karşılaştırma Tablosu

ELIAC protokolünün burada sıralanan diğer simetrik anahtar veya sertifika kullanan protokollere göre önemli bir avantajı, anahtar yönetimine ihtiyaç duymamasıdır. Bu nedenle ELIAC, bir anahtar yönetim sunucusuna ihtiyaç

duymadan çalışabilir. ELIAC'da oturum ve açık anahtarlar oturuma ait olarak üretilmekte ve oturum sonlandığında geçerliliklerini yitirmektedirler.

ELIAC protokolü ile diğer protokollerin özelliklerinin karşılaştırması aşağıdaki tabloda görülmektedir :

Tablo.6.2. Protokol Özellikleri

| Protokol Özelliği               | TLS                                                                               | ELIAC                             | IPSEC                                                                             | KERBEROS                                                                     |
|---------------------------------|-----------------------------------------------------------------------------------|-----------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Çalışma Katmanı                 | Oturum                                                                            | Uygulama                          | İp                                                                                | Uygulama                                                                     |
| Şifreleme Teknolojisi           | Açık anahtar şifrelemesi kullanır                                                 | Açık anahtar şifrelemesi kullanır | Gizli anahtar şifrelemesi kullanır                                                | Gizli anahtar şifrelemesi kullanır                                           |
| Çalışma şekli                   | Sertifika tabanlı                                                                 | Açık anahtar tabanlı              | Simetrik anahtar tabanlı                                                          | 3.tarafa güvenilir                                                           |
| Gizli(Simetrik) Anahtar Üretimi | İstemci – Sunucu Karşılıklı                                                       | Sunucu                            | İstemci – Sunucu Karşılıklı                                                       | Sunucu                                                                       |
| Açık Anahtar Üretimi            | 3.taraf                                                                           | Kullanıcı                         | Kullanıcı veya 3. taraf                                                           | Yok                                                                          |
| Anahtar Yönetimi                | Anahtar iptalinde, geçersiz sertifikaların iptali için iptal sunucusu gerektirir. | Anahtar iptaline gerek yoktur.    | Anahtar iptalinde, geçersiz sertifikaların iptali için iptal sunucusu gerektirir. | Anahtar iptali, kullanıcıyı doğrulama sunucusunda geçersiz kılarak sağlanır. |
| *Karmaşıklık / *Gerçekleme      | Yüksek / Zor                                                                      | Düşük / Kolay                     | Yüksek / Zor                                                                      | Orta / Kolay                                                                 |
| Ortak bilinen                   | Gerçekmez                                                                         | Gerekir                           | Gerçekmez                                                                         | Gerekir                                                                      |
| *Esneklik                       | Yüksek                                                                            | Orta                              | Yüksek                                                                            | Orta                                                                         |

**Karmaşıklık**, protokolün yapısından kaynaklanan detay seviyesidir. Protokollerin yapısındaki diğer sistemlerle olan etkileşim, altprotokollerin ve veri yapılarının fazlalığı, işlem yoğunluğu karmaşıklığı arttıran faktörlerdir.

**Gerçekleme**, ise temelde protokolün karmaşıklığına göre, protokolün uygulanabilirlik derecesidir.

**Esneklik**, protokolün Internet katmanlı yapısında üst veya eşit seviyedeki diğer protokoller ile birlikte çalışabilme ölçüsüdür. Prensip olarak daha alt seviyede yeralan protokoller daha esnek yapıda olabilmektedirler.

## 7. SALDIRILAR VE GÜVENLİK ANALİZİ

Bu bölümdeki protokole yönelik saldırılar 2 ana başlık altında toplanmıştır.

- LDAP'a Yönelik Olası Saldırıları
- ELIAC'a Yönelik Olası Saldırıları

### 7.1. LDAP'a yönelik Olası Saldırıları

LDAP dizin sunucusuna yönelebilecek saldırılar aşağıda sıralanmıştır [3]:

1. Dizin üzerindeki veriye yetkisiz erişim
2. Başkalarının erişimlerini izleyerek, onların tekrar kullanılabilir erişim bilgilerine yetkisiz erişim.
3. Başkalarının erişimlerini izleyerek, dizin üzerindeki veriye yetkisiz erişim.
4. Verinin yetkisiz olarak değiştirilmesi
5. Dizin konfigürasyonunun yetkisiz olarak değiştirilmesi
6. Dizin kaynaklarının, yetkisiz veya fazlaca kullanılması
7. Yerine geçme yöntemiyle istemci'nin kandırılması : Bir istemci bağlantısının yönlendirilmesi veya iletimde olan verinin değiştirilmesi yoluyla, istemcinin bilginin dizinden gelmemiş olmasına rağmen ondan gelmiş olduğuna inandırılması.

Bu tehditlerden 1, 4, 5, ve 6 kötü niyetli istemcilerden gelir. 2, 3, ve 7 numaralı tehditler ise istemci ve sunucunun arasına giren veya sunucuymuş gibi davranan kötü niyetli 3 taraflardan gelir.

LDAP protokolü bu tehditlere karşı aşağıdaki yöntemler sayesinde korunabilir[10] :

1. SASL yöntemi vasıtası ile istemci doğrulaması
2. Başvuranın doğrulanmış kimliğine dayanan erişim denetimi yoluyla istemcinin yetkilendirilmesi
3. TLS protokolü ile veri bütünlüğünün sağlanması veya SASL yöntemiyle ile veri bütünlüğünün sağlanması
4. TLS protokolü ile araya girmenin engellenmesi veya SASL yöntemiyle veri şifrelemesi
5. Hizmet denetimleri yoluyla kaynakların sınırlanması
6. TLS protokolü veya SASL yöntemiyle sunucu doğrulanması

Yerine geçme tipi ip temelli saldırılar LDAP protokol kapsamı dışında yapılmaktadır.

### 7.1.1. Hizmet Aksatma

Hizmet aksatma saldırısı, LDAP sunucusunda bulunan çok sayıda nesne'nin hızlı bir şekilde değiştirilmesi ve çoğullama (replication) için yüksek öncelik (kritik) olarak belirlenmesi yoluyla olur. Bu tip saldırılar eğer LDAP sunucusunda anonim bağlantıya izin veriliyorsa, güvenlik duvarları kullanımı yoluyla tespit edilerek engellenebilir. Diğer bir engelleme yöntemi ise LDAP sunucusuna yazma erişimlerinde güçlü kimlik doğrulama kullanmak, kimliği doğrulanamayan yazma erişimlerine izin vermemektir. Saldırganın yazma erişim yetkisi var ise, güçlü kimlik doğrulama yöntemi ile saldırırganın kimliği tespit edilmiş olur.

### **7.1.2. LDAP Sunucuların Virüs Dağıtım Sistemi Olarak Kullanılması**

LDAP sunucular, yüksek bir erişim yoğunluğuna sahip olduklarından bir virüs dağıtım sistemi olarak kullanılmaları için ideal bir hedef olabilir. Bu tip saldırılar sunucularda koştan virüs tarayıcı uygulamalarla engellenmelidir.

### **7.1.3. Yoğun Saldırıları (Brute Force Attacks)**

Bu tip LDAP sunucusu saldırıları, sunucuda bulunan bütün kimliklerin tespiti ve bunlara bağlı parolalara bir sözlük dosyası yardımıyla yoğun saldırı şeklinde olur. Bu tip saldırılar sadece parola tabanlı kimlik doğrulama yöntemi uygulayan LDAP sunucuları için etkili olabilirler. Güçlü kimlik doğrulama yöntemlerinde parola kullanımı olmadığından bu tip yöntemlere karşı etkisiz kalırlar.

## **7.2. ELIAC'a Yönelik Olası Saldırıları ve ELIAC'ın Saldırıları Karşı Davranışı**

ELIAC protokolü iletişimde kimlik doğrulama, bütünlük ve gizlilik güvenlik servislerini sağlamaktadır. İlkendirme ve El sıkışma protokolü tamamlandıktan sonra istemci kimlik doğrulamalı ve gizli iletişimini devam ettirir. Kimlik doğrulama ilkendirme/el sıkışma aşamasında sağlanır. Burada DS, oturum anahtarını üreterek ilgili TKA ya gönderir ve TKA ile oturum anahtarının ilişkisini kurmuş olur.

Karşılıklı iletişimin gizliliğini sağlayacak olan Oturum anahtarı ve oturum numarası ilkendirme ve el sıkışma protokolünde kullanıcıya şifreli ve imzalı iletildiğinden saldırganın eline geçemez ve kaynağını doğrulanmış olur.

### **7.2.1. Aradaki Adam Saldırısı**

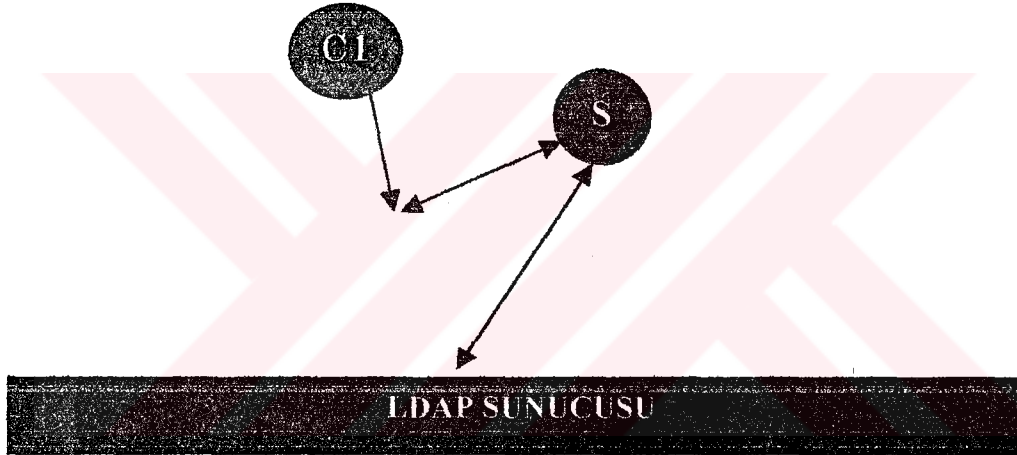
Aradaki adam saldırısı, karşılıklı iletişim halinde olan iki tarafın birbirleriyle iletişim kurduklarını sanmalarına karşın esasen aradaki saldırgan ile iletişimde

bulundukları saldırı tipidir. Bu tip saldırılar, aktif saldırı çeşitlerine girer ve saldırgan tarafından çok aktif bir çalışma gerektirir. Bu tip saldırılarda iletişim verisinin bozulması, değiştirilmesi ekleme/çıkarma yapılması söz konusu olabilir.

### 7.2.1.1. Araya Girme

Araya girme saldırısı, saldırganın istemci ve sunucu arasındaki iletişime müdahale ederek sunucu'ya istemcinin mesajları yerine kendi mesajlarını göndermek suretiyle istemci yerine geçtiği saldırı tipidir.

Şekil 7.1 de S saldırganının, C1 istemcisi yerine geçerek LDAP sunucusu ile yaptığı iletişim görülmektedir.

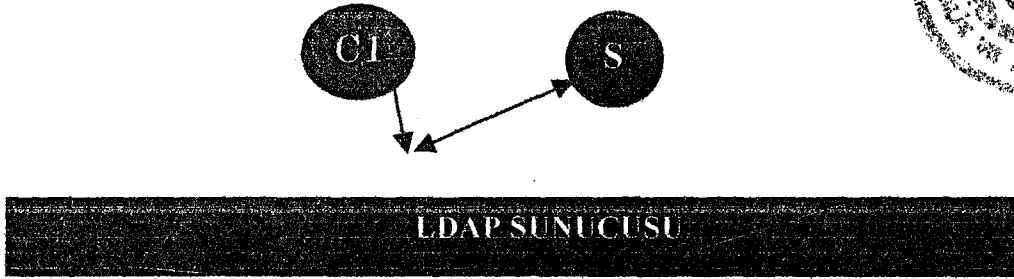
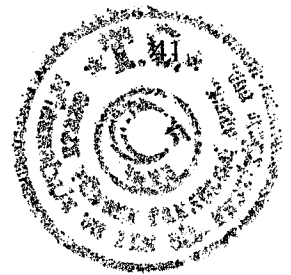


Şekil.7.1. Araya girme saldırısı

### 7.2.1.2. Engelleme

Engelleme saldırısı, saldırganın istemci ve sunucu arasındaki iletişime müdahale ederek sunucu'ya istemcinin mesajları yerine kendi mesajlarını göndermek suretiyle istemci yerine geçtiği saldırıdır.

Şekil.7.2.'de S saldırganının, C1 istemcisinin LDAP sunucusu ile olan iletişimini engellediği görülmektedir.



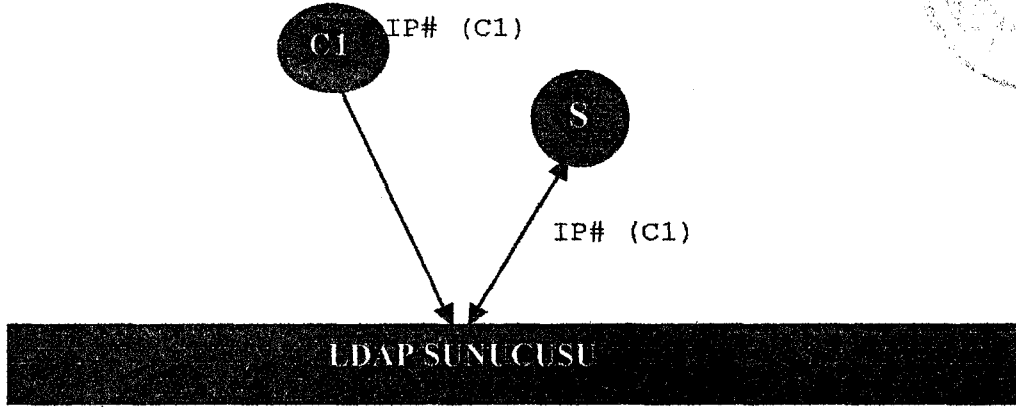
Şekil.7.2. Engelleme saldırısı

ELIAC da iletişim verisinin bütünlüğü iletişim protokolünde iletişim verisinin özeti alınarak sağlanır. Araya girme saldırısı yoluyla iletişim verisinin değiştirilmesi halinde değiştirilmiş verinin özeti ile orijinal verinin özeti birbirini tutmayacak ve bu şekilde saldırı anlaşılmış olacaktır. Engelleme saldırısı ise ağ güvenliğini sağlayıcı metodlar ile tespit edilerek, bunun önüne geçilmelidir.

### 7.2.1.3. IP Taklit Etme

IP taklit etme saldırısı, S saldırganının bir sunucunun güvendiği bir istemci IP adresinin yerine geçerek sunucuya paket gönderdiği saldırı tipidir. IP taklit etme saldırısını tespit etme ağ izleme araçları ile olur. Dış arabiriminizde kaynak ve hedef adreslerinin her ikisinde kendi yerel ağınıza ait bir paket varsa, IP taklit etme saldırısına uğruyorsunuzdur. IP taklit etme saldırısını engellemenin en iyi yolu dış arabiriminize gelen ve kaynak adresi olarak yerel ağınızdaki bir adrese sahip olan paketleri engelleyecek bir filitreleme yönlendiricisi kullanmaktır. Yukarıdaki çizimdeki senaryoda böyle bir filitreleme yönlendiricisi kullanıldığını varsaysak bile iletişim sertifika temelli ve imzalı yapıldığından dolayı IP taklit etme saldırısının tespiti yapılmış ve yetkisiz paketler engellenmiş olur.

Şekil.7.3.'de S saldırganının c1 istemcisinin IP adresini taklit ederek LDAP sunucusuna eriştiği görülmektedir.



Şekil .7.3. IP Taklit etme saldırısı

### 7.2.2. Tekrarlama Saldırısı

Önceden gönderilmiş iletişim verisinin (veya bir kısmının) kaydedilmesi ve tekrarlanması yolu ile doğrulama sistemlerine yapılan saldırılardır. Sabit olan herhangi bir doğrulama bilgisi parola veya diğer bir ortak bilinen kaydedilebilir ve daha sonra bir kaynağa erişim yetkisi sağlamak amacıyla kullanılabilir. ELIAC da iletişim halindeki veri şifrelendiğinden dolayı bu verinin kaydedilmesi ve tekrarlanması sonucu saldırgan hiçbir zaman verinin açık haline ulaşamaz.

Yine de saldırganın iletişim paketlerini tekrarlayarak sunucuyu meşgul etmemesi gerekmektedir. Bu amaçla ELIAC, “Z” zaman damgasını kullanır. Zaman damgası iletişimin gerçekleştiği zamanın açık hali ve özetinden oluşur. Bu şekilde sunucu gerçekleşen iletişimin güncel olduğunu doğrular.

## 7.3. Güvenlik Analizi

ELIAC protokolünde sertifika kullanımı olmadığından açık anahtarlar gizli bir şekilde kotarılmaktadırlar. Açık anahtarların değişiminde parola tabanlı şifreleme tekniği kullanılmakta ve bu şekilde açık anahtarların güvenliği sağlanmaktadır.

Örnek olarak tablo 7.3.1.1.’ de tanımlandığı şekilde ELIAC protokolünde ortak bilinen belirlenmiş kriterlere sahip olmalıdır. Bu kriterler LDAP dizin sunucusunda

saklanır. Ortak bilinenin güncellenmesi esnasında kullanıcının belirlediği yeni ortak bilinen bu kriterler dizin sunucusundan kontrol edilerek onaylanır.

### 7.3.1. ELIAC'ın Zafiyetleri

Her ne kadar ELIAC güvenli bir protokol olsa dahi, bu güvenliğin hassas olduğu noktalar vardır. Bu noktalar aşağıda sıralanmıştır :

#### 7.3.1.1. Ortak Bilinen Güvenliği

ELIAC protokolünde kullanılan ortak bilinen, güvenlik açısından belli bazı kriterlere sahip olmalıdır. Örnek bir güvenlik kriteri Tablo 7.3.1.1.' de verilmiştir. Bu veya benzeri güvenlik seviyesine sahip olan ortak bilinenler yeterince güvenli kabul edilebilirler. Daha düşük bir güvenlik seviyesinde kullanılan ortak bilinenler ise protokolün güvenliğini tehlikeye atacaktadırlar.

Tablo.7.1. Ortak Bilinen Kriterleri

| Ortak Bilinen                | Doğru                | Yanlış          |
|------------------------------|----------------------|-----------------|
| En az 6 karakter             | ✓ Parola13           | ✗ Parol         |
| En az bir büyük harf         | ✓ Parola13           | ✗ parola13      |
| En az bir küçük harf         | ✓ PaROLA13           | ✗ PAROLA13      |
| En az bir alfanümerik sayı   | ✓ Parola1            | ✗ Parola        |
| Aynı OB'nin tekrarlanamaması | ✓ Parola13→Karyola13 | ✗ Parola→Parola |

#### 7.3.1.2. Ortak Bilinen Ele Geçirilme Tehlikesi

ELIAC protokolünde bütün kullanıcıların ortak bilinenleri doğrulama sunucusunda saklanır. Bu noktada doğrulama sunucusunun güvenliği çok önemli ve hassastır. Doğrulama sunucusundaki ortak bilinenlerin ele geçmesi durumunda açık anahtarların, dolayısıyla da iletişimin güvenliği tehlikeye girecektir. ELIAC protokolünde bütün kullanıcıların ortak bilinenleri doğrulama sunucusunda saklanır. Bu noktada doğrulama sunucusunun güvenliği çok önemli ve hassastır. Doğrulama sunucusundaki Ortak bilinenlerin ele geçmesi durumunda açık anahtarların, dolayısıyla da iletişimin güvenliği tehlikeye girecektir. Ortak bilinene yönelik saldırılar şunlardır :

- **Dinleme**

Saldırgan iletişim hattını dinleyerek, iletişim halindeki verilerden yararlı bir bilgi edinmeye çalışır.

- **Tekrar**

Saldırgan daha önceki iletişimlerde gönderilmiş mesajları kaydeder ve tekrar yollar.

- **Aradaki adam Saldırısı**

Saldırgan, iletişim taraflarının mesajlarını keserek kendi mesajları ile değiştirir. Sunucuya yolladığı mesajlarda kullanıcı rolünü oynar, aynı şekilde terside geçerlidir.

- **Ortak bilinen tahmin saldırıları**

Saldırganın, genel olarak kullanılan ortak bilinenlerden oluşan bir sözlüğe sahip olduğu varsayılır. Temelde saldırı'nın sözlüğü kullanabileceği 2 yol vardır :

- **Çevrim Dışı Saldırı**

Saldırgan geçmiş iletişimleri kaydeder ve daha sonra bunları inceleyerek bir ortak bilinen arar. Böyle bir ortak bilinen bulunursa, saldırıyan bunun ortak bilinen olduğunu anlar.

- **Çevrim İçi Saldırı**

Saldırgan tekrarlayarak sözlükten bir ortak bilinen seçer ve kullanıcıyı taklit etmek amacıyla bunu kullanır. Eğer ortak bilinen başarısız olursa, saldırıyan ilgili ortak bilineni sözlükten siler ve yenisi ile saldırıyı tekrar dener.

Bu tip saldırıları önlemenin genel yolu, başarısız girişimlerin sayısını sınırlamaktır.

Ortak bilinenin sözlük saldırıları ile ele geçirilmesi tehlikesine karşı kullanıcıların başarısızlıkla sonuçlanan ortak bilinen kullanarak sisteme giriş denemeleri sınırlandırılmalıdır.

Ortak bilinenin korunabilmesi için en etkin yöntemlerden biri, açık anahtar tekniklerinin kullanımı yoluyla ortak bilinenlerin sunucu tarafında şifreli olarak saklanmasıdır. [12]

Bu tip güvenlik sistemlerinde, sanal ortam güvenliğinin yanısıra fiziksel güvenlik de ön plana çıkmakta ve fiziksel güvenlik, güvenlik prensipleri ile sağlanmaya çalışılmaktadır. Buna bir örnek olarak bir AAA(PKI) sisteminde çevrimiçi SM(CA) lerde saklanan şifreleme özel anahtarlarını ve SM lerin fiziksel olarak da korunmasını kapsayan Sertifika Prensiplerini gösterebiliriz.

### 7.3.1.3. ELIAC Hata Kodlaması

ELIAC protokolünde hata kodlaması çok basittir. Bu haliyle hata kodlaması bazı ek özellikler sağlanarak kuvvetlendirilebilir. Örneğin, karşılaşılabilecek hataların iyi dokümante edilmesi ve bu şekilde çözüme gidilmesi veya hata kodlamasının yazılacak uygulamalar vasıtasıyla ayrıntılandırılmasıdır.



## 8. SONUÇLAR

LDAP dizin sunucusu bağlantısında değişik güvenlik mekanizmaları kullanma imkanları vardır. Ne var ki kullanılacak mekanizma konusunda değişik faktörler gözönünde bulundurulmalıdır. SASL mekanizması diğer bazı mekanizmaların LDAP v3 protokolü ile çalışabilmesi için bir doğrulama çerçeve yapısı sağlar. Dolayısıyla diğer mekanizmaların kullanılmasında bir aracı olarak kullanılabilir. Eğer güvenli bağlantı amacı sadece kimlik doğrulama ve erişim denetimi sağlamak ise bu durumda Kerberos ve TLS gibi gizlilik sağlayan yöntemler yerine SASL vasıtasıyla bir doğrulama mekanizması kullanmak daha mantıklı olabilir. Bu sayede diğer yöntemlerin gizlilik sağlamak amacıyla beraberinde getirdiği karmaşıklık ve performans problemlerinden etkilenilmez. Amaç hem kimlik doğrulama hem de gizlilik sağlamak ise, bu noktada yapılacak seçimde diğer etkenler rol oynar. Örneğin, eğer iletişimde sertifikalar kullanılmak isteniyorsa ve performans problem olmuyorsa (az sayıda kullanıcı, donanım hızlandırıcıların kullanımı vb.) TLS kullanılabilir. TLS, istemci ve sunucular arasında bir “ortak bilinen” gerektirmemesinden dolayı da tercih edilebilir. Yine de TLS de sertifika kullanımı nedeniyle sertifikaların devamlı surette yönetiminin yapılması gerektiği unutulmamalıdır. Gizliliğin sağlanmasın da Kerberos da kullanılabilir. Kerberos kullanımında istemci ile sunucu arasında bir “ortak bilinen” e gereksinim vardır. Dolayısıyla Kerberos “ortak bilinen” kullanılan yerel sistemlerde kullanıma uygundur. Kerberos un kullanımı SASL çerçeve yapısı üzerinden yapılır. TLS kullanımı ise, RFC 2830 da tanımlandığı şekliyle LDAP v3 e “Extended Request” şeklinde entegre edilerek veya SASL üzerinden “External” mekanizması [11] ile sağlanır.

Karşılaştırmalı Tablodan da (Tablo.7.2) görülebileceği gibi, farklı güvenlik protokolleri birbirinden farklı özelliklere sahiptirler. Bu çalışma kapsamında geliştirilen ELIAC protokolünün bu protokollerden ayrılan en önemli özelliği diğer protokollerde olan karmaşıklığa sahip olmaması ve aynı zamanda yüksek bir güvenlik seviyesine sahip olmasıdır.

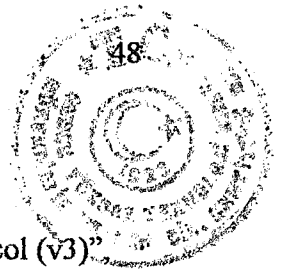
ELIAC, esas olarak LDAP’a erişimde kimlik doğrulama ve gizlilik servisleri sağlamayı hedefleyen bir protokol olmakla beraber, sade ve kolay gerçekleştirilebilir

olması sayesinde diğer uygulama katmanı protokolleri ile de (e-mail, http vb.) kullanılmak üzere esnek bir protokoldür.

ELIAC protokolünde gerekli değişiklikler yapılarak, protokolün ulaşım katmanında çalıştırılması mümkündür. Bu şekilde protokol daha esnek olabilecek ve sadece LDAP'a değil, diğer güvenlik protokolleri gibi çeşitli protokollere de güvenlik desteği verebilecektir. Aynı zamanda protokolün performansı da daha yüksek bir hale getirilebilecektir. Ne var ki, bu uygulama, protokolün gerçekleştirilmesini zorlaştırdığından dolayı, bununla ilgili çalışma, protokolün bir sonraki versiyonuna bırakılmıştır.



## KAYNAKLAR



- [1] M. Wahl, T. Howes and S. Kille, "Lightweight Directory Access protocol (v3)" RFC 2251, December 1997.
- [2] J. Myers, "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.
- [3] M. Wahl, H. Alvestrand, J. Hodges and R. Morgan, "Authentication Methods for LDAP", RFC 2829, May 2000.
- [4] T. Dierks and C. Allen, "TLS v 1.0 (Transport Layer Security)", RFC 2246, January 1999.
- [5] S. Miller, C. Neuman, J. Schiller and J. Saltzer, "Section E.2.1: Kerberos Authentication and Authorization System", M.I.T. Project Athena, Cambridge, Massachusetts, December 21<sup>st</sup>, 1987.
- [6] Y. Amir, C. Danilov, M. Miskin-Amir, J. Stanton, C. Tutu, "On the performance of Wide-Area Synchronous Database Replication", Technical Report, November 18<sup>th</sup>, 2002.
- [7] Postel J. , "Transmission Control Protocol" , RFC 793, October 1981.
- [8] G. Good, "The LDAP Data Interchange Format (LDIF) Technical Specification", RFC 2849, June 2000.
- [9] E. Stokes, D. Byrne, B. Blakley, P. Behera, "Access Control Requirements for LDAP", RFC 2820, May 2000.
- [10] Miller, S., Neuman, C., Schiller, J., and J. Saltzer, "Section E.2.1: Kerberos Authentication and Authorization System", M.I.T. Project Athena, Cambridge, Massachusetts, December 21<sup>st</sup> , 1987.
- [11] J. Hodges, R. Morgan, M. Wahl, "Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830, May 2000.
- [12] Shai Halevi, Hugo Krawczyk, "Public-key cryptography and password protocols" 5<sup>th</sup> ACM Conference on Computer and Communication Security pages 122 - 131 , ACM Press, 1998.
- [13] David W Chadwick University of Salford England "Secure Directories", Proceedings of the NATO Advanced Networking Workshop on Advanced Security Technologies in Networking, Portoroz, Slovenia", June 2000.
- [14] Vesna Hassler Technical University of Vienna "X.500 and LDAP Security A Comparative Overview" IEEE Network November/December 1999.



## ÖZGEÇMİŞ

F. Koray ATSAN, 6 Ocak 1974 de Yalova' da doğdu. Orta ve lise öğrenimini Antalya'da tamamladı. Yüksek öğrenimine Budapeşte Teknik Üniversitesi Elektrik ve Enformatik Müh. bölümünde 2 sene devam ettikten sonra İ.T.Ü. Kontrol ve Bilgisayar Müh. bölümünden 16 Şubat 2000 tarihinde mezun oldu.

Bu tarihten itibaren TÜBİTAK Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsünde araştırmacı olarak görev yapmaktadır. Çalışma konuları arasında PKI güvenlik altyapısı, mesajlaşma güvenliği ve güvenlik protokolleri bulunmaktadır.

