

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

YAZMA RİTMİ YÖNTEMİYLE BİLGİSAYAR
VE AĞ GÜVENLİK SİSTEMİ TASARIMI

Aykut GÜVEN
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

98283

TEZ DANIŞMANI
Yrd.Doç.Dr. İbrahim SOĞUKPINAR

GEBZE
2000

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMAN İSTİFON MERKEZİ



Aykut GÜVEN'in tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 18/07/2000 tarih ve 2000/16 sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği** Anabilim Dalı YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE : İbrahim SOĞUKPINAR, Yrd.Doç.Dr., GYTE
(Tez Danışmanı)

ÜYE : E.Murat ESİN, Yrd.Doç.Dr., GYTE

ÜYE : Ertuğrul KARAÇUHA, Yrd.Doç.Dr., GYTE

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 18/07/2000 tarih ve 2000/17 sayılı kararı.



ÖZET

Bilgisayar sistemlerinin güvenliğini sağlamak üzerine yapılan çalışmalar bilgisayar biliminde oldukça önemli bir yer tutmaktadır. Bilgisayarlar insanların hayatlarını daha kolaylaştırıp günlük yapılması gereken sıradan işlemleri hızla gerçekleştirerek çalışma verimini artırırlar. Özellikle günlük kullanımdan çok önemli ve gizli işlerin yapıldığı sistemlere kadar bilgisayar güvenliği önemli bir yer tutmaktadır.

Bilgisayarların güvenliğini sağlamada geleneksel olarak başvuru olan yöntem sistem üzerinde kullanıcıları tanımlayıp onlara hak vermek yönündedir. Bu klasik yöntemin kullandığı rahatlığa rağmen sisteme kayıtlı herhangi birisinin kullanıcı adı ve parolasını bilen yetkisiz bir kullanıcı, sisteme girerek sahip olmaması gereken bilgilere ulaşabilir. Özellikle askeri yada finansal amaçlarla kullanılan bu tür sistemler mümkün mertebe korunması gereken sistemlerdir. Bu tür sistemlere yetkisiz girişler tehlikeli ve onarılması güç sorunlar ortaya çıkartabilecektir.

Bilgisayar sistemlerinin güvenliğini sağlamada en fazla başvuru olan yöntemlerden birisi kullanıcıların biyolojik özelliklerini bilgisayar sistemine öğretmekten geçer. Yani istemci bilgisayarlarda yer alan sensörler kullanıcıların biyolojik özelliklerini algılayarak kullanıcıları tanırlar. Örneğin bir klavye üzerine yerleştirilmiş olan parmak izi sensörü bu tür bir sisteme iyi bir örnektir. Bu tür sistemler biometrik sistemler olarak adlandırılırlar. Biometrik sistemlerin tek dezavantajları , bu tür sistemleri kurmak için gerekli olan yüksek maliyetlerdir. Bu maliyet yüksekliklerinden dolayı bu tür sistemler çok geniş kullanım alanı bulamamaktadırlar.

Tezin konusu olan Yazmaritmi tabanlı tanıma sistemi ise herhangi bir fazladan donanım ihtiyacı hissetmeden bir güvenlik sistemi tasarımına izin vermektedir. Böylece maliyet açısından herhangi bir dezavantajı yoktur.

Tez çalışmasında kullanıcıların yazmaritimlerini tanıyarak sisteme giriş izni veren bir güvenlik yazılımı tasarımı yapılmıştır. Sistem kullanıcıları yazma

ritimlerinden ayırt edebilmektedir. Sahip olduđu öğrenme algoritması yardımıyla yeni kullanıcıları tanıyabilmekte aynı şekilde sahip olduđu yapay zeka ile sisteme daha önce kayıtlı olan kullanıcıların zamanla değışen yazma ritmi parametrelerini öğrenebilmektedir. Bu özellikleri sistemi dinamik bir sistem haline getirmektedir. Yapılmış olan deneysel çalışmalar sistemin oldukça güvenli olduğunu göstermiştir. Elde edilen sonuçlar böyle bir sistemin endüstriyel amaçlı kullanımının diğeri biometrik sistemlere göre daha avantajlı olduđu sonucunu doğurmaktadır. Ayrıca bu çalışmada biometrik sistemlerin tanımlanması ve açıklamaları verilmiş bir biometrik tanımlayıcı olarak yazmaritmi tabanlı güvenlik sistemlerinin tanımı yapılmıştır.

G.Y.T.E Bilgisayar Mühendisliğı bölümünde Yazmaritmi tanıma sistemi için geliştirilen algoritma tanıtılmış ve YARDS (Yazma Ritmi Doğrulama Sistemi) hakkında geniş bir bilgi verilmiştir. Bu konuda yapılmış olan deneysel sonuçlar anlatılmıştır.



SUMMARY

Studies which have been done in computer science to protect computer systems are very important. Computers make peoples' lives easy and increase productivity but it is difficult to make them secure.

To design computer security systems the traditional way is to create a username and password for all users. Although this is a simple method it is not reliable. Someone who knows a username and password to log in to a system may access parts of it that one would not want them to see. This is very dangerous for important network systems in, for example, military and financial areas.

Biometric applications is the best solution to supply security for computer systems. Biometric systems teaches computers their users' biological differences. For example, a finger print sensor can be placed over a keyboard. But, the most important disadvantage of these systems is their high cost.

In this thesis, security software which understands peoples' writing rythms has been designed. Explanations about biometric systems and biorytmic systems are included.

Algorythms on biorythm which have been designed in the G.Y.T.E Computer Engineering department will be explained and experimental results given.

TEŞEKKÜR

Bu tezin hazırlanma aşamasında sabırlı yardımlarından ve yol göstermelerinden ötürü değerli hocam Yrd.Doç.Dr. İbrahim SOĞUKPINAR'a, Tezi hazırlamam sırasında bütün huysuzluklarıma rağmen bana tahammül eden ve kesinlikle çay servisini kesmeyen sevgili eşim Özlem GÜVEN'e ve isimlerini burada sayamadığım herkese teşekkürlerimi sunarım.



İÇİNDEKİLER DİZİNİ

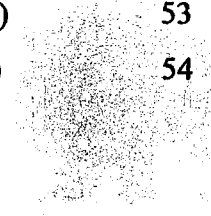
	Sayfa
ÖZET	IV
SUMMARY	VI
TEŞEKKÜR	VII
İÇİNDEKİLER DİZİNİ	VIII
ŞEKİLLER DİZİNİ	X
TABLolar DİZİNİ	XI
1. GİRİŞ	1
1.1. Bilgisayar Sistem Güvenliği Sağlamada Biometrik Yöntemler	2
2. BİOMETRİK GÜVENLİK SİSTEMLERİ	5
2.1. Retina Tanıma Sistemleri	5
2.2. Parmak İzi Tanıma Sistemleri	6
2.3. Yüz Tanıma Sistemleri	6
2.4. El Geometrisi Tanıma Sistemleri	7
2.5. İmza(Yazı) Tanıma Sistemleri	8
2.6. Ses Tanıma Sistemleri	8
2.7. Biometrik Sistemlerin Kuramsal Tasarım Yöntemleri	9
2.7.1. On-line Model	10
2.7.2. Off-line Model	11
2.8. Biometrik Sistemlerin Veri Kayıt Analizleri	11
2.9. Yazma Ritmi Sistemleri	13
2.10. Yazma Ritmi Sistemlerinde Sınıflama Algoritmaları	15
2.11. Minimum-Fark Sınıflandırması	17
2.12. Mahalanobis Sınıflandırması	18
2.13. Covariance Metriği	19
2.14. Mahalanobis Metriği	20
2.15. En Yakın Komşu Kuralı	21
2.16. Yazma Ritmi Sistemlerinde Vektörel Analiz Yöntemi	21
3. YAZMA RİTMİ DOĞRULAMA SİSTEMİ (YARDS) TASARIMI VE GERÇEKLENMESİ	23

3.1. İstemci Tarafı Yazılımı	23
3.2. Veritabanı Yazılımı	24
3.3. Ağ İletişim Yazılımı	27
3.4. Sistemin Bütünleşik Analizi	28
3.5. YARDS Yazılım Mimarisi	32
3.5.1. ADO	32
3.5.2. ODBC (Open Database Connectivity)	35
3.6. Yazma Ritmi Doğrulama Sistemi (YARDS) 'nin Yapısı	37
3.7. Sistemin Kuramsal Yapısı	37
3.7.1. Tanımlama İşlemi	38
3.7.8. Doğrulama İşlemi	39
3.7.9. Karar Verme İşlemi	39
3.7.10. Güncelleme İşlemi	40
3.8. Tasarlanan Sistemin Matematiksel Modeli ve Algoritması	40
3.9. Öğrenen Sistem Tasarımı	43
4. YAZMA RİTMİ DOĞRULAMA SİSTEMİ (YARDS)'NE AİT DENEYSEL SONUÇLAR	51
4.1. Referans Yazma Ritmi Parametreleri	51
5. SONUÇ	56
KAYNAKLAR	57
ÖZGEÇMİŞ	59
EKLER	

ŞEKİLLER DİZİNİ



Şekil	Sayfa
2.1. Biometrik Sistemlerin Genel Yapısı	9
2.2 On-line Model	10
2.3. Off-Line Model	11
2.4. Stephenson Kelimesinin Yazılış Ritmine Ait Gösterim	14
2.5. Klasik İz Tanıma Modeline Ait Gösterim	16
2.6. Minimum-Fark Vektörleri	17
2.7. Min.Fark Sınıflandırma Sistemi	18
2.8. X Değerlerine Ait Dağılım	19
2.9. Mahalanobis Metriği	20
3.1. Yazma Ritmi Giriş Ekranı	24
3.2. Sistemin Genel Yapısı	28
3.3. Kullanıcı Hakkı Kilitlenmesi	30
3.4. Klitlenmiş Hakka Ait Uyarı	30
3.5. Başarısız Giriş	31
3.6. DCOM Mimarisi	33
3.7. ADO Mimarisi	34
3.8. Yazılım Katmanları	35
3.9.ODBC Mimarisi	36
3.10. YARDS'a Ait Alt İşlemler	38
3.11. Biometrik Sisteme Ait Akış Diyagramı	45
3.12. B Dizilerine ait Grafik Gösterilim	49
3.13. G Dizilerine Ait Grafik Gösterilim	50
4.1. Veritabanından Alınan Referans Değerlere Ait	52
4.2.G Dizilerine Ait Grafik Gösterim.(Yetkisiz Kullanıcılara Ait.)	53
4.3. B Dizilerine Ait Grafik Gösterim.(Yetkisiz KullanıcılaraAit.)	53
4.4. G Dizilerine Ait Grafik Gösterim.(Yetkili Kullanıcılara Ait.)	54



TABLÖLAR DİZİNİ

Tablo	Sayfa
1.1. Biometrik Sistemlerin Hata Oranına Ait Gösterim	3
2.1. Biometrik Data Büyüklükleri (Birim Kayıt için)	12
2.2. Biometrik Veri Kıyaslaması (El Geo.Sist.göre normlanmış)	13
3.1. Veritabanının Alan Bilgileri ve Özellikleri	25
3.2. B Dizilerine Ait Öğrenme Parametreleri	47
3.3. G Dizilerine Ait Öğrenme Parametreleri	48
4.1. Aykut Kullanıcısına Ait Referans Değerler	51
4.2. Yetkisiz Kullanıcılara Ait Deneme Değerleri	52

1.GİRİŞ

Son yıllarda bilgisayar teknolojisindeki gelişmeler insanların kendi aralarında haberleşmelerini sağlayan özel yada genel ağların doğmasına sebep olmuştur. Artık her yerde bir kurumun kullandığı dağıtık bir yazılım yada en azından bir çok insanın eğlence veya alışveriş için kullandığı internet gibi global bir ağ mevcuttur.

Dağıtık sistemlerin bu kadar çok kullanılması , bu tür sistemlerin çok iyi güvenlik sistemleriyle donatılması ihtiyacını ortaya çıkartmıştır. Özellikle bir kişisel bilgisayardan bir Sunumcu'ya bağlanılan klasik İstemci/Sunumcu sistemlerde, çoğu programda güvenlik olarak sadece bir “Kullanıcı Adı” ve “Parola” bulunmaktadır. Sisteme giriş yetkisi olmayan bir kullanıcının bu iki parametreyi bilmesi demek sisteme rahatlıkla girip istediği işlemleri yapabilmesi anlamına gelmektedir. Günümüzde bu tür girişler pek çok ağa kolaylıkla ve sıkça yapılmaktadır. Örneğin genel bir ağ olarak internete bir ISS'in ağı kullanılarak “Kullanıcı Adı” ve “Parolası” bilinen birisinin hesabıyla rahatlıkla girilebilmektedir. Bu günümüzde çok sık yapılan bir ihlaldir. Bu tür ihlalleri önlemek bilgisayar ağ güvenlik sisteminin ana görevidir. Günümüzde bilgilerin sağlıklı ve güvenli ortamlarda tutulup işlenmesi çok önemli bir yer tutmaktadır. Aslında bilgisayar tabanlı sistemlerin tasarımında da öncelik verilen konu tasarım kadar önemli olan güvenlik konusudur. Herşeye rağmen yukarıda bahsedildiği üzere sadece güvenlik olarak kullanıcı adı ve parola kullanan bir sistemin güvenliğinden ve güvenilirliğinden bahsetmek oldukça gereksizdir. Kritik iş uygulamalarının çalıştığı sistemlerde yani verilecek yanlış bir komutun gerçek hayatta çok pahalıya mal olacağı düşünülen alanlarda fazladan güvenliği sağlamak için sistemi kullanan insanların biyolojik özelliklerini tanıyarak insanlara kullanım hakkı veren akıllı sistemler kullanılmaktadır. Örneğin finans yada askeri amaçla kullanılan sistemlerde bu tür biyometrik diye adlandırılacak biyolojik tabanlı sistemlerin kullanılması oldukça fazla yer tutmaktadır.

Bilgisayar ağlarına login olabilmek için gerekli olan kullanıcı adı ve şifrenin başkaları tarafından bilinmesi durumunda güvenlikten bahsetmek mümkün olamayacaktır. Bu nedenle mevcut ağ güvenlik yöntemlerine, Yüz, Parmak izi, Ses, Iris/Retina, Avuç içi şekli, İmza Tanıma yöntemleri eklenmektedir. Ancak bu

yöntemler ilave donanım gerektirdikleri için pahalı olmaktadır.[2] Anlaşıldığı üzere bu tür bir güvenlik sisteminin tek dezavantajları ihtiyaçları olan ekstra donanım ve yazılım maliyetlerinin büyüklüğüdür.

Bu tür sistemlerin güvenliğini sağlamada yazma ritmi yöntemini kullanmak bilgisayar güvenliği açısından doğal bir seçenektir. Bu seçeneğin temeli nöro-fizyolojik (neuro-physiological) faktörlere dayanan yazma ritmi'nin kişiye özel olması kabulüne dayanarak, kullanıcıları yazma ritim'lerine göre ayırt etmeyi öngörür[1]. Bu çalışmada, ağ güvenliği için maliyeti düşük olan bir sistem tasarımı yapılarak test edilmiştir.

1.1. Bilgisayar Sistem Güvenliği Sağlamada Biometrik Yöntemler

Günümüzde biometrik sistemlerin kullanılması oldukça yeni bir olgudur. Bilim kurgu olmaktan çıkıp hayatın her aşamasında sıkça kullanılması aslında oldukça kısa bir süre zarfında gerçekleşmiştir. Sadece filmlerde görmeye alıştığımız bir çok uygulama hayatımızın içerisindeki yerini çoktan almıştır[3]. Yapılan araştırmalar biometrik sistemlerin güvenilirliğinin yadsınamaz derecede yüksek olduğunu göstermektedir. Tablo 1.1'de bu konuda yapılmış olan bir araştırmadan alınmış değerler gösterilmiştir[4]. Biometrik sistemler iki ana gruba ayrılırlar bunlardan birincisi fiziksel özelliklere göre çalışan Fiziksel Biometrik Yöntemler diğeri ise insan tepkisi yada alışkanlıkları ile ilgili olan Tepkisel/Davranışsal Biometrik Yöntemlerdir.[5]. Fiziksel Biometrik Yöntemlere örnek olarak Retina/İris Tanıma, Parmakizi Tanıma gibi kullanıcının fiziksel özellikleriyle doğrudan bağlantılı biometrik sistemler gösterilebilirler. Tepkisel/Davranışsal Biometrik Sistemlere ise Yazma ritmi Tanıma Sistemleri, İmza Tanıma gibi biometrik özellikler örnek olarak gösterilebilirler.

Tablo 1.1. Biometrik Sistemlerin Hata Oranına Ait Gösterim

Biometrik Yöntem	Hata Oranı
Retina Tarama	1:10.000.000
İris Tarama	1:131.000
Parmak İzi Tarama	1:500
El Geometrisi Tarama	1:500
İmza(Yazı Yazma) Tarama	1:50
Ses Tarama	1:50
Yüz Tarama	Veri Yok
Vascular Patterns	Veri Yok

Biometrik sistemlerin güvenilirliğinin yüksekliğine ve sahip olma maliyetlerinin her geçen gün yeni araştırmalardan elde edilen sonuçlara paralel olarak düşmesine rağmen bu tür sistemlerin insanların özel hayatlarına bir müdahale şekli olduğu yönünde yoğun bir tereddütte bulunmaktadır. Biometrik sistemlerin sosyal hayatın güvenliği konusunda sunmuş olduğu büyük faydalara rağmen mahremiyet konusundaki çekincelerden dolayı bir çok kullanım sahasında kullanılmadığı da bir gerçektir.[5] Bu tür düşüncelerin yoğunluğuna rağmen biometrik sistemlerin toplum güvenliğinin artırılması konusunda bir çok probleme çözüm olabileceği de açıktır. Örneğin günümüzün yüksek teknolojiye sahip hapisaneleri buna en iyi örnek olarak gösterilebilir. Biometrik sistemlerin bu tür yerlerde kullanılması oldukça faydalı sonuçlar verebilecektir. Çalışma'da incelenen yazma ritmi konusu daha çok sistem güvenliği konusunda irdelenebilecek bir konu olmasından dolayı bu tür sistemlerin bilgisayar ve ağ güvenlik sistemlerindeki güvenliğe olan etkisi konuyla daha ilgilidir. Tablo 1.1'de verilmiş olan yöntemler bir bilgisayar sisteminin güvenliğini sağlamada da gayet rahat kullanılabilirler. Bu tür yöntemler halihazırda bir çok sistemde kullanılmaktadır. Daha önce bahsedildiği gibi bu tür yöntemlerin tek dez avantajı sistem bazında meydana çıkacak olan yüksek maliyetlerdir. Bu maliyetler 100 kullanıcıya sahip bir ağ için küçümsenmeyecek miktarlarda ek maliyet getirmektedir. Bu araştırmada ele alınan yazma ritmi

yöntemiyle geliştirilen algoritma fazladan bir donanım ihtiyacı gerektirmeden güvenli bir sistem tasarımı yapılmasına olanak vermektedir.

Daha ileriki kısımlarda detaylı bir şekilde anlatılacağı üzere bu tür sistemlerin yüksek sayılabilecek biometrik tanıma avantajlarına rağmen küçümsenmeyecek dezavantajları da mevcuttur. Biometrik sistemlerde kullanılan çoğu yöntem için avantajlı ve gerçek hayatta kullanılabilirliği fizible denilemez. Bu mali olumsuzluğa rağmen yapılan araştırmalar bu tür sistemlerin sahip olma maliyetlerinde ciddi düşüşler olmasına neden olmaktadır. Belki de yakın bir gelecekte bu tür sistemlerin gerçekleşmesi sistem bazında bugünkü gibi yüksek maliyetlerde olmayacaktır. Bunun için özellikle sensör teknolojisindeki ilerlemeler biometrik sistemlerin gelişimini etkileyecektir.



2. BİOMETRİK GÜVENLİK SİSTEMLERİ

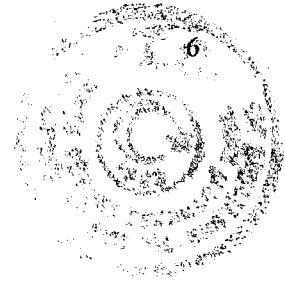
Tablo 1.1 'de ifade edildiği üzere biometrik yöntemleri kullanmak için oldukça fazla seçenek bulunmaktadır. Tablo 1.1 'de verilmiş olan biometrik yöntemler bugüne kadar kararlılığı test ve kabul edilmiş olan yöntemlerdir. Bunlar için daha doğru bir ifade kullanılması gerekirse bu yöntemler bilinen biometrik yöntemlerdir. Bu yöntemlerin ötesinde halihazırda araştırma aşamasında olan bir çok biometrik yöntemde bulunmaktadır. Bunlar arasında en dikkat çekici olanlarından birisi insanların kulak yapılarını algılayıp tanıyan sistemlerdir. Bu da göstermektedir ki insanların kendilerine has olan bütün uzuvları biometrik tanımlayıcı olarak kullanılabilir. Bu da biometrik sistemlerin ana amacını oluşturmaktadır.

Aşağıda biometrik sistemlerin kullanımı ile ilgili daha detaylı açıklamalara yer verilmiştir.[6]

2.1. Retina Tanıma Sistemleri

Retina tanıma sistemleri, isminden de anlaşılacağı üzere insanların göz yapılarını algılayarak ayırt etme prensibine dayanır. Sisteme girmek isteyen bir kullanıcı cam veya diğer bir saydam malzemeden yapılmış olan ekrana gelerek gözünü ekrana yaklaştırır. Gözünü camın arkasında yer alan bir hedef üzerinde yoğunlaştırır. Retina tarama sensörü kullanıcının gözüne bir ışın demeti gönderir ve kullanıcının retinasına ait olan özellikleri algılar. Bu bir nevi kullanıcının retina haritasını çıkartmaktır. Burada kullanılan ışın infared(Kızıl ötesi) dalga boyunda olan bir ışındır. Retina okuyucu tarafından elde edilen görüntü sistem tarafından kullanılarak kullanıcı hakkında karar verilir. Tablo 1.1'de görüldüğü üzere retina tanıma sistemleri hassasiyeti en yüksek biometrik tanıma sistemidir. Tek dezavantajı ise hassasiyetinin yüksekliğiyle doğru orantılı olarak maliyetinin yüksekliğidir.

İris taramada da benzer yöntem kullanılır fakat İris tarama yönteminin Retinal taramaya göre hassasiyeti daha düşüktür. Bunun sebebi insanlarda retina tabakasının iris tabakasına göre daha ayırt edici bir özellik olmasıyla açıklanabilir.



2.2. Parmak İzi Tanıma Sistemleri

İnsanları ayırt etmede kullanılan ve en çok bilinen yöntem parmak izidir. İnsanlar çok uzun bir süreden beri parmak izlerinin ayırt edici özelliğinin farkındadır. Bu bağlamda parmak izi tanıma sistemlerinin tarihi bilgisayarın tarihinden çok öncelere kadar uzanır. İlk parmak izi yöntemini Çin’de yazarların yazmış oldukları metinleri işaretlemek için kullandıkları söylenir. Fakat teknik manada ilk parmak izi kullanımını İngiltere’nin Hindistan’da görevli polis kuvvetlerince tutukluları ayırt etmek için 1800’lü yılların başlarında kullanıldığı bilinmektedir. Bu sistem Henry Sistemi olarak adlandırılmıştır. Yöntem elde edilen parmakizinin küçük parçalara bölünerek bu parçaların karşılaştırılmasını öngörür. Bu sebeple parmak izlerinin ayırt ediciliğini bilgisayar sistemlerinin güvenliğini sağlamak için kullanmak parlak bir fikirdir. Nihayet, günümüzde en çok kullanılan güvenlik sistemlerinden birisi parmak izi tabanlı güvenlik sistemleridir. Bu sistem için kullanılması gereken Parmak İzi Okuyucu, Retina /İris okuyucuya benzer bir çalışma prensibine sahiptir. Kullanıcı parmağını cam bir tabaka üzerine bastırır. Camın arka tarafında yer alan bir CCD kamera görüntüyü yakalayarak sisteme iletir. Sistem bu bilgiyi kullanarak kullanıcıyı tanımaya çalışır.

Bu tür bir sistemin Retina/İris Tanıma sistemine göre dezavantajı hassasiyetinin çok düşük olmasıdır. Donanım maliyeti ise Retina/İris Tanıma sistemlerine göre daha makul bir seviyededir.

2.3. Yüz Tanıma Sistemleri

Yüz tanıma sistemi aslında en doğal biometrik ayırt edici yöntemdir. İnsanlar birbirlerini ayırt etmede yüzlerinden faydalanırlar. Bu sebeple bu yöntem neredeyse insanlık tarihiyle yaşıt bir biometrik ayırt etme yöntemidir. Buna rağmen bu yöntemin bilimsel olarak incelenerek bir ayırt edici özellik olarak bilgisayarlarda kullanılması oldukça yeni sayılabilecek bir konudur. Örneğin bir suçluyu tanımlamada polis tarafından kullanılan yöntem. Suçluyu anlatan insana suçlu hakkında sorular sorup yüz şeklinin belirlenmesine çalışmaktır. Bu konuda polis

ressamlarının standart olarak daha önce çizdikleri bazı yüz parçaları vardır. Verilen tarife göre bu parçalar bir araya getirilerek şuçlunun yüz şekline yakın bir yüz şekli bulunmaya çalışılır. Bilgisayarların hayatın içerisindeki hızlı kullanımı yüz tanıma yöntemleri konusunda da oldukça ilerlemeler sağlanmasına neden olmuştur. Günümüzde yüz tanımlama konusunda kullanılan iki yöntem vardır. Bunlar,

- 1- Yüz Metriği Yöntemi
- 2- Eigenfaces Yöntemi(Yüz Parçaları).

Yüz Metriği Yönteminde yüz üzerinde yerleşik olan organlar arasındaki mesafeler ölçülerek bunlardan bir matematik ifade çıkarılmaya çalışılır. Örneğin gözler arasındaki mesafe, ağız ve burun arasında ki mesafe vb....

Eigenfaces Yönteminde ise polis ressamının yaptığına benzer bir yöntem uygulanır. Yüz 150 parçaya bölünür. Bu 150 parçadan 40 tanesinin belirleyiciliği diğer parçalardan daha fazladır. İlk etapta bu 40 parçadan başlayarak yüz tanımlama işlemine geçilir. Böylece yüz tanımlanmaya çalışılır. Bu yöntem Yüz Metriği Yöntemine göre yeni bir yöntemdir ve hala test aşamasındadır. Araştırmacılar tarafından geliştirilmektedir.

2.4. El Geometrisi Tanıma Sistemleri

El Geometrisi Tanıma Sistemleri, her insanın el yapısının ve şeklinin farklı olduğu kabulüne dayanır. Gerçekte belli bir yaşta sonra insanların el yapıları değişiklik göstermezler. Bu yöntem kullanılarak tasarlanan bir güvenlik sisteminde sistem kullanıcıların parmak uzunluklarından avuç içlerinde yer alan çizgilere kadar bir çok parametreyi algılayarak karar verir. Avuç içini algılamada kullanılan yöntem parmak izi tayininde kullanılan yöntemin aynısıdır. El Geometrisi sistemi kullanılarak tasarlanan bir sistemin hassasiyeti parmak izi sistemleri kadardır. Bu açıdan bakıldığında güvenilirlik ve sahip olma maliyeti açısından özdeş sistemlerdir denilebilirler.

2.5. İmza(Yazı) Tanıma Sistemleri

Her insanın gerçek hayatta kullandığı ,el yazması metinlerde insanları ayırt edici bir özellik olan imzaları bulunmaktadır.Eskiden beri insanlar kendilerini veya kendi ailelerini betimleyen imza, arma yada özel işaretler kullanmışlardır.Gerçekten dikkat edilirse imzalarımızı devamlı aynı şekilde,aynı hızda birbirleriyle oldukça benzer biçimde atarız. İmza Tanıma Sistemleri 'nin temelinde de bu mantık vardır.Eğer herkes imzalarını her zaman aynı biçimde atıyorlarsa bu imzalar elektronik ortama aktarılarak ayırt edici bir güvenlik sistemi olarak kullanılabilir.Bunun için yazıyı elektronik ortama aktaran cihazlar uzun zamandır kullanımdadır. Bu aletler uygun şekilde kalibre edilerek bu tür bir güvenlik sistemi kurmak oldukça kolaydır. Güvenilirlik seviyesi Parmakizi ve El Geometrisi Sistemlerine göre oldukça düşüktür.

2.6. Ses Tanıma Sistemleri

Ses tanıma sistemleri oldukça yaygın kullanılan maliyeti düşük ve günümüz teknolojisinde fazladan hiçbir donanım gerektirmeyen güvenlik sistemlerinden biridir. Ses Tanıma Sistemleri genel olarak iki ana gruba ayrılır.

- 1- Ses Tanıma Sistemleri
- 2- Konuşma Tanıma Sistemleri

Ses tanıma sistemlerinde kullanıcı sisteme parola olarak belirlediği bir kelimeyi bazı heceleri vurgulu olacak şekilde söyler sistem parolanın kendisiyle beraber vurguların zamanlamasına ve siddetine bakarak kullanıcıyı ayırt eder.

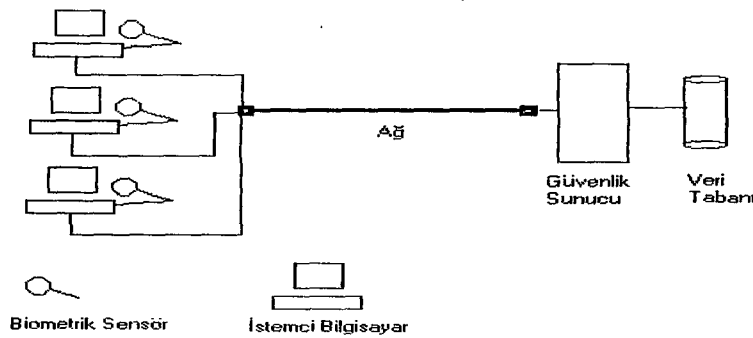
Konuşma Tanıma Sistemleri,Ses Tanıma Sistemlerinden teknik manada daha gelişmiştir.Burada vurgu tamamen ortadan kalkar. Yani kullanıcı özellikle bir vurgu yapmaz.Sistemle normal olarak konuşur.Sistem kullanıcıyı konuşma modundan anlamaya çalışır. Ses tanıma sistemleri hassasiyet olarak en düşük biometrik sistemlerden birisidir. Avantajı ise sahip olma maliyetinin düşüklüğüdür.

Burada en büyük dezavantaj sesin bilgisayar ortamına aktarılırken geçirdiği büyük değişim ve formatındaki bozukluktur. Bu sebeplerden dolayı Ses Tanıma Sistemleri vurgu içermeleri açısından Konuşma Tanıma Sistemlerine göre daha güvenilirdir. Ses Tanıma Sistemleride diğer araştırma ve geliştirme aşamasında olan sistemler gibi hala üzerinde çalışılan biometrik sistemlerden biridir. Fakat burada en önemli kriter ses işleme aşamasını daha doğru hale getirebilecek olan donanım parçalarının geliştirilmesidir. Günümüzde standart PC'lerde kullanılan mikrofonlar ve hoparlörler bilgisayarda ses işlenmesi için iyi bir alternatif sunmamaktadır. Buna rağmen yüksek hassasiyete gerek olmayan uygulamalarda standart ses Girdi/Çıktı aygıtlarını kullanarak bir güvenlik sistemi tasarlamak maliyetler gözönüne alındığında iyi bir seçenek olarak görülmektedir.

Ses işleme yöntemiyle kurulacak bir sistemin en büyük dezavantajı kullanıcıları tanımlamak için veritabanında tutulan kayıtların büyük yerler işgal etmesidir. Tasarlanan güvenlik sistemi ağ üzerine çok fazla yük getirmemelidir.

2.7. Biometrik Sistemlerin Kuramsal Tasarım Yöntemleri

Şimdiye kadar ki bölümlerde biometrik yöntemlerin genel çalışma prensiplerinden öte bu tür sistemlerin çalışabilmeleri için nasıl bir fazladan donanıma ihtiyaç duyduklarını anlatıldı. Örneğin bir parmakizi tanıma sisteminde her istemci için bir CCD kamera içeren sensöre ihtiyaç vardı. Bahsedilen sensörlerin içeriklerinin farklı olmasına rağmen yaptıkları iş aynıdır. Bütün biometrik sistemlerin kurulumu için istemci başına bir sensör gerekmektedir. Daha sonra buradan okutulan bilgiler tasarlanan sistemin mimarisine bağlı olarak işletilirler. Aşağıda bir biometrik sisteme ait kuramsal çizim yer almaktadır.



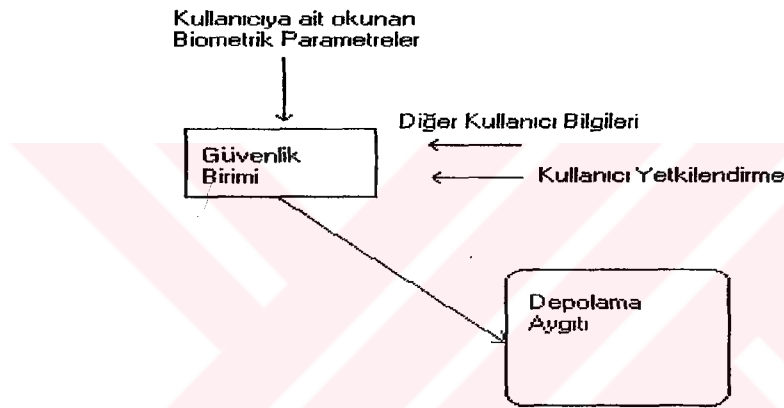
Sekil 2.1 Biometrik Sistemlerin Genel Yapısı

Biometrik sistemlerin tasarımı için uygulanan iki model vardır.

- 1- On-line Model
- 2- Off-line Model

2.7.1. On-line Model

On-line model yapısında, kullanıcı okunması gereken biometrik parametresini sisteme okutturur. Sistem almış olduğu biometrik parametreleri eğer gerekliyse kriptoloyarak ağ üzerinden güvenlik parametrelerinin tutulduğu sunucuya gönderir. Sunucu bu parametreleri veritabanı içerisindeki bilgilerle karşılaştırır. Eğer kullanıcı

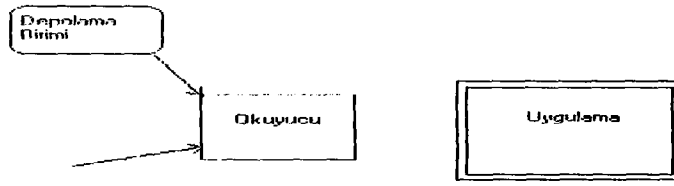


Şekil 2.2 On-line Model

sisteme kayıtlı biriyse sisteme giriş izni gönderir. Kullanıcı tanımlanamadıysa sistem giriş izni vermez. Günümüzde on-line sistemler kullanılmaktadır. On-line bir sistemde önemli olan nokta biometrik bilgileri okuyan sensörlerin sunucuya çok güvenli bir şekilde bağlanmaları gerekmektedir.[7] Bunun için ya sensörler ve sunucu arasında güvenli bir yol tayin edilmelidir yada bilgiler yukarıda bahsedildiği üzere çok iyi şifrelenmiş olarak gönderilmelidir. Bilgilerin herhangi bir nedenden dolayı yetkisiz insanların eline geçmesi sistemin büyük bir zaafa uğramasına sebep olabilir. Öyleyse parametreler öyle bir şekilde şifrelenmeli ki bu parametreler bir şekilde elde edilse bile kullanılamamalılar.

2.7.2. Off-line Model

Off-line sistemlerde ,On-line sistemlerde olduğu gibi gerçek zaman bir doğrulama işlemi yapılmaz. Bunun için gerekli olan bir manyetik kart gibi aparatlar kullanmaktır. Kullanıcıya ait olan bilgiler bu kart üzerinde tutulurlar. Kullanıcı sisteme girmek istediği zaman bu kartı kullanır. Bu tür sistemlerde güvenlik tamamen kullanıcının insiyatifindedir. Kendisine verilmiş olan depolama aygıtını çok iyi korumak zorundadır. Tersi bir durumda sistemin güvenilirliğinden söz etmek gerçekçi olmayacaktır.



Şekil.2.3. Off-Line Model

Bu sebeplerden dolayı off-line bir sistem tasarlanırken güvenlik açısından üzerinde çok daha fazla düşünülmesi gereklidir.

2.8. Biometrik Sistemlerin Veri Kayıt Analizleri

Biometrik sistemlerin seçiminde kullanılan önemli parametrelerden biriside kayıtların veri boyutlarının büyüklüğüdür. Hangi biometrik sistem kullanılırsa kullanılsın veriler mutlaka bir veritabanı üzerinde tutulup yönetilmek zorundadır. Böyle bir durumda ortaya çıkan soru sisteme kayıt edilecek kullanıcılara ait biometrik özelliklerin kayıt uzunlukları ne kadar olacaktır? Tablo 2.1 bir kayıt için gerekli olan miktarları byte olarak göstermektedir.

Tablo 2.1. Biometrik Data Büyüklükleri (Birim Kayıt için)

Biometrik Yöntem	Data Büyüklükleri (bytes)
Retina Tarama	35
İris Tarama	256
Parmakizi Tarama	512 – 1000
El Geometrisi Tarama	9

Tablo 2.1’den anlaşıldığı gibi biometrik yöntemlerin sistem üzerinde kaplamış olduğu alan miktarı birbirlerine nazaran oldukça farklıdır. Örneğin retinal tarama sistemleri iris tarama sistemlerine nazaran yaklaşık 10 kat daha güvenli oldukları halde kullanıcı tanımlamak için ihtiyaç duydukları kayıt alanı;

Ampirik bir hesaplamayla $256 / 35 = 7.31$

Yaklaşık 7 kat daha az yere ihtiyaç duymaktadır. Aynı şekilde bir kullanıcının parmak izini tutmak için gerekli olan alan 512-1000 byte arasında değişmektedir. Parmak izi tanıma sistemlerinin iris ve retina tanıma sistemleri gibi kendisinden kıyaslanamayacak kadar güvenli sistemlerin ihtiyacı olan alandan çok daha fazla bir disk alanına ihtiyaç duyması çok büyük bir dezavantajdır. Veri uzunluğu açısından bakıldığında en uygun biometrik güvenlik sistemi El Geometrisi tanıma sistemleridir. Tablo 2.1’den açıkça görüldüğü üzere ihtiyacı olan alan sadece 9 Byte ‘dır.

Tablo2.1’de verilen değerleri en küçük veri depolama ihtiyacı duyan El Geometrisi Tanıma Sistemine göre normlarsak şöyle bir sonuç elde ederiz.

Tablo 2.2. Biometrik Veri Kıyaslaması (El Geo.Sist.göre normlanmış)

Biometrik Yöntem	Norm
Retina Tarama	4
Iris Tarama	28
Parmakizi Tarama	57 – 111
El Geometrisi Tarama	1

Tablo 2.2 ‘den açıkça görüldüğü üzere bir Parmak İzi Tarama sisteminde tutulan 1 kayıt El Geometrisi Tarama yöntemine göre tasarlanmış bir sisteme göre yaklaşık 100 kat daha fazla yer kaplamaktadır. Bu da açıkça göstermektedir ki bu tür sistemlerin kurulmasında biometrik sistemin güvenlik oranı,maliyeti gibi parametrelerin yanında veri büyüklüklerindeki tasarımcılar tarafından göz önüne alınarak karar vermesi gerekmektedir.Çünkü veri uzunluğunun fazla olması ekstra donanım ihtiyacı yada ağ trafiğini gereksiz yükleme gibi bir çok dezavantajı yanında getirecektir.

2.9. Yazma Ritmi Sistemleri

Bölüm 1.1’de biometrik sistemlerin en büyük dezavantajlarından birinin bütün terminallerde kullanılmak üzere ekstra donanım ihtiyacı hissetmeleriydi.Yazma Ritmi sistemleri diğer biometrik sistemlerden ayıran en temel özellik kendisine has özel bir donanıma ihtiyaç duymamasıdır.Bir Yazma Ritmi sistemi tasarlamak için ihtiyaç duyulan tek girdi aygıtı sadece klavyedir.

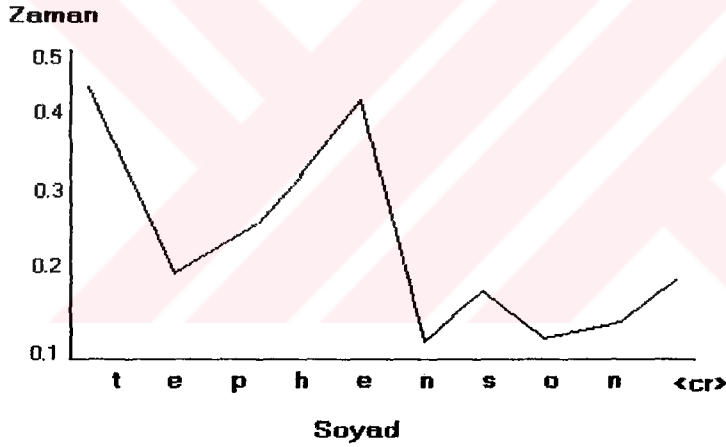
Böyle bir avantaja sahip bir biometrik sistemin gerçek hayatta kullanılmaması için hiç bir sebep yoktur. Özellikle üniversiteler gibi pek çok yerde bu tür bir sistem hem bilimsel hemde teknik manada rahatlıkla kullanılabilir.Çünkü insanların el yazıları gibi klavye yazış şekilleri arasında büyük bir benzerlik vardır.[1]

Gerçektende bir insan klavye üzerinde bir şeyler yazmaya başladığında bu klavye ve doğal olarak bilgisayar üzerinde bir iz bırakır. Yani kullanıcının klavyede karakterler üzerinde gezinmesi aslında onun gayet düzenli olarak yapmış olduğu bir yazma ritmi hareketine tekabül eder.

Diyelim ki klavyeyi kullanarak bilgisayarda bir editöre bir kelime yazmak istiyorsunuz. Teorik olarak kelime içerisinde yer alan harfleri yazarken geçen süre birbirine eşittir. Biz bu konuda özellikle bir ölçüm yapmadık fakat bu konuda Rick Joyce ve Gophal Gupta'nın yapmış oldukları bir deneyde bu süreler tespit edilmiştir. Bir kullanıcıdan soyadı yazılması istenmiş ve yazılma aşamasında geçen süreler tespit edilmiştir;

Stephenson

Kelimesi için geçen süreler ölçülmüş ve bu sürelerle ait bir grafik elde edilmiştir.



Şekil 2.4. Stephenson Kelimesinin Yazılış Ritmine Ait Gösterim

Şekil 2.4 kullanıcının Yazma Ritmi sistemine ait bulgular vermektedir. Kullanıcı, t ve e karakterleri arasında oldukça zaman kaybederken n ve s yada s ve o karakterleri arasında geçişi oldukça hızlı yapabilmektedir.

Rick Joyce ve Gopha Gupta çalışmalarında sadece klavyeden girilen iki karakter arasında geçen zamanı ölçerek kullanıcıların yazmaritimlerini tespit etmeye çalışmışlardır. Aslında yazma ritmi'yle ilgili olarak bir çok çalışmada ölçülen ve değerlendirilmeye çalışılan değer budur.

Daha önce yapılan çalışmalardan farklı olarak bu çalışmada yapılan iki karakter arasında geçen zamanın ötesinde, bir karaktere basarken geçirilen sürenin de tespit edilmiş olmasıdır. Tespit edilen basış süresi algorithmada kullanılmıştır. Yazma ritmi sistemlerinin en önemli dezavantajlarından birisi karar vermek için gerekli olan verilerin ancak tek bir parametreden yani iki tuş arasındaki gecikme süresinden elde edilmesinden kaynaklanmaktadır. Sadece tuş aralıklarını değil tuşlara basılırken geçen sürenin de tespiti yazma ritmi çalışmalarında güvenilirlik oranını arttıracaktır.

2.10. Yazma Ritmi Sistemlerinde Sınıflama Algoritmaları

Yazma ritmi sistemlerinde kullanılan tanıma algoritmalarında bir çok yöntem kullanılmaktadır. Bu yöntemler aşağıda verilmiş olan ana yaklaşımlar yardımıyla türetilirler.

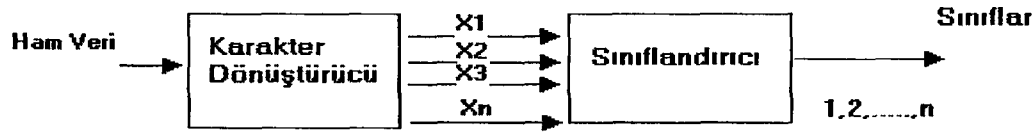
- İstatistik yada bulanık iz tanıma
- Yapısal iz tanıma
- Bilgi tabanlı iz tanıma

Bu tür sistemlerde genellikle yapılan elde edilmiş olan ham veriyi bir sınıflama algoritmasından geçirerek elde edilen verileri sınıflandırılmada kullanılacak referans işaretlere eşitleme mantığına göre çalışır.

Sınıflama yapmada kullanılan yöntemler de kendi içerisinde bir kaç alt gruba ayrılır. Sınıflandırma mantığında tanımlanması gereken işaret bilgisayar ortamına aktarılacak hertürlü gürültüden mümkün mertebe temizlenir. Örneğin bir karakter tanıma sistemi kuracak olursak bir yazı parçasından alınmış olan harfler öncelikle bilgisayar ortamına mümkün mertebe iyileştirilerek yada mevcut durumu bozulmadan aktarılır. Bundan sonraki aşamada tanınması gereken bütün karakterler tek tek alınarak algorithmadan geçirilir. Algorithmadan geçirilen karakterlerden elde edilen parametreler referans olarak veritabanında tutulan değerlerle karşılaştırılır. Bu parametrelerin mümkün mertebe hangi karaktere benzediği konusunda bir karar

verilerek iki deęer birbirine eřitlenir. Scanner’larda kullanılan OCR programları bu tür alıřan sistemlere bir rnektir.

Klasik sınıflandırma sistemlerinde tanımlanmak istenen iřaretler ham halden daha anlamlı ifadelere dnřtrlr. Bunu yapmak iin giriř bilgileri Karakter dnřtrc olarak adlandırılabilcek bir algoritmaya sokulur buradan elde edilen deęerler Sınıflandırma algoritmasına girer , bylece elde edilen deęerin karřılařtırılmasıyla veri hakkında bilgi toplanır.



Klasik İz Tanıma Modeli

řekil 2.5- Klasik İz Tanıma Modeline Ait Gsterim

Burada alınan veriler sayısal X deęerlerine dnřtrlr. Bu deęerler X vektrn oluřtururlar. Sınıflandırıcı olarak adlandırılan program ise bu deęerleri alarak verileri daha nce tanımlanmıř bir sınıfa eřitler.

Yazma ritmi sistemlerinin sınıflandırılmasında kullanılan vektrel yntemlerden bahsedecek olursak , bilinen 2 nemli yntem vardır.

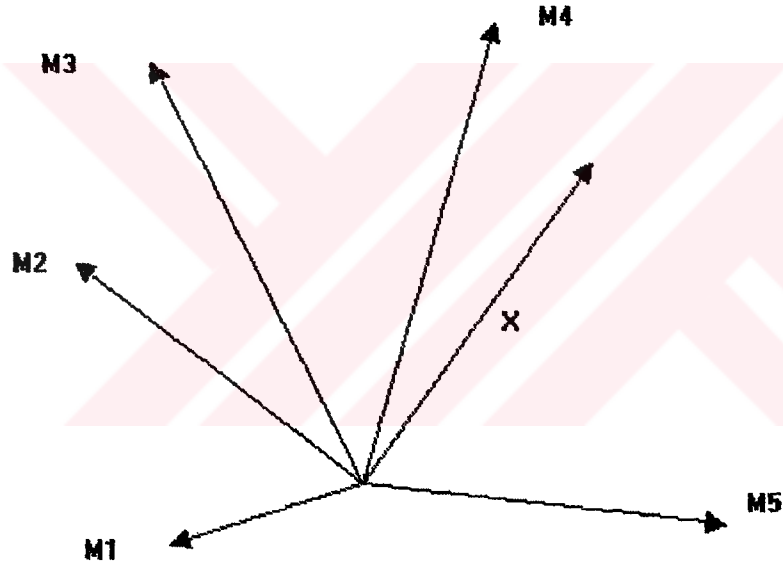
- 1- Minimum-Fark Sınıflandırması
- 2- Mahalanobis Sınıflandırması

2.11. Minimum-Fark Sınıflandırması

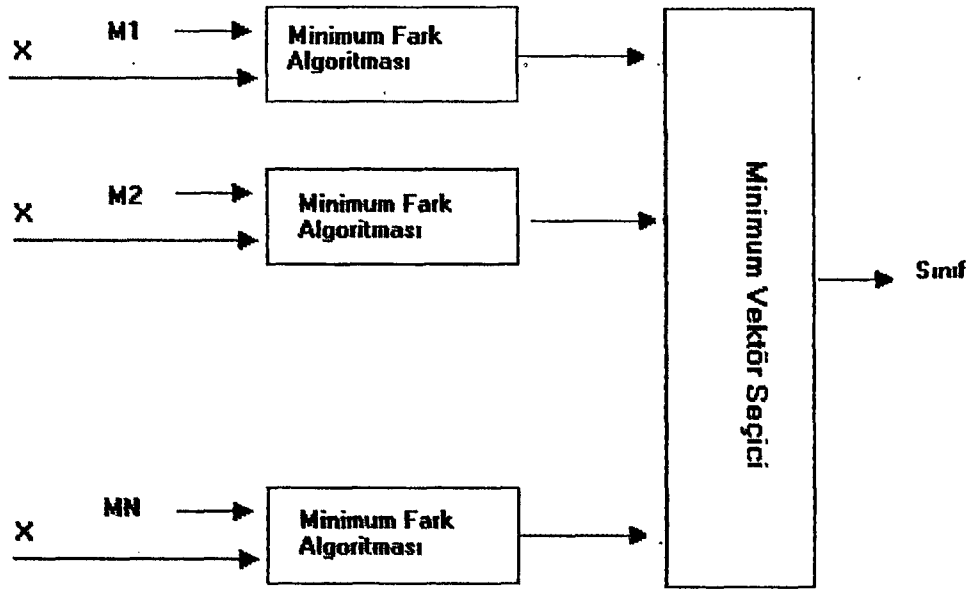
Minimum-Fark Sınıflandırılmasına ait matematiksel ifade oldukça basittir. Burada master olarak kullanılan X vektörleri ve gürültüden arındırılmış M vektörleri vardır. Bu iki vektör arasındaki fark vektörü bulunur.

$$\| X - M_k \|$$

2.1



Sekil 2.6. Minimum-Fark Vektörleri



Şekil 2.7. Min.Fark Sınıflandırma Sistemi

2.12. Mahalanobis Sınıflandırması

Mahalanobis sınıflandırma yöntemi Minimum fark sınıflandırma yöntemine göre daha gelişmiş ve güvenilir bir yöntemdir. Mahalanobis sınıflandırmasında metrik ve sınıflama işlemleri daha gelişmiştir.

Örneğin N boyutlu bir X iz vektörümüz olsun. Bu X vektörünün elemanları şöyle ifade edilebilir $X, X(1), X(2), X(3), X(4), \dots, X(n)$. Bu vektör elemanlarının aritmetik ortalaması alınır.

$$M = [X(1)+X(2)+X(3)+\dots+X(n)] / n \quad 2.2$$

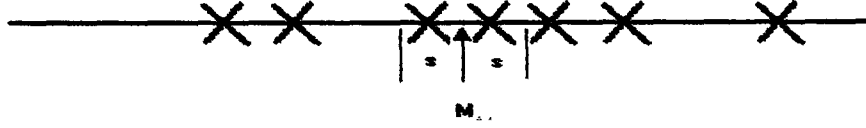
Burada m ortalama kütle diye de adlandırılabilir. Buradan izlere ait varyansı hesaplayacak olursak

$$V = [(X(1)-M)+(X(2)-M)+(X(3)-M)+\dots+(X(n)-M)] / (n-1) \quad 2.3$$

Görüldüğü üzere X ve M aynı boyutta değerlerdir. V ise aynı boyutta değildir (X 'in karesi). O halde V 'nin karekökünü alacak olursak vektör değerlerine ait standart sapmayı bulabiliriz.

$$S = \text{sqrt}(V)$$

2.4



Şekil 2.8. X Değerlerine Ait Dağılım

Bu ifadeler açıkça gösteriyor ki bir m kütlesi etrafında $+s$ ve $-s$ civarındaki salınımlarda izlerin %68'i belirtilen aralıkta kalmaktadır. Aynı şekilde aralığı $+2s$ ve $-2s$ olarak ölçeklersek Izlerin %95'i belirtilen aralıkta yer alacaktır.

2.13. Covariance Metriği

Mahalanobis sınıflandırmasında daha iyi sonuçlar elde etmek için Covariance metriği diye adlandırabileceğimiz bir yardımcı metrik kullanılır. Burada izler bir matris haline getirilir hesaplamalar bu yöntemle yapılır. Bu yöntem vektörler arasındaki farkların daha iyi ölçülmesini sağlar.

D boyutlu bir vektör olduğunu kabul edersek, bu vektöre ait Covariance matrisi aşağıdaki gibi tanımlanır.

$$\begin{bmatrix} C(1,1) & C(1,2) & \dots & C(1,d) \\ C(2,1) & C(2,2) & \dots & C(2,d) \\ \vdots & \vdots & \ddots & \vdots \\ C(n,1) & C(n,2) & \dots & C(n,d) \end{bmatrix}$$

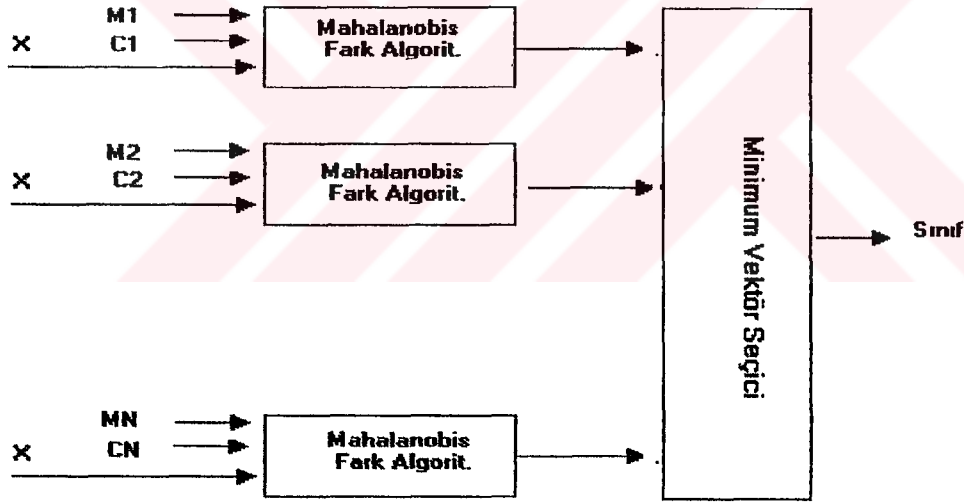
$y = A x$ türünden ifadelerde A matrisi x vektörünü y vektörü türünden vektörlere dönüştürür. Bu tür matrislere dönüşüm matrisleri denilebilir.

Vektör uzayı içerisinde bir x noktasından bir y noktası arasındaki fark hesaplanmak istenirse bunun için aşağıdaki gibi bir bağıntı kullanılması gerekmektedir.

$$r^2 = (x - m_x) C_x^{-1} (x - m_x) \quad 2.4$$

2.14. Mahalanobis Metriği

Mahalanobis metriği yöntem olarak covariance matrisleri kullanır. İki nokta arasındaki Mahalanobis farkı formül 2.4 'deki gibi ifade edilebilir.



Şekil 2.9. Mahalanobis Metriği

Burada m değerleri master (referans) değerleridir. X değerleri ise iz vektörüne ait değerlerdir. $C_1, C_2, C_3, C_4, \dots, C_n$ ise bu değerlere karşılık gelen covariance matrisleri göstermektedir. Metrik bu ifadelerin formülde yerine koyulup hesaplanmasıyla elde edilebilir.

2.15. En Yakın Komşu Kuralı

Biometrik izleri sınıflandırmada kullanılabilecek yöntemlerden birisi de en yakın komşu kuralı olarak adlandırılabilir. Burada X_1 den X_n 'e kadar örnek değerler olduğunu varsayalım. Herhangi bir zamanda elde edilmiş olan X değeri bu örnek değerler içerisinde kendisine en benzer olan değere eşitlenir. Böylece bir sınıflandırma yapılmış olur. Bu yöntem daha çok olasılığa dayanır. Bir izi tanımlamak için çevresindeki izler hesaplamaya konulur ve bir olasılık bulunmaya çalışılır.

$$\eta = \sum_{i=(k+1)/2}^k \binom{k}{i} P(w_m | x)^i [1 - P(w_m | x)]^{k-i} \quad 2.5$$

2.16. Yazma Ritmi Sistemlerinde Vektörel Analiz Yöntemi

Yazma ritmi sistemlerinin tasarımında kullanılan en etkili yöntem elde edilen parametreler yardımıyla bir vektörel analiz yapmaktır. Gerçekte bu yöntem sistemin doğruluğu açısından kullanılabilir hata oranlarına sahip bir algoritma tasarımında bir çok araştırmacı için bir yol göstericidir. Bu vektörler Vuruş(Hammering) vektörleri olarak adlandırılabilirler.[5]

Kullanılan vektörel analiz yöntemi aslında oldukça basittir. Bu çok boyutlu bir uzayda boyutlar üzerinde yer alan vektör parçalarının birbirinden çıkartılarak sonunda normu "0" değerine yakın bir vektör elde edilme çabasıdır. Eğer elde edilen bileşke vektörün normu 0'a bu tercih edilen bir durumdur.

Şekil 2.4'de grafiği verilmiş olan örneğe geri dönülecek olursa şöyle bir yaklaşımda bulunulabilir.

tephenson parolası için tam 8 tane vurma vektörümüz vardır.

- t-e arasında geçen süre 1. Vurma vektörü.
- e-p arasında geçen süre 2. Vurma vektörü.
- p-h arasında geçen süre 3. Vurma vektörü.
- h-e arasında geçen süre 4. Vurma vektörü.
- e-n arasında geçen süre 5. Vurma vektörü.
- n-s arasında geçen süre 6. Vurma vektörü.
- s-o arasında geçen süre 7. Vurma vektörü.
- o-n arasında geçen süre 8. Vurma vektörü.

Bu vektörler m_j ile gösterilecek olursa; $M = \sum m_j$ gibi bir bağıntı elde edilebilir. Benzer bir vektörde $T = \sum t_k$ olacak olursa $k=j$ durumu için vektörel bağıntı , Δ fark vektörü olmak üzere ; $\Delta = \sum (m_j - t_k) = 0$ durumu elde edilmek istenecektir.

3. YAZMA RİTMİ DOĞRULAMA SİSTEMİ (YARDS) TASARIMI VE GERÇEKLENMESİ

Bu bölümde tasarlanan sistemi oluşturan teknolojiler ve sistemin genel yapısı hakkında teknik bilgiler verilmiştir. YARDS genel olarak 4 kısma ayrılabilir.

- 1- İstemci tarafı yazılımı.
- 2- Veritabanı yazılımı
- 3- Ağ iletişim yazılımı (Middleware)

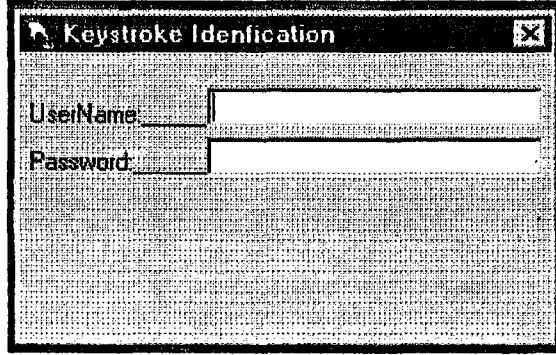
3.1. İstemci Tarafı Yazılımı

İstemci tarafı yazılımı sistem içinde çalışan bütün bilgisayarlarda olan bir yazılımdır. Tasarımı yapılan sistemde yazılım test amaçlı olarak İstemci/Sunucu bir ortamda kullanılmıştır. Fakat mevcut algoritmaya sadık kalınarak bu sistem hemen hemen bütün platformlara uygulanabilir. Örneğin bir browser üzerinde java applet olarak kullanılırsa internet üzerinde bilgi giriş çıkışlarında bir ön güvenlik giriş sayfası olarak kullanılabilir. İstemci Tarafı Yazılımı görünüm olarak klasik bir login ekranıdır. Üzerinde kullanıcı adı ve parolasının bulunduğu bir giriş ekranıdır. Kullanıcı bu ekrana daha önceki bölümlerde bahsedildiği üzere kullanıcı adı ve parolasını girdiğinde İstemci Tarafı Yazılımı girilen bilgileri dinamik olarak bilgisayarın hafızasında tutar. İstemci tarafında herhangi bir yere statik olarak bu bilgiler kesinlikle yazılmaz. Bu tasarlanan sistemin güvenlik önlemlerinden birisidir. Sadece İstemci Tarafı Yazılımı tarafından kullanılan bu bilgiler herhangi bir şekilde Ağ ortamında ulaşıp elde edilemezler.İstemci tarafı Yazılımı çalışmasını bitirdiğinde bu bilgiler hafızadan doğrudan silindikleri için herhangi bir iz bırakmazlar böylece kullanıcılara ait olan bu özel bilgiler bir sonraki girişe kadar sadece veritabanı üzerinde tutulurlar.

Bu yazma ritmi yönteminin tamamen bir sistem üzerinde çalıştığı durumlar için önerilen bir yöntemdir. Aynı şekilde İstemci Tarafı Yazılımı tek bir bilgisayar üzerinde de çalıştırılabilir. Bu durumda bilgiler aynı bilgisayar içerisinde ya bir local

veritabanında yada bir dosya yapısı içersinde tutulabilirler.Burada tek dezavantaj kullanıcı bilgilerinin tutulduğu yerin aynı bilgisayarda yer almasından doğabilecek dezavantajlardır.Bu da günümüz güvenlik şartları gözönüne alındığında kabul edilebilecek bir risktir.

Şekil 3.1’de programın giriş arayüzüne ait çizim görülmektedir.



Şekil 3.1. Yazma Ritmi Giriş Ekranı

3.2.Veritabanı Yazılımı

Bölüm 4’de yazılım ihtiyaç duyduğu referans parametreleri veritabanından aldığı ve güncellemeleri de aynı şekilde veritabanı üzerinde yaptığından bahsedilecektir. Deneme aşamasında program veritabanı yazılımı olarak Oracle 8 veritabanını kullanmıştır. Aynı şekilde program sistem yönetim işlemleri için sunucu üzerinde çalışan bir server programını kullanmaktadır. Bu programda yer alan menülerden yönetim ile ilgili işlemler yapılabilmektedir.Yönetim işlemleri şöyle sıralanabilir.

- 1- Kullanıcılara hak verme hak alma işlemleri.
- 2- Kullanıcıların kayıtlarını silme yada kitleme işlemleri.
- 3- Sistemin hassasiyetiyle ilgili işlemler.
- 4- Sistemin kullanıcıların yanlış giriş opsiyonlarının kayması işlemleri.
- 5- Kullanıcıları takip etme işlemleri.

Server üzerinde bulunacak olan bu yazılımla sistemin bütün güvenlik ve yönetim opsiyonları tek bir yerde toplanmış olur. Böylece sistemin güvenliği açısından bu mimari daha uygun olmaktadır.

Kullanıcıların parametrelerini ve kişisel bilgilerini tutmakta olan veritabanı özellikleri Tablo 3.1’de belirtilmiştir.

Tablo 3.1. Veritabanının Alan Bilgileri ve Özellikleri

ALAN ADI	VERİ TİPİ	VERİ UZUNLUĞU
UserName	Text	25
Password	Text	8
Pfunction	Double	
LockAccount	Logical	True/False
PressTime	Double	
DelayTime	Double	
S1	Double	
S2	Double	
S3	Double	
S4	Double	
S5	Double	
S6	Double	
S7	Double	
S8	Double	
G2	Double	
G3	Double	
G4	Double	
G5	Double	
G6	Double	
G7	Double	
G8	Double	

Tablo 3.1’de Yazma ritmi tabanlı güvenlik sisteminin kullanıcı parametrelerini tuttuğu “security” adlı tabloya ait alan isimleri ve açıklamaları yer almaktadır.

Kullanıcı Adı : Kullanıcı ismi olarak 25 haneli bir alan ayrılmıştır. Bu uzun isimlerinde veritabanında kolayca tanımlanabilmesi için gereklidir.

Parola : Kullanıcının parolasının girildiği “Password” alanı 8 karakter genişliğindedir. Bu program içerisinde sabitlendiği için parola 8 haneden büyük yada küçük olamaz.

Pfonksiyonu : Pfunction alanı program tarafından kullanıcının yazma ritminden türetilen bir parametredir. Her kullanıcıya göre değişik değerler alan bir parametredir. Kullanıcının tanımlanması aşamasında geçici olarak 1 değerini alır. Kullanıcı sisteme parametrelerini öğretip sisteme kendisini tanıttıktan sonra 0-1 arasında bir değer alır. Değeri hiç bir zaman $0 < \text{Pfunction} < 1$ değerinin dışına çıkamaz. Normal kullanıcılar veya abartısız bir yazma ritmi kodlama için Pfunction değeri her zaman 1 değerine yakınsar.

Kullanıcı Kilitle : LockAccount kullanıcıların haklarıyla ilgili bir alandır. Daha önceden herhangi bir sebepten dolayı kullanıcı hakları kısıtlandıysa bu alanın değeri true ‘dan false de ğerine döndürölür. Sistem yöneticisi tarafından bu değeri tekrar true değerine getirilinceye kadar kullanıcı sisteme giremez.

Basma Zamanı : G.Y.T.E’de tasarlanan sistem diğeri tasarlanmış sistemlerden farklı olarak tuşlara basış sürelerinin de ölçer ve bu ölçüm değerlerini karar verme ve doğrulama gibi bir çok yerde kullanır. PressTime alanı bir kullanıcının parolasını kodlarken karakterlere ne kadar süre ile bastığının bilgisini saklar. Örneğin kullanıcı parolası olan “19511951” şifresini yazarken 10 msn. harcamış olabilir. PressTime alanı bu zaman değerini tutar.

PressTime değerini şöyle formüle edebiliriz ;

$$\text{PressTime} = \sum_{i=1}^8 S_i$$

Gecikme Zamanı : Tabloda yer alan diğeri bir parametre ise DelayTime dır. DelayTime’nın işlevide PressTime’la aynıdır. Bu parametre bir standart yazma ritmi sistemi tasarlanırken ölçölmesi gereken tuş aralarındaki gecikme zamanlarıdır.

DelayTime değerini şöyle ifade edebiliriz.

$$\text{DelayTime} = \sum_{i=2}^8 G_i$$

S Parametreleri daha önce ifade edildiği gibi S dizilerinin değerlerini saklarlar. Tabloda görünen S1,S2,.....S8 alanları bu dizilerin aynı isimli değerlerini saklarlar.Bu değerler kullanıcı sisteme giriş yapmak istediği zaman yazılıma gönderilirler.

G Parametreleri’de benzer şekilde G dizilerinin değerlerini saklarlar. Tabloda görünen G1,G2,.....G8 alanları bu dizilerin aynı isimli değerlerini saklarlar.Bu değerler kullanıcı sisteme giriş yapmak istediği zaman yazılıma gönderilirler.

3.3. Ağ İletişim Yazılımı

Tasarlanan sistem bir Ağ yazılımıdır.Genel manasıyla bu tür sistemler İstemci tarafı,Ağ ara yazılımı ve server tarafı yazılımları olmak üzere 3 ana gruba ayrılırlar. Bizim tasarımı yaptığımız sistemde Ağ iletişim işlemini yapmada Microsoft ODBC yazılımını kullandık. Ağ ortamlarında veri iletişimi sağlamanın en kolay yollarından biri bu tür middleware yazılımları kullanmaktır. Bu tür yazılımları kullanmak istemeyenler için yapması gereken Ağ ortamında iletişim sağlayabilecek güvenli bir yazılım yazmaktır. Bunun için genelde socketler kullanılır.

Gerçekten de socketler kullanılarak TCP/IP gibi bir protokol üzerinde yazılım yapıp kullanmakta iyi sonuçlar doğurmaktadır.Böyle bir yazılım middleware yazılımlara göre daha hızlı çalışırlar fakat güvenilirlikleri konusunda aynı şeyi söylemek söz konusu değildir. Bu nedenle fazla veri akışı olmayan işlemlerde middleware türü hazır programlar kullanmak daha iyi olacaktır.

Windows ortamında ODBC tanımlamak için kontrol panelden ODBC bağlantısı yapmayı sağlayan program çalıştırılmalıdır.Bu API (Application Program Interface) programı kullanılarak istenilen herhangi bir veritabanına ulaşip işlem yapmak mümkündür. Buradan istenilen veritabanına ait sürücü seçilerek veritabanı dosyası için bir bağlantı kurulur.

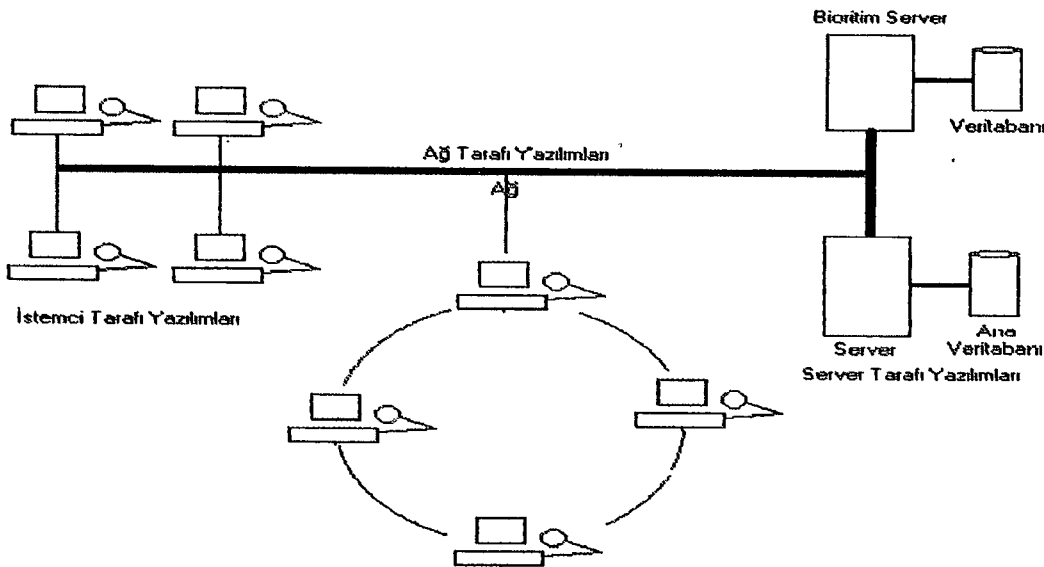
3.4.Sistemin Bütünleşik Analizi

Şekil 3.2 'de sembolik olarak görüldüğü üzere yazma ritmi tabanlı güvenlik sistemi değişik platformlara kolayca entegre edilebilir. Daha önce kabaca bahsedilen 3 ana yazılım katmanının birbirleriyle ilişkisine göz atacağız. Bunun yanında şekil 3.2'de gösterildiği gibi heterojen bir ağ ortamında tasarlanan sistemin nasıl çalışabileceği konusunda fikir yürüteceğiz.

Tasarlanan sistemin 3 ana kısımdan oluştuğu daha önce söylenmişti. Fakat bu 3 ana parçanın çalışma ve etkileşimleri konusunda daha önceki bölümlerde çok fazla bahsedilmedi. Aslında sistemin kararlı ve istenilen doğrultuda çalışabilmesi bu 3 ana parçanın birbirleriyle aralarındaki ilişkinin sağlamlığına ve uyumuna bağlı olduğu açıktır.

Sunucu tarafı yazılımları iki ayrı yazılımdan oluşur. Bunlardan birincisi kullanıcıların güvenlik bilgilerinin tutulduğu veritabanı yönetim sistemidir. Diğeri ise güvenlik programının yönetim işlemlerini yapan yönetici programıdır.

Yönetim programı sunucu tarafında bulunan ve çalışan bir programdır. Sistemin tümü gözönüne alındığında kendisinden devamlı veri talep edilen programdır. İstemci tarafından gönderilen verileri ve istekleri kendi içerisinde olan ve bölüm 4'de detaylı olarak belirtilmiş algoritmayı temel alarak işler.



Şekil.3.2. Sistemin Genel Yapısı

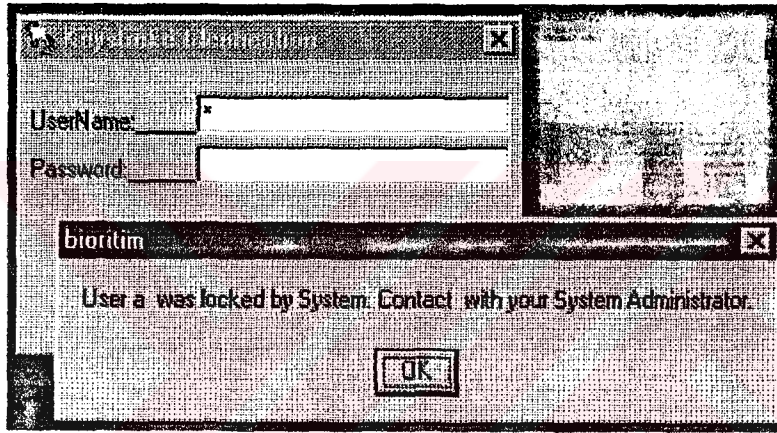
Yazma ritmi tabanlı güvenlik sisteminin çalışma adımları aşağıdaki gibidir.

- 1- İstemci program kullanıcı verilerini alır.
- 2- Ağ iletişim yazılımı verileri gönderir.(Kriptolu yada Kriptosuz)
- 3- Sunucu programı verileri alır.
- 4- Red/Kabul cevabı İstemci yazılıma gönderilir.

İstemci programın amacı verileri toplamaktır. Bunu yaparken tamamen dinamik hafıza kullanır. Bunun nedeni bölüm 4’de açıklanacağı üzere İstemci tarafı yazılımın kullanıldığı istemci bilgisayar üzerinde herhangi bir iz bırakmamasıdır. Böylece herhangi bir şekilde sisteme giriş izni olmayan birisinin İstemci tarafı bir bilgisayar üzerinden sistem hack etme olanağı ortadan kalkmaktadır. Çünkü kullanıcılara ait parametrelerin hepsi sadece sunucu bilgisayar üzerinde bulunmasıdır.

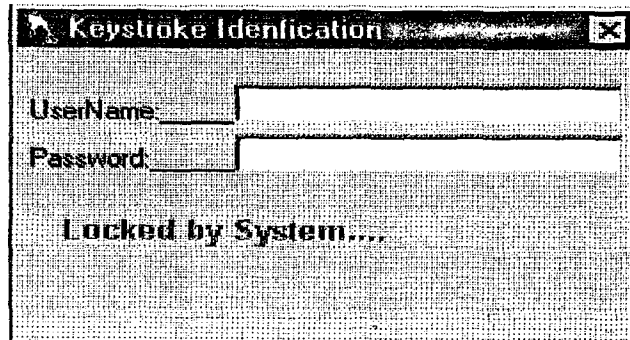
İstemci tarafında çalışan ve klasik login ekranlarını çağrıştıran yazılım her zaman dinleme modundadır. Program herhangi bir şekilde üzerinde oynama yapıldığı zaman – örneğin mouse ile tıklama yada textbox’larına bir karakter girilmesi gibi- çalışmaya başlar. Kullanıcı ismi girilmesi işlemi yapıldığı zaman İstemci programı Sunucu programa böyle bir kullanıcıya ait kaydın sistemde olup olmadığı konusunda bir sorgu gönderir.Eğer sunucudan gelen cevap olumsuzsa İstemci tarafı programı sisteme girmek isteyen kişiyi bu konuda uyaracaktır. Eğer Sunucu tarafı yazılımından gelen cevap olumluysa yani böyle bir kullanıcı sistemde kayıtlıysa İstemci programı kullanıcıyı ikinci aşamaya geçirecektir. Burada kullanıcı sisteme yazma ritmi parolasını girer. Parola girme aşaması tamamlandıktan sonra program almış olduğu parolayı ve parametrik değerleri tekrar sunucu bilgisayara gönderir. Sunucu bilgisayarda gelen parametrelerden öncelikle kullanıcının parolası kontrol edilir. Eğer verilmiş olan kullanıcıadı ile parola birbirlerini tutuyorsa Sunucu yazılımı kullanıcının biometrik parametrelerini kontrol eder.Bu kontrolün hangi kriterlere göre yapıldığı bölüm 5’de detaylı olarak anlatılmıştır. Parametre kontrolü sonucunda sunucu yazılımı kullanıcıyı tanıdığı kararına varırsa yazılım son gelen parametreleri karar verme aşamasında kullanılan referans parametreler olarak veritabanına kayıt edecektir.

Bu doğrulama işlemleri sırasında herhangi bir sebepten dolayı program kullanıcı hakkında tam karar veremezse İstemci programına kullanıcının parametre bilgilerini tekrar girmesi konusunda bir istek gönderir. Bu istek istemci program tarafından kullanıcıya bildirilir ve sisteme girebilmek için parolasını tekrar girmesi istenir. Bu aşamadan sonra yukarıda açıklanan işlemler tekrar eder. Herhangi bir sebepten dolayı yazma ritmi doğrulama işlemi tamamlanamazsa Sunucu programı kullanıcıya ait hakları geçici olarak kilitler bu aşamadan sonra sistem yöneticisi tarafından ilgili kullanıcının hesabı tekrar aktif hale getirilinceye kadar bu kullanıcı ismiyle sisteme giriş yapılamaz böyle bir istek yapıldığında sistem kullanıcıya sistem üzerinde haklarında kısıtlama olduğunu ve sistem yöneticisine başvurması gerektiği konusunda bir uyarı çıkartacaktır.



Şekil 3.3. Kullanıcı Hakkı Kilitlenmesi

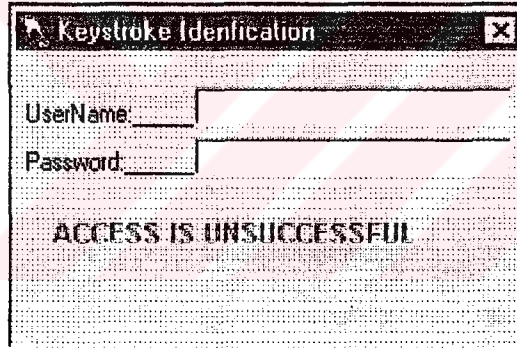
Bu uyarıdan sonra İstemci programı kendisini kilitleyecek ve kullanıcının herhangi bir işlem yapmasına izin vermeyecektir. Sistem kullanıcının form üzerine müdahale etmesini engeller yani bilgisayar kullanılamayacak hale gelir.



Şekil 3.4. Kilitlenmiş Hakka Ait Uyarı

Bu aşamadan sonra istemci programı üzerindeki bütün komponentler kullanılamaz hale gelecektir. Bu kullanıcıyı herhangi bir sebepten dolayı hesabı kilitlendiğinde sistem yöneticisine başvurmasına neden olacaktır. Bilindiği gibi genel amaçlı networklerde herhangi birisinin Kullanıcı adı ve Parolasını bilen birisi sisteme rahatça girebilir. Bu girişten ise hesabın gerçek sahibinin çoğu zaman haberi olmaz. Tasarımda böyle bir hak kilitlenme opsiyonunu yaparak biz aynı zamanda bir kullanıcının kullanıcıadı ve parolasının bir başkası tarafından bilindiği uyarısını da dolaylı yoldan kendisine bildirmiş oluyoruz. Böyle bir opsiyonla sistem içerisinde kayıtlı olan bütün kullanıcılar Kullanıcı adlarına ve parolalarına daha iyi sahip olabileceklerdir.

Herhangi bir kullanıcının Kullanıcı adı ve Parolasıyla sisteme girmeye çalışan birisi sistem tarafından tanınmadığı zaman şekil 3.5'deki gibi bir uyarıyla karşılaşacaktır.



Şekil 3.5. Başarısız Giriş

Şekil 3.5 'de görülen "Access is Unsuccessful" yazısı belli bir periyotta yanıp sönecek ve sisteme girmeye çalışan kullanıcıya uyarı verecektir.

Yukarıda anlatılan programlar Windows ortam baz alınarak hazırlanmıştır. Fakat günümüzde bir çok ağ ortamında değişik ortamlar ve platformlar mevcuttur. Örneğin bir ağ sisteminde Windows, Dos, Novell, Unix, Linux * gibi bir çok işletim sistemi ve bununla beraber olarak bir çok mimari bulunabilir. Şekil 3.2'in aslında anlatmak istediği de tasarlanan yazma ritmi sisteminin bir çok platform üzerinde kullanılabilirliğinde olabileceğidir. Biz tezimizin konusu itibarıyla böyle bir

çalışmayı deneysel yapmadık fakat tasarlanmış olan algoritmanın bahsedilen platformlar üzerinde yeniden derlenmesiyle İstemci programı bahsedilen bütün platformlar üzerinde rahatça çalışabilir. Programı başka platformlarda Bir diğer yöntem ise java gibi kodları interpreteri yardımıyla bütün platformlarda çalışan bir programın yazılarak bu platformlarda kullanılması olabilir. Gerçektende günümüzde java bir programlama dili olarak platform üstü programlar geliştirmek için geliştiricilere çok fazla olanaklar sunmaktadır. Bir middleware program olarak ODBC yerine JDBC kullanmak programın çalışma mantığı ve performansı açısından hiç bir olumsuzluğa neden olmayacaktır.

3.5. YARDS Yazılım Mimarisi

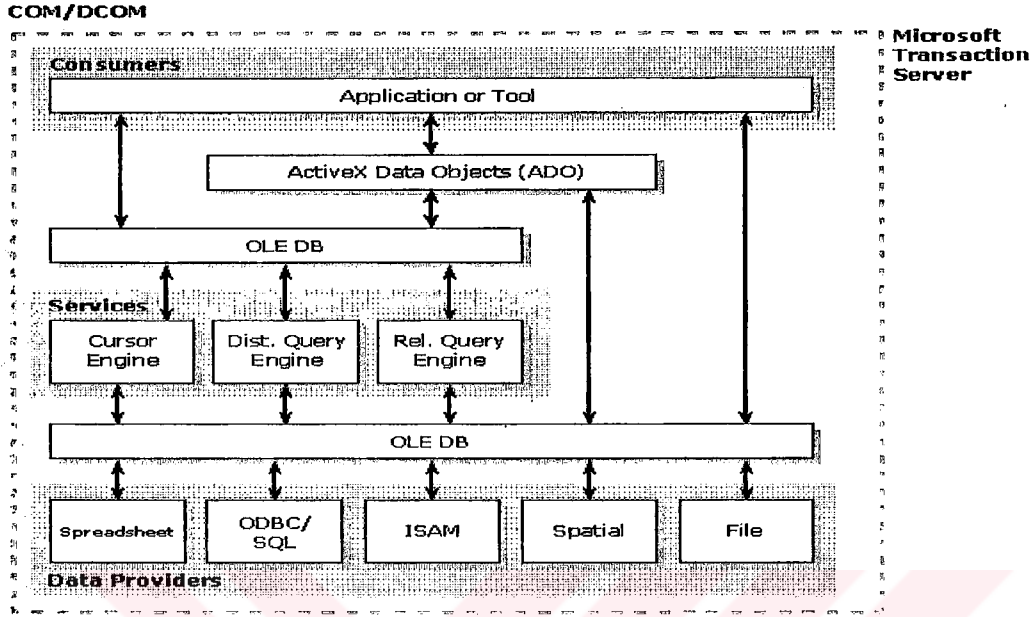
Programın geliştirme ortamı olarak Windows® ortamının kullanıldığı daha önceki kısımlarda belirtilmişti. Test yazılımın geliştirilmesi İstemci ve Sunucu yazılımlarının geliştirilmesi aşamalarında Visual Basic 6.0 programı kullanıldı. Sunucu tarafında kullanılan veritabanı yönetim yazılımı olarak önce Access 7.0 programı daha sonrada Oracle 8.0 programları kullanıldı.

Programın geliştirme ve middleware yazılımla iletişimini sağlamada son yıllarda hızla kullanımı yaygınlaşan ve güçlü bir veri iletişim standardı olan ADO 2.0, ActiveX® Data Objects kullanıldı.

3.5.1. ADO

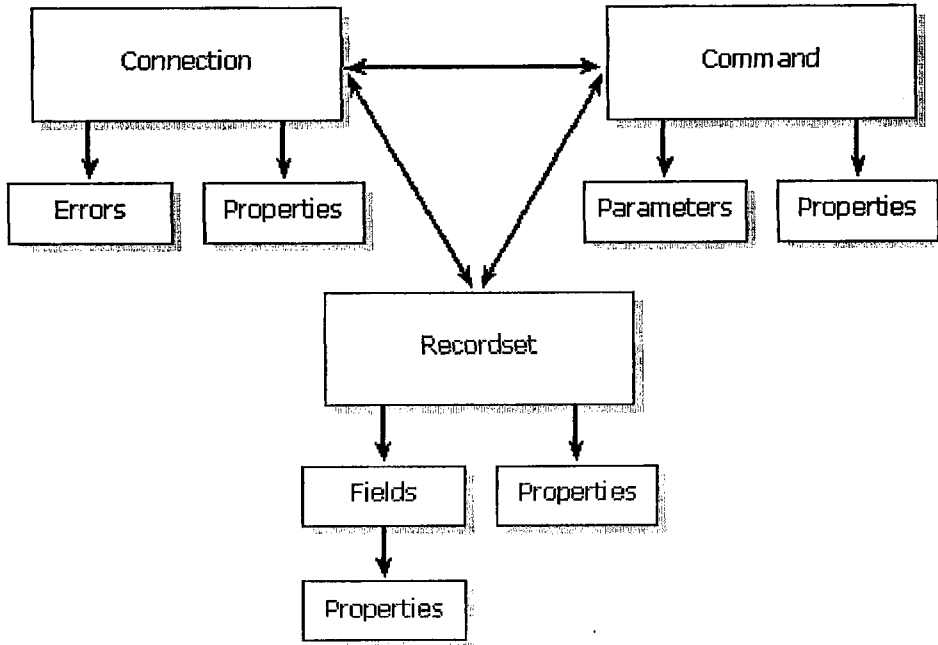
ADO ,ilişkisel ve ilişkisel olmayan veri kaynaklarına çok katmanlı yapı mimarisinde ulaşım olanağı sağlayan bir teknolojidir. Bu teknoloji kullanılarak Visual Basic,C++,java gibi bir çok programlama aracı kullanılarak veri kaynaklarına ulaşılabilir.[12] Gerçektende dağıtık veri kaynaklarına ulaşmak internet ve intranet üzerinde çalışan programlar geliştirmek açısından bakıldığında

ADO geliştiricilere daha eski teknolojilere oranla çok daha esnek davranma imkanı sunmaktadır.



Şekil 3.6. DCOM Mimarisi

Şekil 3.6 Microsoft firmasının COM (Component Object Model) olarak adlandırılan mimarisine ait bir kavramsal çizimi göstermektedir. Burada görülmekte olan bütün parçalar(component) sadece kendilerine verilmiş olan sınırlı görevleri yapmaktadırlar. Ancak parçaların tümünün oluşturduğu sistem kendisine verilmiş olan görev olan veritabanına ulaşp işlem yapma imkanı vermektedir. Örneğin Data Providers olarak adlandırılmış olan kısım veri kaynaklarını ifade eder bu veri kaynağı bir ilişkisel veritabanı olacağı gibi bir masaüstü veritabanı yada bir excel sayfası da olabilir. Şekil 3.6 'daki çizimde alt kısımda gösterilen objeler daha düşük seviyede olan objelerdir.Bu sebeple daha hızlı çalışırlar.Örneğin uygulama programından doğrudan OLE DB objesine ulaşmak arada ADO objesini kullanarak ulaşmaktan daha hızlı olacaktır. Fakat esneklik açısından olaya yaklaşıldığında birinci yöntem çok iyi bir alternatif sunmamaktadır.



Sekil 3.7-ADO Mimarisi

Şekil 3.7 'de ADO 'nun parçaları gözükmemektedir. Teknik olarak ADO 3 ana parçadan oluşur. Bunlar;

- 1- Connection(Bağlantı)
- 2- Command(Komut)
- 3- Recordset(Kayıt)

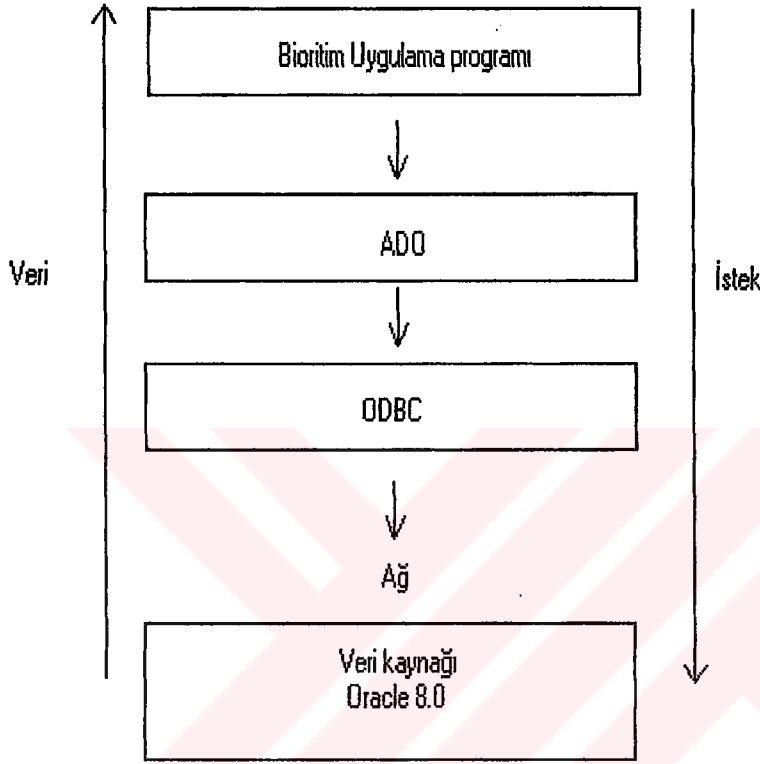
Bu 3 ana parçada şekil 3.7 'de görüldüğü üzere alt parçaların birleşmesiyle oluşmuştur. Mikro düzeyde bakacak olursak bu alt parçalarki sayıları 7 dir.ADO'yu oluşturan parçalar olarak adlandırılabilirler.

Bağlantı parçasında bağlantı yapılacak veri kaynağı ve bağlantının cinsi ile ilgili bilgiler bulunur. Örneğin bağlantı bir ODBC için DSN ismi ,Kullanıcı adı ve parola gibi.

Komut parçası ise işletilecek komutların yazıldığı yer olarak adlandırılabilir. Burada SQL olarak adlandırılan komutlar saklanabilir.

Kayıt parçası ise bir verikaynağından döndürülen kayıtların saklandığı ve değiştirildiği kısımdır.

Şekil 3.8 Yazma ritmi programının ADO ve ODBC mimarisinde kavramsal tasarım şekli gösterilmiştir.

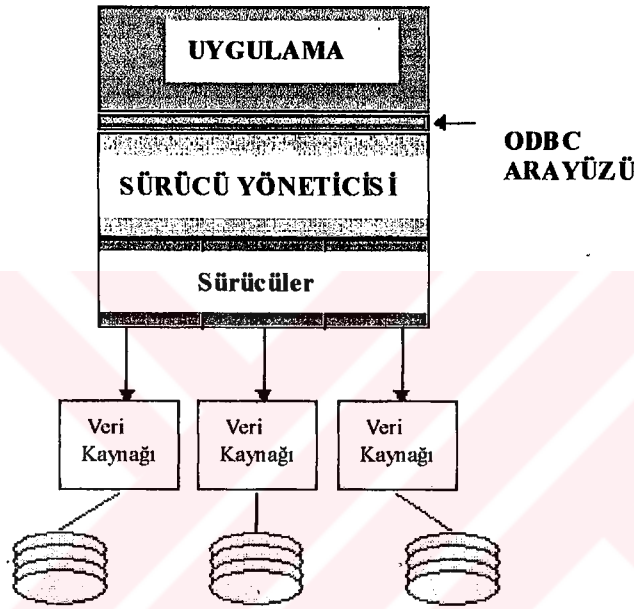


Şekil 3.8. Yazılım Katmanları

3.5.2. ODBC (Open Database Connectivity)

ODBC programı Microsoft firması tarafından, ilişkisel ve ilişkisel olmayan veritabanlarına ulaşmak için geliştirilmiştir. Özellikle heterojen veritabanı mimarilerinde kullanımı bir çok esneklikler sağlamaktadır. Tasarlanan Yazma ritmi programında da ODBC kullanarak daha önceki bölümlerde bahsedilen network iletişim yazılımını kendimizin yazmamıza gerek kalmadı . Gerçekten de ODBC bizim tasarladığımız gibi ilişkisel veritabanlarına ulaşip işlem yapmada oldukça kullanışlı bir programdır.

Şekil 3.9 'da ODBC mimarisi hakkında bir gösterim vardır. ODBC yapısında ADO'da olduğu gibi katlı bir yapı ifade eder. Burada amaç uygulama programıyla veri kaynağı arasındaki iletişimi sağlamaktır. Bunu yapmak için ODBC programının bağlantının yapılacağı kaynağın DSN(Data Source Name)'ini bilmesi gerekir. Kullanılan diğer veri kaynaklarının cinsine göre de farklı bazı ek bilgiler gerekebilir.Örneğin eğer uygulama kodunun içerisinde belirtilmemişse veritabanına ulaşma hakkı olan bir kullanıcının kullanıcı adı ve parolası gibi opsiyonlarda bağlantı için gereklidir.



Şekil 3.9.ODBC Mimarisi

ODBC mimarisi oldukça basit bir mimaridir. Mimarinin en üst katmanında uygulama programı yer alır. Uygulama programının altında ise ODBC arabirimi başlar,bu arabirim uygulama programı ve ODBC Sürücü Yöneticisi ile iletişim kurmayı sağlayan ara bir katmandır. Sürücü Yöneticisi kendi alt katmanında yer alan ve doğrudan veri kaynaklarına bağlanmış olan sürücülerini kontrol eder.İlgili sürücülere sorguları gönderir. Gelen cevap ve kayıtları(Recordset) ya buffer da saklar yada uygulama programına gönderir.

3.6. Yazma Ritmi Doğrulama Sistemi (YARDS) 'nin Yapısı

YARDS, Yazma Ritmi Doğrulama Sistemi'nin ilk harflerinden oluşmuştur ve G.Y.T.E'de yazma ritmi tabanlı tasarlanan sisteme verilmiş olan isimdir.

Yazma ritmi tabanlı güvenlik sistemlerinde önemli olan ne yazıldığıнын yanında, yazılan karakterlerin nasıl yazıldığıdır. Yani aynı zamanda kullanıcı, parolasını özel bir şekilde yazarak kodlamaktadır. Bu çalışmada tasarlanan yöntemde, bir sisteme giriş için gerekli olan parola iki şekilde kontrol edilir.

- 1- Sisteme bağlanan kullanıcının parametreleri ("Kullanıcı Adı", "Parola") kontrol edilir.
- 2- Kullanıcı eğer sisteme kayıtlı bir kullanıcıysa kullanıcının yazma ritmi kontrol edilir.
- 3- Sisteme giriş izni verilir yada verilmez.(Gerekirse kullanıcının hesabı kilitlenir.)

3.7. Sistemin Kuramsal Yapısı

Tasarlanan Yazma ritmi Temelli Bilgisayar Güvenlik Sistemi'nin yapısı aşağıdaki gibi ifade edilebilir. Aşağıda verilen 4 işlem, blok diyagramı Şekil 3.10'da verilmiş olan yazma ritmi yönteminin temelini oluşturur.

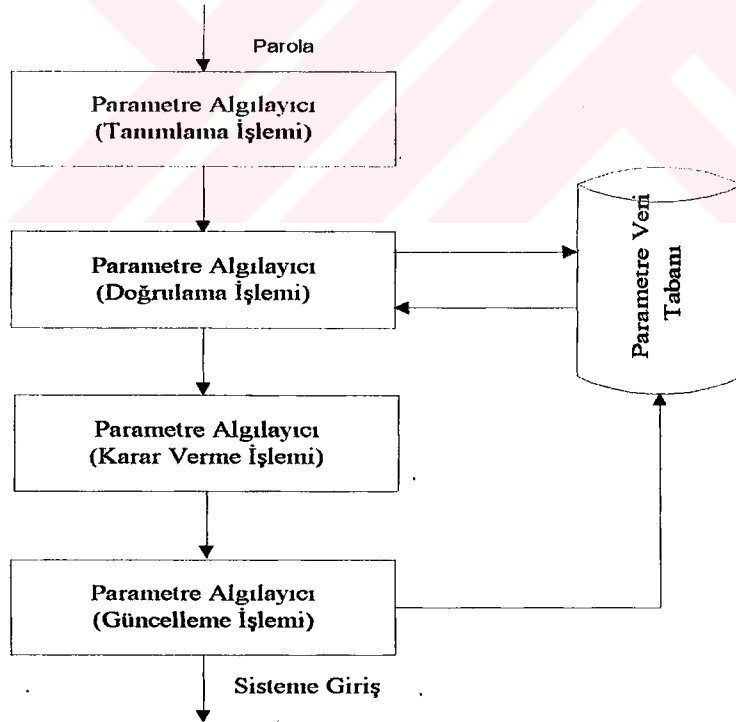
- 1- Tanımlama İşlemi
- 2- Doğrulama İşlemi
- 3- Karar Verme İşlemi
- 4- Güncelleme İşlemi

Yukarıda verilen yazma ritmi işlem adımları, gerçekleştirilen yazma ritmi yazılımı içerisinde hiyerarşik olarak işletilirler. Verilmiş olan işlemler birbirlerine doğrudan bağlı olarak çalışırlar. Her bir işlem ihtiyacı olan parametreleri bir önceki işlemin sonucundan alır. Dört işlemten oluşan sistemin sağlıklı çalışması alınan verilerin kesin doğruya yakın bir biçimde algılanması ve işlenmesiyle elde edilir. İşlemler

içerisinde ortaya çıkabilecek herhangi bir aksaklık sistemin tümünün kararsız ve yanlış cevap verecek şekilde çalışmasına sebep olacaktır. Karar Verme İşlemi, Sistemin güvenliğinin artırılması için ihtiyaca göre özelleştirilebilir. Karar Verme İşlemindeki esneklik tamamen sistemin güvenliğinden sorumlu olan yöneticinin inisiyatifindedir. Sistemin çalışması aşağıda açıklanmıştır.

3.7.1.Tanımlama İşlemi

Parametre algılayıcısı olarak adlandırabileceğimiz Tanımlama İşlemi sistemin en önemli adımlarından bir tanesidir. Çünkü bu adım, sisteme girmek için, gerek sisteme ulaşma hakkı olan kullanıcıların gerekse yetkisiz kişilerin sistemle ilk temas noktasıdır. Burada karşılaşılan klasik bir login ekranıdır. Buraya bir giriş yapıldığı zaman (Kullanıcı Adı ve Parola) sistem çalışmaya başlar. İşlemin görevi kendisine girilen bilgileri ve kendisinin giriş ritminden oluşturduğu bilgileri kayıt etmesidir. Bu işlem sonunda sisteme giriş yapmak için talepte bulunan kullanıcıya ait bilgiler daha sonraki işlemlerde kullanılmak üzere tanımlanırlar.



Şekil 3.10. YARDS'a Ait Alt İşlemler

3.7.8. Doğrulama İşlemi

Doğrulama, Tanımlama İşlemi sonunda elde edilen parametrelerin referans değerleriyle karşılaştırıldığı adımdır. Burada öncelikle yapılan işlem, klasik yapıda olduğu gibi sistemde girilen isimde bir kullanıcı olup olmadığının testi yapılır. Eğer sonuç olumluysa, sırasıyla Kullanıcıya ait parolanın doğru olup olmadığı kontrol edilir. Buraya kadar bir sorun yoksa bioritmik parametrelerin karşılaştırılması aşaması gelir. Bunun için veritabanı içerisinde referans değerler alınır elde edilen değerlerin bu referans değerleriyle uyumlu olup olmadığı bir algoritmaya göre test edilir ve vektörel analizi yapılır. Bu işlem sonucunda daha anlamlı parametreler elde edilirler. Bu yeni parametreler bir sonraki işlemde karar verme mekanizmasında kullanılabilecek anlamda parametrelerdir.

3.7.9. Karar Verme İşlemi

Doğrulama İşlemi sonunda elde edilen parametreler Karar Verme İşlemi'nin içerisinde yer alan algoritmaya göre kabul edilebilir veya edilmez. Karar Verme İşlemi'nin 4 tane bulunabileceği durum vardır. Bunları şöyle sıralayabiliriz.

- 1- Yüksek Düzey Güvenlik
- 2- Orta Düzey Güvenlik (Tanımlı olan Güvenlik Düzeyi)
- 3- Düşük Düzey Güvenlik
- 4- Yazma ritmi Güvenlik Sistemi kapalı.

Yüksek Düzey Güvenlik: Sistemin Yüksek Düzey Güvenlik ayarında bulunması sistemin referans değerinden sadece son derece az sapmaları kabul etmesi demektir. Tasarlanan sistemde bu oran yüzde 5*' den küçüktür. Yani referans parametreleri ile girilen parametreler arası bu değer üzerinde olduğunda kullanıcı sisteme giremeyecektir.

Orta Düzey Güvenlik: Bu ayar sistemin tanımlı olarak bulunduğu güvenlik ayarıdır. Bu seviye ayarda kabul edilebilir sapmanın miktarı daha da arttırılmıştır. Bu oran yüzde 8*' den küçüktür.

Düşük Düzey Güvenlik: Bu sistemin güvenliğinin oldukça düşük olduğu bir güvenlik ayar miktarıdır. Bu oran yöneticinin isteğine göre değiştirilebilir.

Yazma Ritmi Güvenlik Sistemi Kapalı: Bu opsiyon bazı sebeplerden dolayı yazma ritmi güvenlik sisteminin kapatılması gerektiği zamanlarda geçerlidir. Sadece Kullanıcı Adı ve parola yazılarak (eğer sistemde kullanıcı kayıtlıysa) sisteme girilebilir. (* Yukarıdaki oranlar sistem yöneticileri tarafından değiştirilebilirler.)

3.7.10. Güncelleme İşlemi

Tasarımı yapılan sistem dinamik bir sistemdir. Bunun amacı kullanıcıların zamanla yazmaritimlerinin değişebileceği gerçeğidir. Sisteme girme talebinde bulunan kullanıcının eğer parametreleri kabul edilebilir sınırlar içerisinde olursa Karar Verme İşlemi ,Tanımlama İşlemi'nden beri saklanan parametreleri Güncelleme İşlemi'ne iletir. Güncelleme İşlemi'nin herhangi bir karar verme yada fonksiyon türetme gibi bir görevi yoktur. Sadece almış olduğu parametreleri yeni referans parametreler olarak veritabanı'na günceller. Kullanıcının bir sonraki sisteme girme isteğinde bu parametreler referans parametreler olarak gönderilirler. Bu yapı, sistemi zamanla öğrenen bir yapıya kavuşturmaktadır.

3.8. Tasarlanan Sistemin Matematiksel Modeli ve Algoritması

Bioritmik sistem daha önceki bölümde bahsedildiği ve Şekil.3.10' da gösterildiği üzere ihtiyacı olan parametreleri kendi içerisinde üretmektedir. Sistemin girdi olarak alabildiği iki adet parametre vardır. Bunlar B(I) dizisi şeklinde ifade edilebilecek tuş basma zamanları ve G(I) şeklinde ifade edilebilecek tuş gecikme zamanlarıdır. Sistemin diğer aşamalarında ihtiyacı olan diğer parametreler ve fonksiyonlar bu iki diziden türetilirler.

Yazma ritmi sisteminin tasarımında matematiksel modelleme yapılırken aşağıdaki notasyonlar kullanılmıştır.

$B(I)$ = Herhangi bir I . tuşa basma işlemi yapılırken geçen süre.

$G(I)$: $I-1$.tuştan I . tuşa basılırken arada geçen gecikme süresi.

Θ_b :Referans B vektörü ile ölçülen B vektörü arasındaki açının farkı.

Θ_g : Referans G vektörü ile ölçülen G vektörü arasındaki açının farkı.

Γ_b : B vektörü için ağırlık çarpanı.

Γ_g : G vektörü için ağırlık çarpanı

ξ : Karar fonksiyonu

Ψ_b : B vektörü

Φ_g : G vektörü

Ω : ξ için ağırlık çarpanı

B_i 'nin N boyutlu bir uzayda birbirlerine dik ve ortogonal boyutlardan herhangi birisi olduğu kabulünü yaparsak; Şifre Uzunluğu olan N , N boyutlu bir uzaya karşılık gelmektedir. Yani şifrenin her bir karakteri N boyutlu uzayın bir boyutunu tasvir etmektedir.

Sistemin kullanıcı girişinden türettiği parametrelerden B ve G dizileri tanımlanabilir.

$$B = \{B(1), B(2), B(3), B(4), B(5), \dots, B(N-\Theta), \dots, B(N)\} \quad 3.1$$

$$G = \{G(1), G(2), G(3), G(4), G(5), \dots, G(N-\Theta), \dots, G(N-1)\} \quad 3.2$$

Yukarıda tanımlanan B dizisinin elemanlarından bir N boyutlu bir vektör türetilir.

$$\Psi_b = \sum B(i) I_i \quad ; \quad N \text{ Boyutlu } \Psi_b \text{ vektörü şeklinde tanımlanabilir.} \quad 3.3$$

$$\Phi_g = \sum G(j) I_j \quad ; \quad N-1 \text{ Boyutlu } \Phi_g \text{ vektörü şeklinde tanımlanabilir.} \quad 3.4$$

Ψ_b , Φ_g vektörleri sistemden gerçek zamanlı olarak türetilmiş olan vektörlerdir. Aynı zamanda sistemin referans olarak kullandığı ve sistemin daha önce üretmiş olduğu iki adet benzer vektör şöyle ifade edilebilir.

$\Psi_{br} = \sum B_r(i) I_i$; N Boyutlu Ψ_{br} vektörü (Referans Vektörü) şeklinde tanımlanabilir. 3.5

$\Phi_{gr} = \sum G_r(j) I_j$; N-1 Boyutlu Φ_{gr} vektörü (Referans Vektörü) şeklinde tanımlanabilir. 3.6

A ve B iki benzer ve aynı boyutta vektör olmak üzere ; N boyutlu uzayda iki vektör arasındaki açı aşağıdaki gibi gösterilebilir.

$$\cos \theta = AB / [(\sum A^2)(\sum B^2)]^{1/2} \text{ şeklinde ifade edilebilir.} \quad 3.7$$

O halde;

$$\cos \theta_b = \Psi_b \Psi_{br} / [(\sum \Psi_b)^2 (\sum \Psi_{br})^2]^{1/2} \quad 3.8$$

$$\cos \theta_g = \Phi_g \Phi_{gr} / [(\sum \Phi_g)^2 (\sum \Phi_{gr})^2]^{1/2} \quad 3.9$$

Burada Ψ_b ve Ψ_{br} vektörleri birbirlerine ne kadar benzerlerse $\cos \theta_b$ 'nin değeri o kadar 1'e yaklaşır. $\cos \theta_b$ 'nin teorik (ideal) değeri 1 dir.

Aynı şekilde Φ_g ve Φ_{gr} vektörleri birbirlerine ne kadar benzerlerse $\cos \theta_g$ 'nin değeri o kadar 1'e yaklaşır. O halde $\cos \theta_g$ 'nin teorik(ideal) değeri de 1'dir.

$\cos \theta_b$ değeri için ağırlık çarpanı Γ_b ; $\cos \theta_g$ değeri için ağırlık çarpanı Γ_g olarak tanımlanırsa;

Γ_b ve Γ_g terimleri arasındaki ilişki aşağıdaki gibi tanımlanabilir.

$$\Gamma_b + \Gamma_g = 1 \quad 3.10$$

$$\Gamma_g = 1 - \Gamma_b \quad 3.11$$

Γ_b ve Γ_g karar fonksiyonuna direkt olarak etki eden çarpanlardır. Yukarıdaki ifadelerden ξ karar fonksiyonu türetilir.

$$\xi = \Gamma_g \cos \theta_g + \Gamma_b \cos \theta_b \quad 3.12$$

Ψ_b, Ψ_{br} ve Φ_g, Φ_{gr} vektörleri arasındaki açı farkının ağırlık fonksiyonlarıyla çarpımından ibarettir.

3.11 ve 3.12 'den aşağıdaki bağıntı elde edilir.

$$\xi = \Gamma_g (\cos \theta_g - \cos \theta_b) + \cos \theta_b \quad 3.13$$

3.13'de ifade edilen ξ fonksiyonu bioritmik tanımlama için ayırteci bir karar verebilecek olan karar fonksiyonudur. Bu karar için gerekli olan şart şöyle ifade edilebilir. i değeri sistemin herhangi bir zamandaki durumunu ifade etmek üzere ;

ξ_{i-1} : Veritabanından alınan i . durumdan önceki girişe ait karar fonksiyonu olmak üzere;

Ω : Ağırlık çarpanı, $0 < \Omega < 1$, değerleri arasındadır.

$$\xi_{i-1} - (\Omega \xi_{i-1}) < \xi_i < \xi_{i-1} + (\Omega \xi_{i-1}) \quad 3.14$$

Eğer elde edilen değer 3.14 de ifade edildiği gibi kabul edilebilir bir aralık içerisindeyse biometrik sistem bu kullanıcıya giriş izni verecektir.

3.9. Öğrenen Sistem Tasarımı

Şekil 3.11 'den de açıkca görüldüğü üzere sistem kendi içerisinde bir karar verme inisiyatifine sahiptir. Sistemin karar verme süresi kullanıcıların yazmaritimlerdeki düzene bağlıdır. İyi bir bilgisayar kullanıcısı için bu süre kısadır. Çünkü bu kullanıcının oturmuş, karakteristik bir bioritmi vardır. Dolayısıyla algoritmada karar verilmesi için gerekli olan kriterler çok daha çabuk sağlanır. Diğer

tarafından bilgisayarla çok yakın ilişkisi olmamış birisinin kendisine has ayırt edici biyometrik özelliği belirgin olmadığı için sistem bu kullanıcıyı tanımada çok daha fazla zamana ihtiyaç duyacaktır.[8]

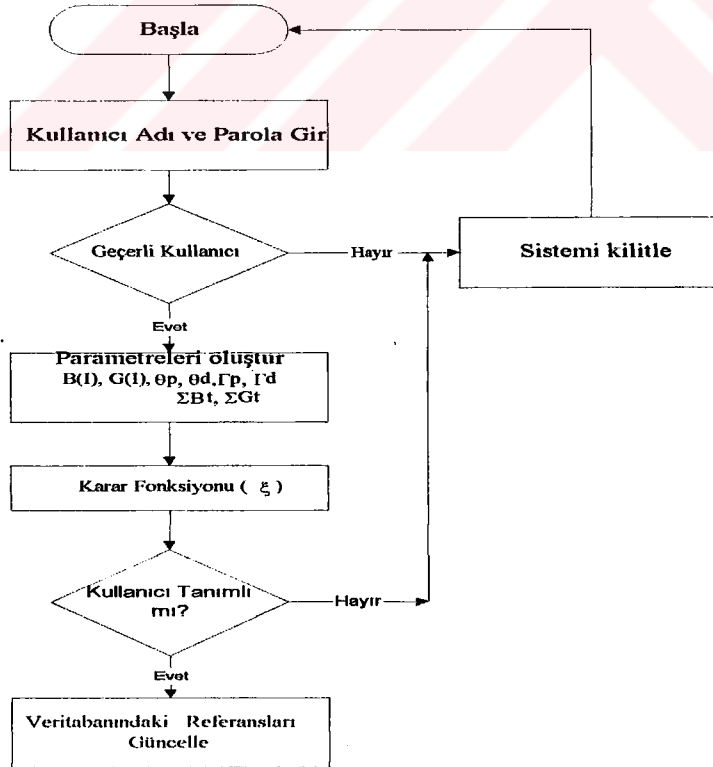
Tasarlanan sistemin en önemli özelliği zamanla kullanıcının yazmaritimlerindeki değişimleri yakalamasıdır. Biyometrik tanımlayıcı olarak yazma ritmi yöntemi insanların psikolojik durumlarına bağlı olarak oldukça değişken bir olgudur. Dolayısıyla sistem tarafından kullanılan referans parametreleri bir önceki bölümde de ifade edildiği üzere devamlı güncellenmek zorundadır. Bu güncelleme aslında sistemin kullanıcının yazmaritimlerini zamanla öğrenmesi manasına gelmektedir.

Aşağıda öğrenen sistem tasarımının çalışma sistematığını anlatan bir akış diyagramına yer verilmiştir. Çalışma prensibi akış diyagramından da görüldüğü üzere oldukça basittir. Öncelikle kullanıcı yazılıma kullanıcı adını girer. Yazılım eğer kullanıcı sistem tarafından tanınıyorsa biyometrik parolasını girmesini ister. Eğer kullanıcı adı sistemde kayıtlı değilse sistem böyle bir kullanıcının mevcut olmadığını söyleyerek sistemi kilitleyecektir. Kullanıcı adı doğrulandıktan sonra girilen parolayla beraber kullanıcının biyometrik parametreleri yazılım tarafından okunur. Bundan sonraki aşama yazılımın yapacağı işlem basamaklarıdır. Yazılım öncelikle parolanın mevcut kullanıcı için geçerli parola olup olmadığını inceler eğer cevap olumluysa parametreleri incelemeye başlar. Bunun için öncelikle yaptığı bu parametreleri algoritma içine sokar ve sonuçta B ve G dizilerinden başka P fonksiyonu ve Ω gibi parametreleri elde eder. Bunları hesaplarken karşılaştırma ve karar verme işlemlerinde kullanılması için gerekli olan referans parametreler veritabanından yazılıma gönderilir. Doğrulama ve karar verme işlemi istenen kriterlere uygun sonuçlar verirse yazılım kullanıcıyı sisteme sokar ve daha önce sahip olduğu haklar doğrultusunda kaynaklara erişim hakkı verir. Eğer doğrulama ve karar verme işlemi esnasında herhangi bir sebepten dolayı bir güvensizlik meydana gelirse yazılım kullanıcıdan parolasını tekrar girmesini ister bu bir nevi döngüdür. Bu döngünün sayısı yazılım içersinden opsiyonel olarak ayarlanabilir. Döngü sayısının tanımlı değeri 3'dür. Eğer kullanıcı parolasını 3 kez yanlış girerse o kullanıcı adına ait haklar geçici olarak kısıtlanırlar. Kullanıcı adının tekrar aktif olması için sistemde yönetici olarak tanımlı bir kullanıcının hakkını tekrar aktif hale getirmesi gerekir.

Öğrenen sistem karar aşamasında kısım 3.11’de bahsedilen algoritmayı kullanır. Eğer elde edilen değer 3.15 de ifade edildiği gibi kabul edilebilir bir aralık içerisindeyse biometrik sistem bu kullanıcıya giriş izni verecektir.

Yeni bir kullanıcı için problem bu parametrelerin ilk kullanımda veritabanına nasıl yerleştirileceği sorusudur. Çünkü yazma ritmi parametrelerinin programa tanıtılması için bir süreç gereklidir. Bunun için genelde kullanılan yöntem kullanıcıların yazma ritimlerini, yazma basınçlarını algılamak için belli bir süre için yazma işlemini yapmalarını sağlamaktır [9]. Bizim tasarlanan biometrik sistemde kullandığınız yöntem On-Line [10] olarak tabir edilen bir tanımlama sistemi örneğidir. Dolayısıyla referans olarak kullanılacak parametrelerin sisteme çok iyi öğretilmesi gerekmektedir. Bunun için yapılması gereken aşamalar aşağıda belirtilmiştir.

- Kullanıcı sisteme kaydını yaptırır. (Kullanıcı Adı, Parola)
- Kullanıcı sisteme kendini tanıtırma aşamasına geçer. (Test yazışları)
- Sistem yazan kişiyi tanıdığına karar verir ve Test yazışlarını durdurur.
- Geçerli parametreleri veritabanına yerleştirir.



Şekil 3.11. Biometrik Sisteme Ait Akış Diyagramı

Kullanıcı yazmaya başladığı zaman program geri planda bir öğrenme alt programını çalıştırır. Kullanıcının yazdığı bütün parametreler veri tabanına kayıt edilirler. Kullanıcıya göre değişen bir sayıda program deneme yapılmasına karar verir yapılan denemelerin stabilleşmesinden sonra program deneme yapılmasına son verir. Burada ki genel kriter 3.15 formülünde gösterildiği üzere ;

$$\xi_{i-1} - (\Omega \xi_{i-1}) < \xi_i < \xi_{i-1} + (\Omega \xi_{i-1})$$

şartlarına uygun bir durumun oluşmuş olmasıdır. Test yazışları bu kritere uygun hale geldiğinde program kullanıcıya ait yazma parametrelerini öğrendiği kararına varır. Yukarıdaki ifade göstermektedir ki sistem karar verme aşamasında da öğrenme aşamasında 3.15’de ifade edilen şartı kullanır. Bu da öğrenme ve öğrenme aşamasının sonunda yani karar verme aşamalarında bir mantık bütünlüğü oluşmasına neden olmaktadır.

Aşağıda örnek olarak bir kullanıcının öğrenim aşamasına ait parametreler ve bu parametrelere karşılık gelen B ve G dizileri verilmiştir. Görüldüğü üzere B ve G dizileri birbirlerine oldukça yaklaştırmışlardır.

Öğrenim aşamasında kullanıcıların yazmaritimleri başlangıçta düzgün olmayabilir. Bu Kullanıcı bilgisayar kullanırken ki psikolojik durumlarından bilgisayar kullanma alışkanlıklarının yetersizliğine kadar bir çok sebeple bağdaştırılabilir.

B parametreleri,G parametrelerine nazaran her zaman daha dağınık çıkmaktadır. Öğrenim aşamasının belli bir aşamasından sonra öğrenme alt programı kullanıcıya parametrelerini öğrendiği mesajını verip son parametreleri veritabanına referans değerler olarak kopyalıyacaktır.

B ve G parametreleri aşağıda verilmiştir.

Tablo 3.2. B Dizilerine Ait Öğrenme Parametreleri

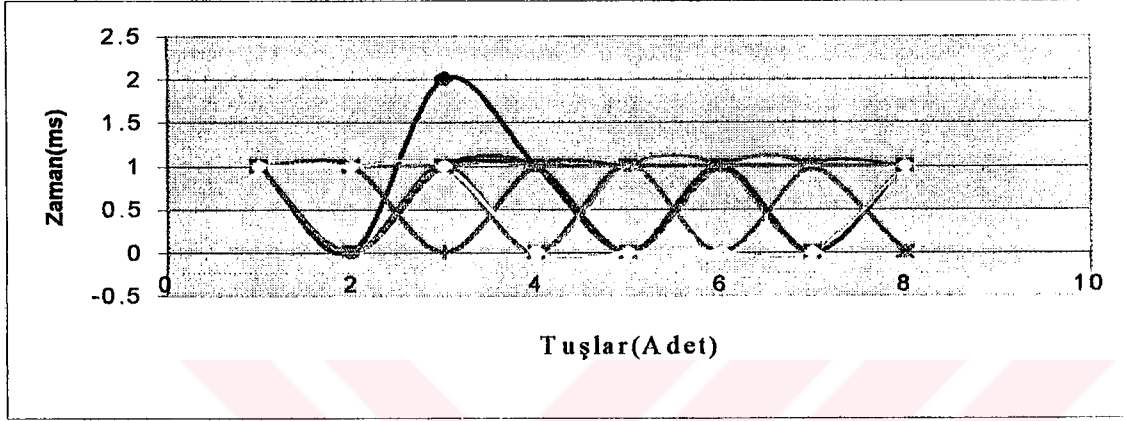
S1	S2	S3	S4	S5	S6	S7	S8
1	2	1	1	1	1	1	2
1	0	1	0	1	1	1	1
2	3	2	1	0	2	1	1
2	2	2	1	1	1	1	1
1	1	0	2	1	0	1	1
2	1	2	1	2	1	1	2
2	1	2	1	2	2	1	1
1	1	0	1	2	1	1	1
2	1	2	1	2	2	1	1
1	2	2	1	2	1	1	1
1	1	2	2	1	1	2	1
2	1	1	1	1	2	2	1
1	1	2	1	2	1	2	1
1	1	1	0	2	1	1	1
3	1	0	2	1	2	1	1
2	1	1	2	1	1	1	1
1	2	1	1	1	2	2	2
1	1	2	1	1	1	1	1
1	1	1	1	1	1	0	1
2	1	1	1	2	1	1	1
1	0	1	0	1	1	1	1
2	1	1	1	1	1	2	1
3	1	1	1	1	1	2	1
2	1	1	0	1	1	1	1
3	1	0	1	1	1	0	1
1	0	1	0	1	0	1	0
1	0	1	0	1	1	1	1
2	1	1	1	1	1	1	1
1	0	1	2	1	1	1	1
2	1	1	1	1	1	1	0
2	1	1	0	1	1	1	1
1	0	1	0	1	2	1	1
2	1	2	1	1	0	1	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	0	1	0	1	0	1	0
1	1	0	0	1	1	1	1
1	0	1	0	1	0	1	0
1	1	0	0	1	1	1	1
0	1	1	1	1	0	1	1
1	1	0	1	0	1	0	1
1	0	1	1	1	0	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	0	1	1
1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	1
1	1	1	0	0	0	1	1

Tablo 3.3. G Dizilerine Ait Öğrenme Parametreleri

[illegible]

G dizilerinin değeri belli bir süreden sonra belli değerler etrafında salınmaktadır. Bu ise yazma ritmi yöntemiyle bir güvenlik sistemi tasarlamak için yeterli imkanı vermektedir. Bu salınımlar kullanıcıların yazma ritimlerinin tespiti için gerekli parametreleri vermektedir.

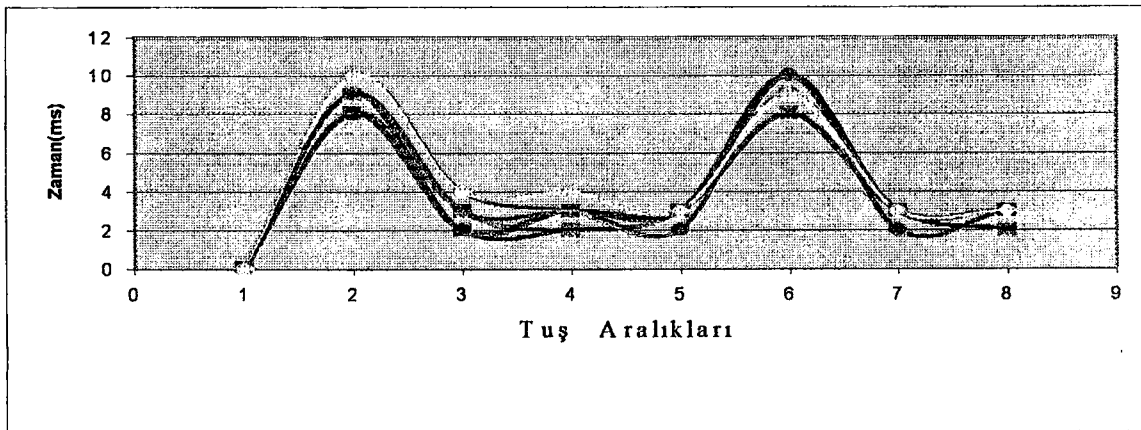
Bu salınımlara ait grafiksel gösterimler detaylı olarak gösterilmiştir. Grafiklerden görüleceği üzere B dizilerine bağlı öğrenme parametreleri G dizilerine göre daha düzensizdir.



Şekil 3.12. B Dizilerine ait Grafik Gösterilim

B dizilerinin tespitinde fiziksel olgu bir tuşa basarkenki geçen süredir. Dolayısıyla B dizilerinin sayısal büyüklüğü G dizilerine göre oldukça küçüktür. G dizileri iki tuş basma aralığında geçen sürelerdir.

B ve G dizilerine ait grafikleri çizecek olursak Şekil 3.12 ve Şekil 3.13 grafiklerine benzer şekiller elde edebiliriz.



Şekil 3.13. G Dizilerine Ait Grafik Gösterilim

Şekil 3.12'den görüldüğü gibi B dizilerine ait grafik kararsız bir durum sergilemektedir. Şekil 3.13'den anlaşıldığı üzere G Dizilerine ait karalı durumlarda grafikler birbirleriyle özdeştir.



4. YAZMA RİTMİ DOĞRULAMA SİSTEMİ (YARDS)'NE AİT DENEYSEL SONUÇLAR

Bu bölümde tasarlanan sistemin sabit parametrelerinin tespiti ve tasarımı yapılmış olan yazılımın güvenilirliğinin testi için bir çalışma yapılmıştır. Değişik kişiler üzerinde yapılan çalışmada yetkisiz kullanıcıların sisteme giriş yapmaları istenmiş kendilerine sisteme kayıtlı olan bir kullanıcı adı ve parola verilmiş kullanıcıların sisteme girmek için yaptıkları denemeler kayıt edilmiş ve hata oranları tespit edilmiştir. Deneysel çalışmada ,”Aykut” kullanıcısının.Kullanıcı Adı ve Parolası değişik insanlara verilmiş ve sisteme girmeleri istenmiştir.

4.1. Referans Yazma Ritmi Parametreleri

- Kullanıcı 1,2,3,...,11 sadece 1 kez sisteme girdikten sonra denemeyi bırakmışlardır.
- Kullanıcı 12 ‘e ise ilk girişinden sonra tekrar devam etmesi söylenmiş. 159. Denemesinde sisteme tekrar girebilmiştir.(ilk girişi 112.denemesinde)

Kullanıcı Adı: Aykut

Parola:19511951

ξ Karar Fonksiyonu: 0,9813774350

Tablo 4.1. Aykut Kullanıcısına Ait Referans Değerler

B Parametreleri		G Parametreleri	
B(1):	1	G(1):	7
B(2):	1	G(2):	2
B(3):	2	G(3):	2
B(4):	1	G(4):	2
B(5):	1	G(5):	8
B(6):	1	G(6):	2
B(7):	1	G(7):	1
B(8):	1		

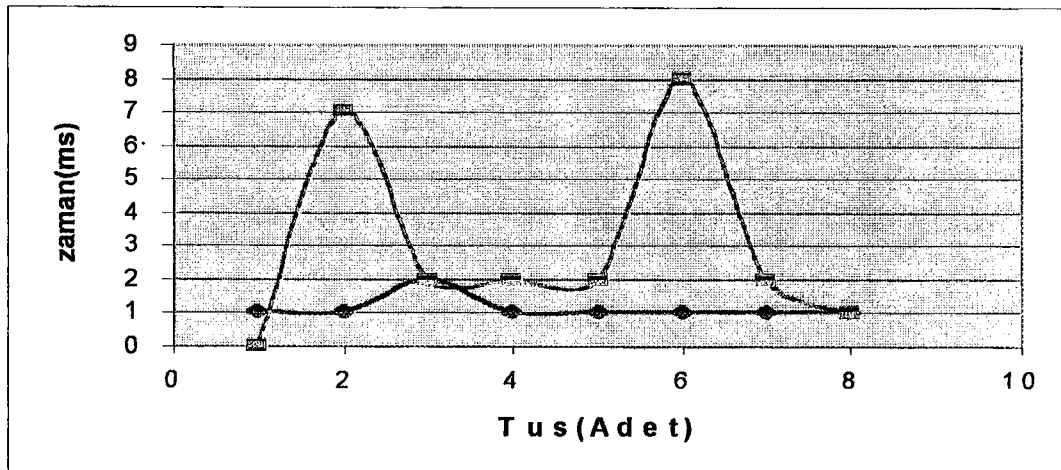
Tablo 4.2. Yetkisiz Kullanıcılara Ait Deneme Değerleri

Kullanıcı No:	Toplam Deneme Sayısı	Başarılı olma Sayısı
1	89	1
2	98	1
3	113	1
4	105	1
5	100	1
6	108	1
7	104	1
8	107	1
9	105	1
10	104	1
11	103	1
12	159	2

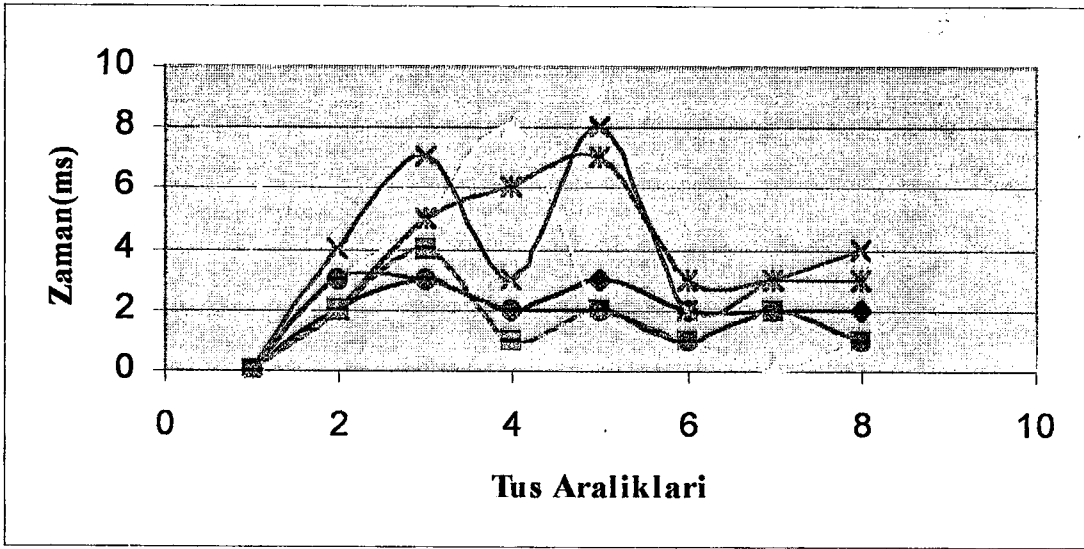
Bölüm 4.2 'de sistemin matematiksel formülasyonunda ifade edildiği gibi bazı parametreler sistem yöneticisi tarafından belirlenebiliyordu. Deneysel olarak yapılan çalışmada aşağıdaki parametre değerlerinin optimum değerler olduğu kanısına varılmıştır.

$$\Gamma_b = 0.1, \quad \Gamma_s = 0.9, \quad \Omega = 0.04$$

Yukarıdaki değerler değiştirilerek daha değişik sonuçlar alınabilir. Örneğin Ω değerinin değişmesi durumunda sistemin buna cevabı radikal bir şekilde farklılık gösterecektir. Şekil 4.5'de veritabanından alınan D ve G dizilerine ait olan değişim görülmektedir.

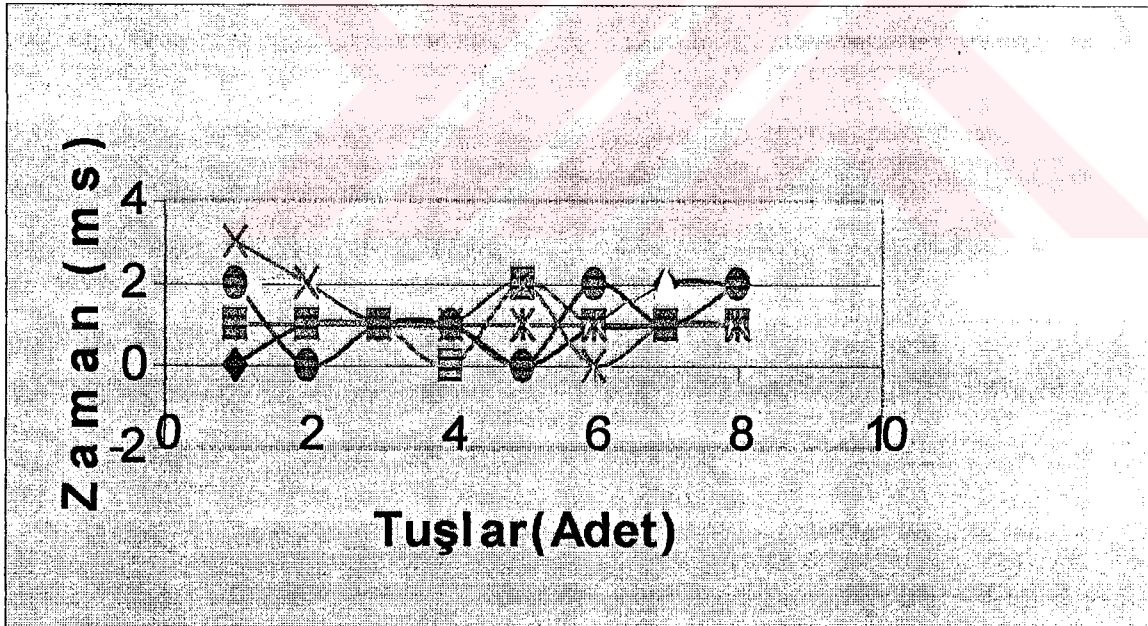


Şekil 4.1. Veritabanından Alınan Referans Değerlere Ait

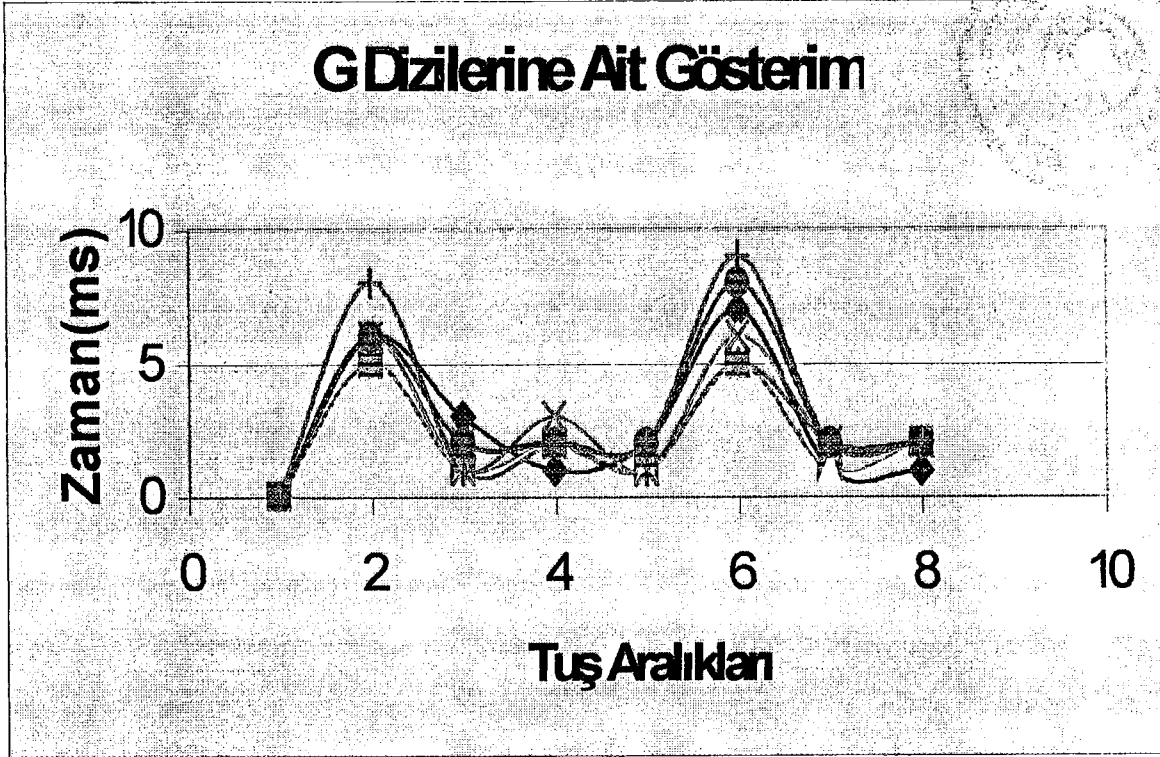
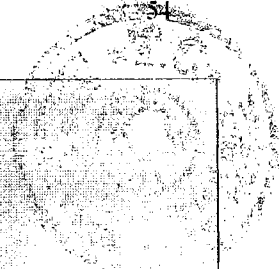


Şekil 4.2.G Dizilerine Ait Grafik Gösterim. (Yetkisiz Kullanıcılara Ait.)

Şekil 4.3 'den görüldüğü üzere B dizilerine bağlı grafik çok keskin değişimler göstermektedir. Bu sebeple B dizisinden türetilen vektörlerin direkt olarak karar fonksiyonunu etkilemesi sistemi kararsız hale getirebilir. Bu nedenle daha önce ifade edildiği gibi Γ , ağırlık çarpanı 0.1 olarak kabul edilmiştir.



Şekil 4.3- B Dizilerine Ait Grafik Gösterim. (Yetkisiz Kullanıcılara Ait.)



Şekil 4.4. G Dizilerine Ait Grafik Gösterim.(Yetkili Kullanıcılara Ait.)

Şekil 4.4’de sisteme yetkili bir kişinin kendi haklarını kullanarak, girdiğinde parametrelerin aldığı değerlerin değişimi görülmektedir. Kullanıcı sisteme 7 kez üst üste girdiğinde parametreleri küçük salınımlar yaparak değişmektedir.

Şekil 4.4’de ki G dizisine ait grafik açık bir şekilde ifade ediyor ki kullanıcının girdiği 7 adet kabul edilmiş girişin grafikleri birbirlerine oldukça benzerdir. Bu da dizilerin birbirleriyle benzer oldukları manasına gelmektedir. G dizilerinden elde edilen fonksiyonların karar fonksiyonuna etkisi B dizilerinden daha fazladır. Bu G dizilerinin daha kararlı olmasından kaynaklanmaktadır. Bu sebeplerden dolayı deneyde $\Gamma_s = 0.9$ olarak kabul edilmiştir. Bu bulgulardan Karar Fonksiyonunun referans değeri etrafındaki yakın salınımlar içerisindeki değerler sistem için anlamlıdır denilebilir.

Sistemin hata oranının tespitini hesaplamak istersek, Tablo 4.2’deki verileri kullanabiliriz. Genel manasıyla sistemin ortalama hata oranı tüm kullanıcıların aritmetik ortalamasını bularak hesaplanabilir.

σ Hata oranının göstermek üzere;

$$\sigma = \{\sum_{n=1}^N [\text{Giriş Sayısı}]\} / N \quad 4.1$$

Formül 4.1 'in sonucu sisteme ait genel bir hata oranı verecektir. Bu hata oranı ortalama olarak bir kullanıcının sistem güvenliğini kırması için kaç deneme yapması gerektiğine dair bir sayıdır.

Bu formülü kullanarak sistemin güvenlik hesaplamaları şöyle yapılabilir.

Tablo 4.2'deki verileri kullanarak ;

$$\sigma = \{\sum_{n=1}^N [\text{Giriş Sayısı}]\} / N$$

$$\sigma = \sum_{n=1}^{12} [89+98+113+105+100+108+104+107+105+104+103+112] / 12$$

$$\sigma = 104$$

104 sayısı ortalama olarak sistemin kırılması için yapılması gereken deneme sayısıdır. Fakat bölüm 3'de bahsedildiği üzere kullanıcıların deneme yapma sayıları sistemce kontrol edilmekteydi. Yani kullanıcı hakkı olmayan birisi sisteme girmek istediğinde 3. veya 5. Denemesinde (Bu opsiyonel bir sayıdır.Admin hakkı olan birisi tarafından değiştirilebilir.) sistem tarafından hakları kilitlenecek ve istemci bilgisayarı kullanamayacaktır. Dolayısıyla test amacı hariç hiç kimsenin sistem üzerinde 100 deneme yapma hakkı yoktur. Dolayısıyla sistem oldukça yüksek güvenliğe sahiptir.

5.SONUÇ

Biometrik sistemler her geçen gün popülerliği artan ve üzerinde araştırma geliştirme yapılan konulardan biridir. Gerçekten de günümüz dünyasında bilgisayarların hayatımıza girmediği yer kalmamış gibidir. Bilgisayarların bu kadar çok kullanılması güvenliklerinin sağlanması hususunda büyük sorumluluklar ve sorunlar ortaya çıkartmaktadır. Veri güvenliği gibi bir konunun kurumlar açısından önemi tartışılmaz.

Kuşkusuz bu çalışmanın konusu olan YARDS sisteminin kullanma maliyetinin düşüklüğü ve güvenilirliğinin yüksek olması nedeniyle biometrik sistemler arasında doğal bir seçenek olduğu görünmektedir. Yazma ritmi her insana has bir özellik taşıdığından bir tür biyolojik imza olarak düşünülebilir.[11] Yazma ritmi sistemlerinin daha iyi algoritmalar kullanılarak geliştirilmesi ve yazma ritmi tabanlı teknolojilere ağırlık verilmesi gerçekten de güvenli bilgisayar sistemleri oluşturulması açısından iyi olacaktır.

YARDS bir veritabanına ihtiyaç duymaktadır. Gerekli gördüğü yazma ritmi parametrelerini veritabanından sağlayarak güvenlik onayı vermektedir. Günümüzde güvenlik sistemlerinin tasarlanmasında bir çok taşıyıcı ortam kullanılabilmektedir. Bu ortamların en önemlisi ve en çok kullanılanları manyetik kartlar ve smart kartlardır. Kullanıcıların kodlanmış olan yazma ritimleri bu tür kartlar üzerinde saklanarak korunabilir. Böylece tamamen farklı bir fiziksel yapıya sahip bir güvenlik sistemi oluşturulabilir.

Genellikle biometrik güvenlik sistemlerinin güvenilirliğini yükseltmek için gerekli olan sağlam bir algoritma üzerine kurulmuş iyi bir programa sahip olmaktır. YARDS daha da geliştirilip mevcut sistemler üzerinde uygulanabilir. Örneğin bir işletim sistemi olarak MS NT® içerisine bu kodu gömmek ortaya etkili bir koruma sistemi çıkaracaktır. Bizim de bu aşamadan sonra yapmamız gereken programın parametrik hassasiyetini "Thread"ler kullanarak arttırmak ve elde edilen programı başta MS NT® ve diğer işletim sistemleriyle bütünleştirerek kullanımını arttırmaktır.

KAYNAKLAR

- [1] Rick JOYCE and Gopal GUPTA. "Identity Authorization Based on Keystroke Latencies". Communications of the ACM, 168-176, February 1990.
- [2] Ken PHILIPS, "Biometrics Identification looms on landscape of network logins.", PC Week March, 1997.
- [3] Susan RYCE, "Totalitarian Technology", <http://www.newdawnmagazine.com.au/50a.htm>
- [4] Thomas RUGGLES, "Comparison of Biometric techniques", <http://biometric-consulting.com/bio.htm>
- [5] Ken PHILIPS, "Moving to biometric standards.", <http://www.zdnet.com/pcweek/reviews/1027/27stand.html>, 1997.
- [6] Aykut Güven, İbrahim Soğukpınar "Yazma ritmi Yöntemiyle Bilgisayar ve Ağ Güvenlik Sistemi Tasarımı", BAS2000, Ankara, Haz. 2000
- [7] Fabian MONROSE, "Biometrics for Automatic Identity verification", New York State University TR1998-772.
- [8] Aykut Güven, İbrahim Soğukpınar "Kullanıcı Yetkilendirilmesinde Yazma ritmi Yöntemiyle Öğrenen Tanıma Sistemi Tasarımı", TAINN2000, İzmir, Haz. 2000
- [9] Saleh BLEHA, Charles SLIVINSKY, Bassam HUSSEIN. "Computer-Access Security Systems Using Keystroke Dynamics". IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL 12, NO. 12
- [10] Saleh BLEHA, "Recognition systems based on keystroke dynamics", PH.D. dissertation, Univ. Missouri-Columbia, May 1988

[11] Fabian MONROSE and Aviel RUBIN. “Authentication via Keystroke Dynamics.” <http://www.acm.org/pubs/articles/proceedings/commsec/266420/p48-monrose/p48-monrose.pdf>

[12] Microsoft Corp, Microsoft Developer Network



ÖZGEÇMİŞ

Aykut Güven ,1974 yılında İstanbul’da doğdu.İlk ve Orta öğrenimini İstanbul’da tamamladı.1991 yılında Haydarpaşa Teknik Lisesi Elektrik bölümünden mezun oldu. 1992 yılında İstanbul Teknik Üniversitesi Fizik Mühendisliği bölümünde eğitimine devam etti. 1997 yılında katıhal elektroniği konusunda eğitimini tamamlayarak mezun oldu. 1998 yılında Gebze Yük.Tek.Enstitüsü Bilgisayar Mühendisliği bölümüne öğrenci olarak girdi. Halen Doğu Üniversitesi Bilgisayar Mühendisliği Bölümünde Uzman olarak çalışmaktadır.



EK

Yazma Ritmi Programına Ait Kaynak Kodlar

```

Dim AConn As New ADODB.Connection //Değişken tanımlamaları yapılıyor.
Dim AComm As New ADODB.Command
Dim ARec As New ADODB.Recordset
Dim GRec As New ADODB.Recordset
Dim DataL1 As New ADODB.Recordset
Dim z, m, f, k As Long
Dim s(8)
Dim g(8)
Dim Flag, SetupFlag As Boolean
Dim i As Integer
Dim UserName, NewUser, Password As String
Dim EPassword As String
Dim EPressTime, FPressTime As Integer
Dim EDelayTime, FDelayTime As Integer
Dim ViolationConstant As Integer
Dim EAngle As Double
Dim EPFunction As Double
Dim PressTime As Integer
Dim DelayTime As Integer
Dim Angle As Double
Dim ViolationControl As Integer
Dim PFunction As Double
Dim SecurityLayer As Double
Dim SecurityLayerPass As Double
Dim SecurityLayerPass1 As Double
Dim SecurityLayerPass2 As Double
Dim CARPAN As Double

```

```

Dim ES1, ES2, ES3, ES4, ES5, ES6, ES7, ES8 As Double
Dim EG2, EG3, EG4, EG5, EG6, EG7, EG8 As Double
'Constants that related with Learning step of the program
Dim LearnMeFlag As Boolean
Dim intTw As Integer // Değişken tanımlamaları sonu.

```

```

Private Sub DBCombo1_DblClick(Area As Integer)
AD = DBCombo1.Text
Access.RecordSource = "select * from Security where UserName='" & AD &'"
Access.Refresh
MsgBox Access.Recordset.RecordCount
Access.Recordset.Edit
Access.Recordset.Fields("LockAccount") = "True"
Access.Recordset.Update
Access.RecordSource = "select * from security where LockAccount=False"
Access.Refresh
End Sub

```

```

Private Sub DBList1_Click()
On Error GoTo err
Data1.RecordSource = "select * from security where UserName='" &
DBList1.Text & "'"
Data1.Refresh
Data1.Recordset.Edit
Data1.Recordset.Fields("LockAccount") = "FALSE"
Data1.Recordset.Update
Data1.Refresh
Data1.RecordSource = "Select * from security where LockAccount=TRUE"
Data1.Refresh
Exit Sub
err: MsgBox err.Description
End Sub

```

```

Private Sub DBList1_LostFocus()

```

DBList1.Visible = False

End Sub

Private Sub DataList1_DblClick()

On Error GoTo err

Dim LRec As New ADODB.Recordset

LRec.Open "select * from security where UserName='" & DataList1.Text &
 "'", AConn, adOpenKeyset, adLockOptimistic, adCmdCommand

LRec.Fields("LockAccount") = "FALSE"

LRec.Update

LRec.Close

LRec.Open "Select * from security where LockAccount=TRUE", AConn,
 adOpenKeyset, adLockOptimistic, adCmdCommand

Set DataList1.RowSource = LRec

DataList1.ListField = "UserName"

Exit Sub

err: MsgBox err.Description

End Sub

Private Sub DataList1_LostFocus()

DataList1.Visible = False

End Sub

Private Sub exit_Click()

Form1.security.Visible = False

End Sub

Private Sub Form_Click()

On Error GoTo err

Form1.Icon = LoadPicture(App.Path + "\key04.ico")

Text1.Enabled = True

Label3.Visible = False

Text1.Text = ""

Text2.Text = ""

Text1.Enabled = True

Text2.SetFocus

z = 0

m = 0

f = 0

i = 0

SetupFlag = False

Exit Sub

err: MsgBox err.discription

End Sub

Private Sub Form_Load()// Veritabanına ulaşılıyor.

On Error GoTo err

Dim denek As Long

AConn.Open ("DSN=BIORYTM;User=BIORYTM;PWD=bio;")

 ARec.CursorType = adOpenKeyset

 ARec.LockType = adLockOptimistic

 ARec.Open "Select * from security", AConn, , , adCmdCommand

LearnMeFlag = False

intTw = 0

ViolationControl = 5

z = 0

m = 0

f = 0

i = 0

CARPAN = 0.05

Form1.security.Visible = False

Flag = True

FPressTime = 2

FDelayTime = 4

Exit Sub

err: MsgBox err.Description

End Sub

Private Sub Form_Unload(Cancel As Integer)

```

AConn.Close
DataL1.Close
ARec.Close
GRec.Close
End Sub

Private Sub HR_Click()
FPressTime = 2
FDelayTime = 2
CARPAN = 0.03
ViolationConstant = 5
Form1.security.Visible = False
HR.Checked = True
LR.Checked = False
MR.Checked = False
NS.Checked = False
Text2.Text = ""
Text2.SetFocus
End Sub

Private Sub LR_Click()// Program hassasiyetiyle ilgili kodlar.
FPressTime = 4
FDelayTime = 8
CARPAN = 0.08
ViolationConstant = 20
Form1.security.Visible = False
LR.Checked = True
MR.Checked = False
HR.Checked = False
NS.Checked = False
Text2.Text = ""
Text2.SetFocus
End Sub

Private Sub MR_Click()
FPressTime = 3

```

```

FDelayTime = 4
CARPAN = 0.04
ViolationConstant = 10
Form1.security.Visible = False
MR.Checked = True
HR.Checked = False
LR.Checked = False
NS.Checked = False
Text2.Text = ""
Text2.SetFocus
End Sub

```

```
Private Sub NoLimit_Click()
```

```

ViolationControl = 150
ViolationConstant = 150
S5.Checked = False
S10.Checked = False
S15.Checked = False
NoLimit.Checked = True
End Sub

```

```
Private Sub NS_Click()
```

```

vbsonuc = MsgBox("This action may cause to damage your system.Are you
sure to do this!", vbOKCancel)

```

```
If vbsonuc = vbOK Then
```

```

CARPAN = 10
NS.Checked = True
MR.Checked = False
LR.Checked = False
HR.Checked = False

```

```
Else
```

```

Text2.Text = ""
Text2.SetFocus
Exit Sub

```

```
End If
```

```

Form1.security.Visible = False
Text2.Text = ""
Text2.SetFocus
End Sub

Private Sub S10_Click()
ViolationControl = 10
ViolationConstant = 10
S5.Checked = False
S10.Checked = True
S15.Checked = False
NoLimit.Checked = False
End Sub

Private Sub S15_Click()
ViolationControl = 15
ViolationConstant = 15
S5.Checked = False
S10.Checked = False
S15.Checked = True
NoLimit.Checked = False
End Sub

Private Sub S5_Click()
ViolationControl = 5
ViolationConstant = 5
S5.Checked = True
S10.Checked = False
S15.Checked = False
NoLimit.Checked = False
End Sub

Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
i = i + 1
f = 0
Timer1.Enabled = True
Timer2.Enabled = False
End Sub

```

```

Private Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
On Error GoTo err
Timer1.Enabled = False
Timer2.Enabled = True
s(i) = z
g(i) = k
If SetupFlag = False Then
If Len(Text1.Text) = 8 Then
Text1.Enabled = False
Password = Text1.Text
BioAccess
End If
Else
If Len(Text1.Text) = 8 Then
Text1.Text = ""
i = 0
BioAccess
End If
End If
z = 0
Exit Sub
err: MsgBox err.Description
End Sub

Private Sub Text2_GotFocus()
Text2.Text = ""
End Sub

Private Sub Text2_LostFocus()
On Error GoTo err
Dim LRec As New ADODB.Recordset
Dim GecRec As New ADODB.Recordset
Dim LComm As New ADODB.Command
// Sisteme yapılan girişler bu kodlarla kontrol edilirler.
Dim kayit As Long

```

```

LRec.Open "Select * from Security where UserName='" & Text2.Text & "'",
AConn, adOpenKeyset, adLockOptimistic, adCmdCommand
GecRec.Open "Select * from LearnUser where UserName='" & Text2.Text &
"", AConn, adOpenKeyset, adLockOptimistic, adCmdCommand
UserName = Text2.Text
If UserName = "LAYER" Then
Form1.security.Visible = True
Exit Sub
End If
If UserName <> "setup" Then
SecurityLayerPass = LRec.Fields("PFunction")
SecurityLayerPass1 = SecurityLayerPass + CARPAN * SecurityLayerPass
SecurityLayerPass2 = SecurityLayerPass - CARPAN * SecurityLayerPass
If LRec.Fields("LockAccount") = True Then
MsgBox ("User" & " " & UserName & " " & " was locked by System. Contact
with your System Administrator.")
Label3.Visible = True
Label3.Caption = "Locked by System...."
Text1.Enabled = False
End If
If LRec.RecordCount = 0 Then
MsgBox ("You are not a valid User who has permissions for this system..")
Text2.SetFocus
End If
Else
Form1.Caption = " Knowing the User" & " " & NewUser
LearnMeFlag = True
SetupFlag = True
NewUser = InputBox("Enter New User name")
Frame1.Visible = True
GecRec.AddNew
GecRec.Fields("UserName") = NewUser
GecRec.Update
Text1.SetFocus

```

MsgBox (" User is Added to Database.Please type your 'BioPassword' carefully")

End If

LRec.Close

GecRec.Close

Exit Sub

err: MsgBox err.Description

End Sub

Private Sub Timer1_Timer()

On Error GoTo err

z = z + Timer1.Interval

Exit Sub

err: MsgBox err.Description

End Sub

Private Sub Timer2_Timer()

On err GoTo err

f = f + Timer2.Interval

k = f

Exit Sub

err: MsgBox err.Description

End Sub

Private Sub BioAccess()

On Error GoTo err

Dim LRec As New ADODB.Recordset

Dim n, Teta As Double

Label5.Visible = True

LRec.Open "Select * from Security where UserName='" & UserName & "'",

AConn, adOpenKeyset, adLockOptimistic, adCmdCommand

Set ARec = LRec

PressTime = s(1) + s(2) + s(3) + s(4) + s(5) + s(6) + s(7) + s(8)

DelayTime = g(2) + g(3) + g(4) + g(5) + g(6) + g(7) + g(8)

If SetupFlag = True Then

PFunction = 1

InsertParameter

Exit Sub

End If

EPassword = LRec.Fields("Password")

If Password = EPassword Then

BioControl

Else

MsgBox ("Password is not valid for this system....")

End If

Exit Sub

err: MsgBox err.Description

End Sub

Private Sub BioControl()

On Error GoTo err

Dim Angled As Double

If UserName <> "setup" Then

ES1 = ARec.Fields("s1")// ReferansYazma Ritmi parametreleri veritabanından

ES2 = ARec.Fields("s2")// okunuyorlar.

ES3 = ARec.Fields("s3")

ES4 = ARec.Fields("s4")

ES5 = ARec.Fields("s5")

ES6 = ARec.Fields("s6")

ES7 = ARec.Fields("s7")

ES8 = ARec.Fields("s8")

EG2 = ARec.Fields("g2")

EG3 = ARec.Fields("g3")

EG4 = ARec.Fields("g4")

EG5 = ARec.Fields("g5")

EG6 = ARec.Fields("g6")

EG7 = ARec.Fields("g7")

EG8 = ARec.Fields("g8")

EPressTime = ARec.Fields("PressTime")

EDelayTime = ARec.Fields("DelayTime")

ParameterControl

If Flag = True Then

$$\text{AngleP} = (s(1) * ES1 + s(2) * ES2 + s(3) * ES3 + s(4) * ES4 + s(5) * ES5 + s(6) * ES6 + s(7) * ES7 + s(8) * ES8) / (\text{Sqr}(s(1)^2 + s(2)^2 + s(3)^2 + s(4)^2 + s(5)^2 + s(6)^2 + s(7)^2 + s(8)^2) * (\text{Sqr}(ES1^2 + ES2^2 + ES3^2 + ES4^2 + ES5^2 + ES6^2 + ES7^2 + ES8^2)))$$

$$\text{Angled} = (g(2) * EG2 + g(3) * EG3 + g(4) * EG4 + g(5) * EG5 + g(6) * EG6 + g(7) * EG7 + g(8) * EG8) / (\text{Sqr}(g(2)^2 + g(3)^2 + g(4)^2 + g(5)^2 + g(6)^2 + g(7)^2 + g(8)^2) * (\text{Sqr}(EG2^2 + EG3^2 + EG4^2 + EG5^2 + EG6^2 + EG7^2 + EG8^2)))$$

// Vektörel işlemler yapılıyor.

PFunction = 0.9 * Angled + 0.1 * AngleP

End If

If PFunction < SecurityLayerPass2 Or PFunction > SecurityLayerPass1 Then//

Karar aşaması

AccessControl

Text1.Text = ""

Text2.Text = ""

Text2.SetFocus

Text1.Enabled = True

i = 0

Else

Form1.Icon = LoadPicture(App.Path + "\handshak.ico")

Timer3.Enabled = False

Label3.Visible = True

Label3.Caption = "USER ACCEPTED....."

MsgBox ("User" & " " & UserName & " " & "have a nice day....")

ViolationControl = ViolationConstant

UpdateParameter

End If

End If

Exit Sub

err: MsgBox err.Description

End Sub

Private Sub InsertParameter()

Dim LRec As New ADODB.Recordset

```

LRec.CursorType = adOpenKeyset
LRec.LockType = adLockOptimistic
LRec.Open "Select * from LearnUser where UserName='" & NewUser &
""", AConn, , , adCmdCommand
If LRec.RecordCount <> 0 Then
LRec.AddNew
LRec.Fields("s1") = s(1)// Yeni parametreler veritabanına güncelleniyor.
LRec.Fields("s2") = s(2)
LRec.Fields("s3") = s(3)
LRec.Fields("s4") = s(4)
LRec.Fields("s5") = s(5)
LRec.Fields("s6") = s(6)
LRec.Fields("s7") = s(7)
LRec.Fields("s8") = s(8)
LRec.Fields("g2") = g(2)
LRec.Fields("g3") = g(3)
LRec.Fields("g4") = g(4)
LRec.Fields("g5") = g(5)
LRec.Fields("g6") = g(6)
LRec.Fields("g7") = g(7)
LRec.Fields("g8") = g(8)
LRec.Fields("UserName") = NewUser
LRec.Fields("Password") = Password
LRec.Fields("PressTime") = PressTime
LRec.Fields("DelayTime") = DelayTime
LRec.Fields("PFunction") = PFunction
LRec.Update
LRec.Close
Text1.SetFocus
Else
LRec.Close
Exit Sub
End If
End Sub

```

Private Sub UpdateParameter()

ARec.Fields("s1") = s(1)

ARec.Fields("s2") = s(2)

ARec.Fields("s3") = s(3)

ARec.Fields("s4") = s(4)

ARec.Fields("s5") = s(5)

ARec.Fields("s6") = s(6)

ARec.Fields("s7") = s(7)

ARec.Fields("s8") = s(8)

ARec.Fields("g2") = g(2)

ARec.Fields("g3") = g(3)

ARec.Fields("g4") = g(4)

ARec.Fields("g5") = g(5)

ARec.Fields("g6") = g(6)

ARec.Fields("g7") = g(7)

ARec.Fields("g8") = g(8)

ARec.Fields("PressTime") = PressTime

ARec.Fields("DelayTime") = DelayTime

ARec.Fields("PFunction") = PFunction

ARec.Update

End Sub

Private Sub ParameterControl()

Dim ES(8)

Dim EG(8)

For n = 2 To 8

be = "g" & n

EG(n) = ARec.Fields(be)

Next n

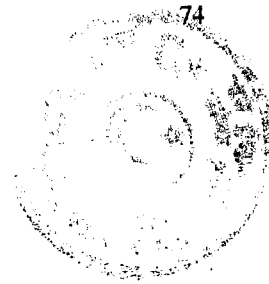
For m = 1 To 8

a = "s" & m

ES(m) = ARec.Fields(a)

Next m

Flag = True



```

For i = 1 To 8
FarkS = Abs(s(i) - ES(i))
    If FarkS > FPressTime Then
        Flag = False
        PFunction = 0
        Exit Sub
    End If
Next i
For i = 2 To 8
FarkG = Abs(g(i) - EG(i))
    If FarkG > FDelayTime Then
        Flag = False
        PFunction = 0
        Exit Sub
    End If
Next i
End Sub
Private Sub AccessControl()
ViolationControl = ViolationControl - 1
Timer3.Enabled = True
End Sub
Private Sub Timer3_Timer()
    If ViolationControl = 0 Then
        MsgBox ("User" & " " & UserName & "'s Account was locked.Please contact
with System Administrator.")
        ARec.Fields("LockAccount") = True
        ARec.Update
        Timer3.Enabled = False
    End If
    Form1.Label3.Caption = "ACCESS IS UNSUCCESSFUL"
    If Form1.Label3.Visible = False Then
        Form1.Label3.Visible = True
    Else
        Form1.Label3.Visible = False
    End If

```

End If

End Sub

Private Sub LockAccount()

ARec.Fields("LockAccount") = "True"

ARec.Update

End Sub

Private Sub Timer4_Timer()

intTw = intTw + 50

If intTw < 1000 Then

If Labela.Visible = True Then

Labela.Visible = False

Labelb.Visible = False

Else

Labela.Visible = True

Labelb.Visible = True

End If

Else

Frame1.Visible = False

Timer4.Enabled = False

End If

End Sub

Private Sub UnlockUsers_Click()

DataL1.Open "Select * from Security where LockAccount= True ", AConn,
adOpenKeyset, adLockOptimistic, adCmdCommand

DataList1.Visible = True

Set DataList1.RowSource = DataL1

DataList1.ListField = "UserName"

DataList1.Refresh

End Sub