İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

AKILLI KARTLAR İLE AĞ GÜVENLİK SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

Müh. Hakan Uluöz

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

MAYIS 2003

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

AKILLI KARTLAR İLE AĞ GÜVENLİK SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

Müh. Hakan Uluöz
504991035

Tezin Enstitüye Verildiği Tarih : 05 Mayıs 2003
Tezin Savunulduğu Tarih : 29 Mayıs 2003

Tez Danışmanı : **Doç.Dr. Coşkun SÖNMEZ**

Diğer Jüri Üyeleri **Prof.Dr. Bülent ÖRENCİK**

Yrd.Doç.Dr. Gökhan YAVUZ

MAYIS 2003

ÖNSÖZ

Tez çalışması sırasında, desteğini sağlayan, birikimi ile tezin yönlendirilmesini sağlayan, ciddi rahatsızlığına rağmen ilgisini eksiltmeyen değerli danışman öğretmenim Sayın Doç. Dr. Coşkun Sönmez'e teşekkürlerimi ve saygılarımı sunarım.

Tez çalışması esnasında maddi ve manevi destek sağlayan tüm DOS Bilgisayar A.Ş. çalışanlarına, özellikle projede referans alınan akıllı kart ile ilgili bana çok yardımcı olan sayın Müh. Murat Karagöz'e teşekkürlerimi sunarım.

Tezin tasarlanması ve kaleme alınmasında sağladığı destek, yardım, moral ve lisans eğitiminden beri süregelen dostluğundan dolayı Sayın Y. Müh. İbrahim Niyazi Ülgür'e teşekkür ve sevgilerimi sunarım.

Ulaştığım noktada olmamı sağlayan ilköğretimden yüksek lisans eğitimime kadar bana emeği geçen tüm eğitmenlerime teşekkür ve saygılarımı sunarım.

Bu tezin hazırlanmasında edindiğim bilgileri bilim dünyasına kazandıran tüm akademisyen, yazılımcı, yazar, araştırmacılara, yayınların elimize ulaşmasını sağlayan tüm yayıncı, internet sitesi çalışanları, denetçilere, hayatın anlamı olan tüm arkadaş ve dostlarıma teşekkürlerimi sunar başarılarının devamını dilerim...

MAYIS 2003 Hakan ULUÖZ

İÇİNDEKİLER

Sayfa No

KISALTMALAR ..... **v**
TABLO LİSTESİ ..... **vi**
ŞEKİL LİSTESİ ..... **vii**
ÖZET ..... **viii**
SUMMARY ..... **x**

1. GİRİŞ **1**

2. AKILLI KARTLAR **4**
2.1. Donanım 4
2.1.1. Temel bileşenler 4
2.1.2. Haberleşme protokollerine göre akıllı kartlar 5
2.1.3. İletişim arabirimlerine göre akıllı kartlar 6
2.1.4. Akıllı kartlar üzerinde donanımsal güvenlik 7
2.2. Yazılım 7
2.2.1. İşletim sistemi 8
2.2.2. Dosya sistemi 9

3. VERİ ŞİFRELEME – DOĞRULAMA METODLARI **11**
3.1. DES (Data Encryption Standart) Yöntemi 11
3.2. RSA (Rivest, Shamir and Adleman Algorithm) Yöntemi 14
3.3. Veri Şifreleme Algoritmalarına Genel Bir Bakış 16

4. İŞLETİM SİSTEMLERİ – GÜVENLİK ÖNLEMLERİ ve AÇIKLARI **19**
4.1. Kullanıcı Koruması 19
4.2. Paylaşım ve Ağ Koruması 21
4.3. Güvenlik Açıkları 22

5. TASARIM **24**
5.1. Akıllı Kart 24
5.1.1. Akıllı kart seçimi 24
5.1.1.1. Şifreleme-güvenlik yetenekleri 24
5.1.1.2. Akıllı kart güvenliği ve esnekliği 25
5.1.1.3. Akıllı kart veri saklama kabiliyeti 26
5.1.1.4. Akıllı kart iletişim tipi 26
5.1.2. Akıllı kart dosya yapısı 27

5.1.2.1. Çoklu uygulama desteği ve akıllı kart uygulama tanıtım metodu 27
5.1.2.2. Uygulama dosya yapısı 30
5.1.2.3. Dosya erişim hakları ve anahtar küme dosyaları yapısı..... 32
5.1.3. Okuyucu seçimi 33
5.2. Sunucu Güvenlik Sistemi (HSM – Host Security Module) 34
5.2.1. HSM akıllı kartı 35
5.2.2. HSM iletişim donanım ve yazılımı 35
5.3. Uç Birim Güvenlik Modülü (TSM – Terminal Security Module).....36
5.4. Sistemin Çalışma Şekli 38
5.4.1. Ağa bağlantı 38
5.4.1.1. Akıllı karttan kullanıcı bilgilerinin alınması..... 38
5.4.1.2. Ağa bağlantı sağlanması 39
5.4.2. Uç birimler arası haberleşme 40
5.4.2.1. Karşı uç birimin açık olması 40
5.4.2.2. Karşı uç birimin kilitli olması 41
5.4.3. Uç birim dosyalarının güvenliği 43
5.5. Ek Özellikler 44
5.5.1. Kişisel bilgiler uygulaması 44
5.5.2. E-Cüzdan uygulaması 44
5.5.3. Geçiş güvenlik uygulaması 45

6. GERÇEKLEME 46
6.1. Türetilmiş Anahtarların Oluşturulması 46
6.1.1. Akıllı kart yardımı ile türetilmiş anahtar oluşturma 46
6.1.2. TSM'in geçici oturum anahtarı üretmesi 47
6.2. Paket Yapısı 48
6.2.1. Bağlantı isteği ve cevabı 49
6.2.2. Kontrol paketleri 50
6.3. Uç Birimin Açılması, Ağa Bağlanma ve Haberleşme 51
6.3.1. TSM modülü 51
6.3.2. HSM modülü..... 56

7. SONUÇLAR 57

8. TARTIŞMA 59

KAYNAKLAR 62

ÖZGEÇMİŞ.....64

KISALTMALAR

USB : Universal Serial Bus (Evrensel Seri Arabirim)
PIN : Personal Identification Number (Kişisel Tanıtım Numarası)
EMV : Europay, MasterCard, Visa
POS : Point Of Sale - Point of Service (Satış Noktası - Servis Noktası)
SSL : Secure Sockets Layer (Güvenli Soket Katmanı)
KB : Kilobyte
ATR : Answer To Reset (Reset Cevabı)
RF : Radio Frequency (Radyo Frekans)
GSM : Global System For Mobile Telecomunications (Mobil İletişim Global Sistemi)
RSA : Rivest, Shamir and Adleman Algorithm
DES : Data Encryption Standart (Veri Şifreleme Standardı)
3DES : Tripple Data Encryption Standart (Üçlü Veri Şifreleme Standardı)
ISO : International Standards Organization (Uluslararası Standartlar Organizasyonu)
MF : Master File – Mandatory File (Temel Dosya – Zorunlu Dosya)
AID : Application Identifier (Uygulama Tanıtıcısı)
DF : Dedicated File – Directory File (Klasör – Dizin Dosyası)
ECB : Electronic Code Book (Elektronik Kod Kitabı)
CBC : Chained Block Cipher (Zincireleme Blok Kodlayıcı)
CFM : Cipher Feedback Mode (Kodlama Geribesleme Modu)
IP : Interner Protocol (İnternet Protokolü)
PC/SC : Personal Computer/ SmartCard (Kişisel Bilgisayar / Akıllı kart)
TCP : Transmission Control Protocol (İletişim Kontrol Protokolü)
API : Application Programming Interfaces (Uygulama Programlama Arayüzü)
PUK : PIN Unblock Code (PIN Açıcı Kod)

TABLO LİSTESİ

Sayfa No

Tablo 3.1. RSA için anahtar uzunluğuna göre anahtarın kırılma zamanı 15
Tablo 5.1. AID kayıtlarındaki uygulama tipleri..... 28
Tablo 5.2. AID kayıtlarındaki dosya tipleri 29

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 Akıllı kart blok diagramı. 5
Şekil 2.2 AID oluşturma yöntemleri. 10
Şekil 3.1 DES algoritmasının temel çalışma mantığı 12
Şekil 3.2 Sırası ile DES ECB ve CBC Modları 13
Şekil 3.3 DES CFM modu. 14
Şekil 3.4 Mesaj doğrulama protokolü blok diagramı 17
Şekil 5.1 Kullanılacak AID yapısı 29
Şekil 5.2 Akıllı kartta oluşturulacak dosya yapısı. 31
Şekil 5.3 Bağlanma işlemi durum diagramı 39
Şekil 5.4 Uç birimlerden birinin kilitli olması halinde haberleşme. 42
Şekil 6.1 Her iki uç birimin açık olması halinde anahtar oluşturma mantığı. 47
Şekil 6.2 Uç birimlerden birinin kilitli olması haline anahtar oluşturma mantığı. .. 48
Şekil 6.3 Genel paket yapısı. 49
Şekil 6.4 Bağlanma isteği paketinin genel yapısı 50
Şekil 6.5 Uç birimlerden biri kilitli olduğunda bağlantı paketi 50
Şekil 6.6 Kontrol paketi genel yapısı 51

AKILLI KARTLAR İLE AĞ GÜVENLİK SİSTEMİ TASARIMI

ÖZET

Veri güvenliği konusu, bilgisayarların iş yaşamında insan ve dolayısı ile şirketlerin vazgeçilmez bir parçası haline gelmesi ile günümüzde önemi hayli artmış ve halen de konu üzerinde detaylı çalışmaların yapıldığı konulardan biridir. Artık bir ciddi bir kurum bilgisayar ve ağ altyapısını kurmaya karar verme aşamasındayken uygulama ve işletim sistemi seçiminde güvenlik en az performans, statik çalışma ve maliyet kadar göz önünde bulundurulmaktadır. Özellikle, internet ortamının artık her kurum tarafından yoğun olarak kullanılması, güvenlik konusundaki problemlerin kaynağını oluşturmaktadır.

Günümüzde, her modern sistemde olması gerektiği gibi, yeni nesil tüm işletim sistemleri belirli seviyelerde güvenlik önlemlerine sahip bulunmaktadır. Bunlar, bir kullanıcı adı ve şifresi gibi basit bir seviyeden uç birimler arasında yüksek güvenlikli şifreleme algoritmaları ile veri gizleme yöntemlerine kadar yüksek seviyelere çıkabilmektedir. Bu seviyelerin iki ucu belirgin bir şekilde farklı gibi görünse de temel eksiklikleri aynıdır: Eğer iletişimin sağlandığı herhangi bir uç birim istenmeyen bir kişi tarafından ele geçirilirse, sistemin güvenlik duvarları tamamen yıkılmış olacaktır. Dolayısı ile uç birimleri günümüzdeki gibi sadece bir kullanıcı adı ve şifresi ile korumak mümkün değildir. Zira bu bilgiler istenmeyen kişilerin eline rahatlıkla geçebileceği gibi, geliştirilmiş bir çok program bu adımı rahatlıkla geçebilmektedir. Örneğin bir Linux sistemine tanımlı herhangi bir kullanıcı ile bağlanıp, kullanıcıya yönetici (root) yetkileri verebilen, ya da Microsoft Windows NT sınıfı işletim sistemlerini yönetici (Administrator) yetkileri ile açabilen programların sayısı gün geçtikçe artmaktadır.

Konunun öneminin önemli ağlara yapılan saldırılar karşısında giderek daha fazla anlaşılması üzere piyasaya bu konuda değişik ürünler çıkmaya başlamıştır. Örnek vermek gerekirse, birçok firma Microsoft işletim sistemleri için bir akıllı kart vasıtası ile sisteme giriş yapılmasını sağlayan ürünler piyasaya çıkarmaktadır. Zira, çok yakın bir zamanda akıllı kart okuyucuya sahip klavyeler sıradan klavyelerin yerini almaya başlayacak görünümündedir. Diğer ilginç bir ürün ise, aynı yöntemi farklı bir çevresel birim kullanarak uygular: Kullanılan cihaz yeni nesil seri arabirimlere bağlanabilen bir anahtardır. Her iki örnekte de sağlanan kullanıcının bilgisayarına erişim hakkının kazanılmasına yöneliktir ve sadece kullanıcı kart ya da anahtarı üzerinde bir verinin doğrulanması şeklinde çok kaba bir yöntem kullanır.

Veri güvenliğinin tam olarak sağlanabilmesi için başlanması gerekli nokta, açık bir şekilde belli olmaktadır. İlk aşamada, bir kullanıcıya ya da belirli bir kullanıcı gurubuna ait bir bilgisayar sadece yetkili kişilerce açılabilmelidir, zira güvenliğin zayıf halkalarından biri bu noktadır. Bir üst aşamada bilgisayarın veri depolama için kullandığı cihazlardaki önemli bilgiler, belli bir algoritmaya göre şifrelenmiş bir şekilde saklanmalıdır. Son aşamasında ise ağdaki haberleşme, ağ performansının kabul edilebilir

seviyelerde tutulabilmesi için ancak gerekli olduğunda belirli bir şifreleme düzeni ile korunmuş olmalıdır. Bu üç temel adım, birbirleri ile bağlantılı bir şekilde çalıştırılabilir ise uç birimlerden başlanarak, ağın güvenliği sağlanmış olacaktır.

Bu tezin konusu içinde yer alan sistemde, güvenlik için akıllı kartlardan faydalanılacaktır. Bu seçimin nedeninin ayrıntıları ile tez içerisinde sunulacaktır. Tez içerisinde güvenlik için savunulan konu aslen "güvenlik için konulan tüm adımların insan kontrolünden çıkarılması"dır[1]. Kısacası, insan hatalarından kaynaklanan açıkların kapatılması hedeflenmektedir.

Akıllı kart kullanılan benzer uygulamalardan farklı olarak, tasarlanan sistemde bir akıllı kart üzerindeki veri doğrulama yöntemi değil, iletişimde bulunan tarafların karşılıklı olarak birbirlerini doğrulaması hedeflenmiştir. Bu yöntem ile hem uç birimler korunmuş olacağı gibi, ağ üzerindeki güvenli haberleşme de akıllı kartlar aracılığı ile dinamik oturum anahtarları yaratılması sağlanacak bu şekilde de kullanıcıların hiç bir müdahalesi ve bilgisi olmadan ağdaki veriler gizlenecektir.

Akıllı kartlar günümüzde sadece şifre doğrulayabilen hafıza birimleri olmaktan çıkarak insan hayatının aktif birer parçası olma yönünde hızla aşamalar kat etmektedirler. 2005 yılı itibarı ile kullanılan tüm kredi kartları artık akıllı kartlardan oluşacaktır. Bu kadar gelişkin bir cihazı sadece güvenlik için kullanmak, açıkçası yeteneklerinin büyük bir kısmını çöpe atmak demek olacaktır.

Bu tez içerinde savunulan temanın asıl üyelerinden biri de bir akıllı kart olduğuna göre, sistemi onun yeteneklerine göre tasarlamak daha doğru olacaktır. Bu nedenle kullanılacak kart sadece bir güvenlik aracı olmaktan öte, kişisel bilgileri, acil sağlık bilgilerini tutan bir kimlik, kapıları açabilen bir anahtar, ve finansal işlemler yapılabilmesini sağlayabilen güvenli bir cüzdan olarak kullanabilecektir[2,3]. Biraz hayal gücü zorlaması ile belki de on yıllık bir zaman zarfı sonrasındaki günlük insan hayatının bir izdüşümü tasvir edilmiş olacaktır. Ancak unutulmaması gereken tasarlanan sistemin bir ütopya değil fiziksel olarak şu andaki donanım ve yazılım koşulları ile gerçekleştirilebilir olmasıdır. Gerçekleştirilebilmesindeki tek ihtiyaç, bahsi geçen uygulamaların gereksinimleri olan donanım ve yazılımların, tasarlanan akıllı kartı desteklemesidir. Bu, belki de, EMV projesi ile başlayan akıllı kartlar üzerindeki uygulamaların standartlaşması adımının yıllar sonra alacağı hal ile sorun olmaktan çıkacaktır.

DESIGNING A NETWORK SECURITY SYSTEM WITH SMART CARDS

SUMMARY

Data Security is one of the popular topics still been worked carefully on today, as the computers become a very critical partner in human and so far, companies work progress. Nowadays, security concepts and performance is one of the main factors, as overall system performance, stabilty and of course cost, on selection process of the software and operating system when building a network. As the "internet" has been widely used by any type of companies, this leads the main problems of the data and network securtiy.

Today, as it is quite commonly expected every operating system has some security features coming built-in. This features starts from basic authentication methods as a username-password combination and may reach to higher levels as high encryption standarts between nodes. Though the ends of this scale seems so diffirent, they have the same weakness: If any of the node is captured, the total system's trustworthiness can easily be released. So, it is clearly seen that protecting nodes and also the server by means of a user-password combination is not fully meets the security complaints, hence recovering this pair is so easy for an eavesdropper by either a close observation, or by means of trojan programmes like "keycatcher" reporting the keys pressed to his owner. There are also some programmes that can pass administrative rigths to a simple user (even to guest users). The number of these type of "hacking" programmes increases day by day, and can be easily found on internet thorough the sites giving hacking information.

As the increasing number of assaults to popular networks, software and hardware companies began releasing products on this topic. For example, huge number of companies have products that provide secure logon to Microsoft's NT based systems via a smartcard. And also, new generation keyboards are shipped with a built-in smartcard reader for this reason. Another product doing the same process is a key that is connected to the new genaration serial buses (USB). Both applications, restirict usage of a computer without having the key or the smartcard by authenticating the PIN on the key or smartcard.

The way for securing the data and the network, as clearly seen needs some steps. On first stage, a computer even a node or single computer must be operated only by registered users, hence this is one of the the weakest ring of the circle. On the next stage, private data on the storage devices must be encrypted. At last stage, network logging and data transfer must be encrypted. For keeping the network performance on acceptable levels secure messaging have to be used as needed. As the computer technology develops with high pace, the encryption and authentication algorithms must solely depend on dynamic datas. If these three steps can be built with close relation to each other, security for terminals and datas on the network can be obtained.

Smartcards will be used for the system that is described on this thesis. The reasons for this choice will be explained widely further. The main thema for the security on this thesis is "disabling human control on security steps"[1]. So, briefly the target is, filling human reasoned blanks.

Besides the applications using smartcards, the system designed will not use the mthod like verifying a specific data on the smartcard, but will use the cross-authentication of several smartcards on communicating terminals. By this method, the terminals should be verified and moreover the data on the netework should be secured without any information or interference of any user by the dynamic session keys produced by the smartcards.

Smartcards are more than "pin verifying memory pieces" today, and solely start to have active roles on our lives. By the year 2005, all our magnetic credit cards will be replaced by chipcards as the EMV conformations as it offers high security for blocking the frauds. So, using this type of device as only a authentication device will be littering its adventages and potential.

As the smartcard is one of the leading role players of this thesis, it will be much more accurate designing the whole sytem according to its capabilities. So far, the smartcard will not only be the authenticating device for the system, but also an identity card holding personal informations, a medical card holding emergency data and also the drugs kept using, a key for doors and also keeping the credits that can be used for transactions[2,3]. Perhaps, seeming futurastic, but this will probably be one projection of human life in decades. One thing to remember is the system designed is not an utopia for now also, recent technologies on hardware and software enables maintaining this project. The only need is a strict co-operation or perhaps a standardization. This may be possible as the standardiztion of applications on smartcards initialized by EMV standarts will reach a consensus covering whole applications.

1. GİRİŞ

Günümüzde bilgisayar sistemlerinin güvenliği önemi gittikçe artan ve halihazırda hayatımızın pek çok noktasında karşımıza çıkan bir konudur.Artık insanlar bankalardaki hesaplarını internet üzerinden işletmekte, internet üzerinden kredi kartlarını kullanarak alışveriş yapmaktadırlar. En basit olarak, POS cihazlarında kredi kartları kullanılarak alışveriş yapılmaktadır. İşi gereği bilgisayar kullanan insanlar ise kendine ya da şirketine ait özel bilgileri korumak ya da uzak bir bilgisayara erişirken hem kendi hem de erişilen sistemin güvenliği için belli yöntemler seçmek zorunda kalmaktadır. Her ne kadar görülmese de bu işleri yaparken yoğun bir güvenlik çemberince sarıldıklarından artık hemen hemen hiçkimsenin şüphesi yoktur. Zira, artık çalınan kredi kartı bilgileri, ya da banka hesaplarının şifreleri günlük hayatta karşılaşılan sıradan haberler olmuşlardır. En sıradan kullanıcı bile bu tarz işlemleri yaparken "güvenlik" sorununu düşünmektedir..

Kredi kartlarındaki yolsuzluklar "veri güvenliği" konusunun önem kazanmaya başlamasında önemli bir aşama olmuştur. Konu üzerinde çalışan VISA, Europay gibi firmaların da desteği ile çeşitli veri şifreleme ve saklama standartları oluşmuş ve yaygın olarak kullanılmaya başlanmıştır. İnternet ortamında ticaretin yaygınlaşması ise tam bir dönüm noktası olarak kabul edilebilir. Bu tarihten sonra veri güvenliği sadece çok özel kurumların kullandığı bir kavram olmaktan çıkıp sıradan insanların bile dikkat ettiği bir nokta olmuştur. Bugün internet üzerinde işlem yapan herhangi bir kullanıcı, kulaktan dolma bilgilerle bile olsa SSL gibi teknolojileri aramaktadır.

İçinde bulunulan çağın "bilgi çağı" olarak adlandırılması önemli bir sorunu da beraberinde getirmiştir. Artık değer yargılarının tepesinde "bilgi" oturmakta, bu da bilgiyi en değerli meta haline getirmektedir. Her yücelen kavramda olduğu gibi popüler olan bilgi kavramı da kötü niyetli kişilerin en çok ilgilendikleri ve kısa yoldan ele geçirmeye çabaladıkları bir odak olmuştur. İnternet ve ağ altyapısının gelişmesine paralel olarak,

gelişmiş toplumlarda eski moda hırsızlıkların yerini neredeyse tamamen “bilgi hırsızlığı” almıştır. Zira artık, insanlar finansal birikimlerini bile dijital bilgi parçacıkları ile korur hale gelmiş bulunmaktadır.

Akıllı Kartlar

Akıllı kartların yeteneklerinin ve güvenliğinin gün geçtikçe artması, güvenlik kıstasları çok sert olan bir çok uygulama için bu cihazları gözde haline getirmiştir. Artık basit şifre doğrulayarak çalışan modellerden, çok hızlı RSA şifreleme yapabilen[4-6] modellere kadar geniş bir ürün yelpazesi bulunmaktadır. Bu tezde kullanılması düşünülen temel güvenlik cihazı bir akıllı kart olduğu için 2. bölümde akıllı kartlar, teknolojileri ve güvenilirlikleri konusuna değinilecektir.

Şifreleme, Doğrulama Yöntemleri

Veri güvenliğinin temelinin dayandırıldığı konu aslında verilerin hem yetkisiz kişilerin eline geçmemesi hem de bu bilgilerin aktarılırken olası bir izleyiciye karşı anlaşılmaz hale getirilmesidir. Bu nedenle 3. bölümde hali hazırda kullanılan şifreleme ve doğrulama yöntemlerinin temellerine değinilecektir. Aralarındaki temel farklılıklar ve performansları irdelenecektir

İşletim sistemleri: Güvenlik Önlemleri, Güvenlik Açıkları

Tez içerisinde gerçekleştirilmeye çalışılan konulardan biri de, bir işletim sisteminin nasıl güvenli hale getirileceği, yeniden güvenli bir işletim sistemi yaratmaktan öte, yeni nesil bir güvenlik paketinin bir işletim sistemine entegre edilmesidir. İşletim sistemlerinin ortak karakteristikleri ele alınarak genel olarak işletim sistemlerine eklenebilecek bir paket yaratmak amaçlanmıştır. Bu nedenle 4. bölümde genel olarak işletim sistemlerinin kullanıcı kabul etme, veri ve ağ güvenliği özellikleri incelenecek genel olarak zayıf yanlarından bahsedilecektir.

Modüller ve Tasarım

Tezin konusu üzere akıllı kartlar kullanılarak bir güvenlik paketinin nasıl tasarlanabileceği 5. bölümde ele alınacaktır. Bu bölümde, incelen akıllı kart, şifreleme-doğrulama yöntemleri ele alınacak, işletim sistemlerinde belirlenen aksaklıkları gidermeye yönelik bir sistem tasarlanacaktır. Tasarım aşamasında akıllı kartların

yetenekleri de göz önünde tutularak, çok fonksiyonlu bir tasarım söz konusu olacaktır. Kullanılacak akıllı kart, güvenlik kartı özelliğinin yanı sıra, kimlik bilgileri, acil ilaç-hastalık bilgileri, yetkili olduğu kapıları açma ve bir e-cüzdan özelliklerine de sahip olacaktır.

Gerçekleme

Temelleri verilen sistemin hayata geçirilme prensipleri 6. bölümde ele alınacaktır. Bu bölümde tasarım aşamasında planlanan bazı fonksiyonların nasıl gerçekleştirilebileceği üzerinde durulacaktır.

Sonuçlar

Tezin 7. bölümünde genel hatları ile tasarlanan sistem özetlenecek, elde edilen sonuçlar objektif bir şekilde yansıtılacaktır.

Tartışma

Tezin 8. bölümünde, tezin gerek teknolojik eksikliklerden gerekse de zaman problemi ile üzerinde durulamayan noktaları irdelenecek, yapılabilecek geliştirmeler açıklanacaktır.

Bu tez üzerinde yer alan bilgilerin kaynakları, tezin sonunda yer alan "Kaynaklar" başlığı altında verilecektir.

Tez yazımında verilemeyecek derecedeki uzun açıklamalar, tezin en sonunda ekler bölümünde sunulacaktır.

Tezi yapan araştırmacıya ait bilgiler, daha ileri araştırma yapmak isteyen araştırmacılara bilgi sunmak amacı ile yine tezin son bölümünde verilecektir.

2. AKILLI KARTLAR

Akıllı kartlar özellikle GSM uygulamalarının yaygınlaşması ile birlikte popüler hale gelmişlerdir. İlk örnekleri, sadece veri okunup yazılabilen bellek modülleri iken günümüzde metal kontaklara ihtiyaç duymayan, bir çok aritmetik işlemin yanısıra, yüksek güvenlik sistemlerine sahip olan cihazlar haline gelmişlerdir. Bu bölümde donanım ve yazılım alt başlıkları altında akıllı kartlara ilişkin genel veriler aktarılmaya çalışılacaktır.

2.1 Donanım

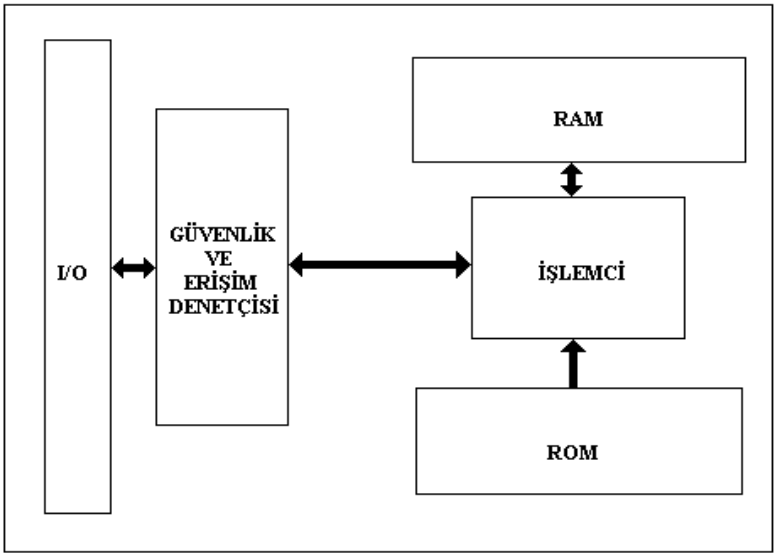
Akılı kartlar her ne kadar performans ve yapısal olarak çeşitli farklılıklar içerse de, temel olarak kullanılan hafıza, güvenlik, erişim ve denetim işlemleri için aynı bileşenlere sahiptirler.

2.1.1 Temel bileşenler

Günümüz teknolojisine sahip bir akıllı kart giriş için bir klavyesi, çıkış için de bir monitörü olmasa da hemen hemen modern bir bilgisayarın sahip olduğu tüm özelliklere sahiptir.

Akıllı kart sisteminin temelinde bir işlemci oturmaktadır. Doğası gereği akıllı kartların kontak sayısı sınırlıdır bu nedenle dış dünya ile bağlantısını bir seri yol vasıtası ile sağlar. Kullanıcılara ayrılmış bir RAM bloğu ve kartın iç işlemlerine ayrılmış bir ROM bloğu bulunur. “Security Logic (Güvenlik Birimi)” arabirimi ise verinin dışarıya verilip verilmeme koşullarını kontrol eder[7]. Aslen bir çok kartta RAM ve ROM blokları bir EEPROM ile oluşturulur. Hangi alanın RAM hangi alanın ROM gibi kullanılacağı ise güvenlik birimi tarafından belirlenir. Günümüzde kullanılan kartların hafızaları kullanım amacına yönelik olarak değişmekle beraber genellikle 256B ile 64KB arasında

değişmektedir. Bu değerlerin çok ötesinde özel amaçlara yönelik kartlar da mevcuttur, ancak maliyet analizi yapılacak olunursa, kart fiyatlarının bellek miktarları ile doğrusal olarak değişmediği gözlenebilir.



Şekil 2.1 Akıllı kart blok diyagramı.

Kartların boyutsal, kontak bilgileri ve elektriksel özellikleri ISO 7816 (1, 2, 3. bölümler) standardı tarafından belirlenmiştir.Bu standartlara göre akıllı kartlar haberleşme tiplerine göre iki bölüme ayrılırlar[8]:

2.1.2 Haberleşme protokollerine göre akıllı kartlar

- T0 protokollü kartlar : Bu protokolde iletişim karakter bazlı yapılır. Ve her bir karakter aktarıldıktan sonra onay alınır.
- T1 protokollü kartlar : Bu protokolde iletişim bloklar vasıtası ile yapılır. Her blok sonunda onay alınır, dolayısı ile aynı bit hızında çalışan bit T1 kart T0 kart karttan çok daha çabuk iletişimini tamamlayabilir.

Genelde akıllı kartların iletişim hızı 9600bps'tir. Ancak akıllı kart bu hızı yukarıya çekebilir. Kart hızının anlaşılabilmesi ve protokolünün tayini önemli bir noktadır. Zira bir

okuyucu T0 ve T1 protokolündeki kartları destekleyebilmeli, daha da önemlisi bunları ayırt edebilmelidir. Bu sorun akıllı kart okuyucuya ilk takıldığında çözülür. Tüm kartlar "Power On Reset" özelliğine sahiptir, yani kart, okuyucuya takıldığında, güç alır almaz "ATR – Answer To Reset" bloğu yayımlar. Bu veri kümesinin içinde kartın protokolü, bu noktadan sonra hangi hızda haberleşebileceği gibi bilgiler yer alır. Okuyucu taraf bu bilgiler ışığı altında kartın özelliklerini belirler ve iletişim kuralları belirginleşmiş olur[8].

2.1.3 İletişim arabirimlerine göre akıllı kartlar

Akıllı kartlar iletişim arabirimlerine göre de sınıflandırılabilirler. Bunlar:

- Kontaklı akıllı kartlar
- Kontaksız (RF) akıllı kartlar
- Hibrit ya da ikili (Dual) kartlar

Kontaklı kartlar, kart yüzeyi üzerinde genellikle altın ya da platin metal kontaklara sahiptir. Kartın asıl iş yapan birimi bu kontaklar ve çok yakın çevresidir, bu nedenle daha küçük boyuta indirgenebilirler. Genellikle iki tip kontaklı akıllı kart görülür, bunlar tam boy, "full size", akıllı kartlar (kontürlü telefon kartları gibi), ve mikro boy "micro size" ya da "sim size", kısaca SIM denilen küçük boyuttaki (GSM SIM kartları gibi) kartlardır. İç yapı olarak tamamen aynı olmalarına karşı plastik kart ebatları oldukça farklıdır. Kontaksız kartlar ise tam boy ebadındadırlar ve çok özel uygulamalar haricinde SIM boyutuna küçültülemezler. Zira kart, plastik kısmını anten olarak kullanmaktadır. Ancak, gelişen teknoloji sayesinde anteni çipinin üzerinde olan, "on chip antenna" [9], kartlar da piyasaya çıkmaya başlamıştır. İkili ve hibrit kartlar ise her iki teknolojiyi de bünyesinde barındıran kartlardır. Hibrit kartlar, tek bir çekirdeğe sahiptir, sadece iletişim arabirimi RF ve kontaklar olmak üzere iki adettir ve her durumda işleyen kart aslında aynıdır. İkili kartlar ise basitçe bir RF, bir de kontaklı kartın birleştirilmesinden oluşur ve kontak kısmını kullanan çip ile RF anteni kullanan çip farklıdır.

2.1.4 Akıllı kartlar üzerinde donanımsal güvenlik

Teknolojinin gelişmesine paralel olarak, kartların güvenlik seviyelerinin arttırılmasına yönelik çalışmalarla, akıllı kart bünyesinde bulunan tüm birimler tek bir çekirdeğe toplanmıştır. Böylelikle arabirimler arasındaki sinyallerin izlenmesi olasılığının önüne geçilmiştir. Bu nokta akıllı kartlarda güvenlik konusunda ne kadar titiz davranıldığının altını çizecek bir uygulamadır.

Akıllı kartların, özellikle finansal alanlarda kullanılmaya başlanmaları üzerine, kart üzerindeki "güvenlik birimi" sadece şifre doğrulayabilen bir yapı olmaktan çıkmaya başlamıştır. Bugün, çoğu yüksek seviyeli güvenlik kartında gelişmiş şifreleme işlemlerini mümkün kılmak ve hızlandırmak amacı ile bir şifreleme yardımcı işlemcisi (crypto co-processor) bulunmaktadır. Doğal olarak bu eklentiler kart maliyetlerine de yansımaktadır. Gelişkin RSA kodlaması yapabilen bir kart 1024 bitlik anahtarlar kullanarak 10-100 sekizlikten oluşan verileri bir ila iki saniye arasında şifreleyebilmektedir[4]. DES ve 3DES (Tripple DES, TDES) artık bu tip kartlarda standart olmuş algoritmalardır. PIN doğrulama ise halen kartlarda bulunmaktadır, ancak bu doğrulama yöntemi yazılım desteği ile geliştirilerek kartın dosyaları için özelleştirilebilmektedir.

2.2 Yazılım

Akıllı kartların günümüzde birer küçük bilgisayar formunu almaları artık basit bir bellek yönetim mikrokodu ile çalıştırılmalarını imkansız hale getirmiştir. Her gün gelişen teknolojiye ayak uydurabilmek amacı ile kartların donanımsal yapıları geliştirilse de, donanımda yapılabilecek değişiklik ve geliştirmeler sınırlı olduğundan dolayı, geliştirme sürecinde yaratıcılığa daha açık ve yaratıcılığı daha kolay olan yazılım önemli bir konuma gelmiştir. Nasıl ki modern bir bilgisayar tam randımanlı çalışabilmesi için modern bir işletim sistemine ihtiyaç duyuyorsa, yeni teknolojiye sahip bir akıllı kartın işletim sistemi de kartın en az donanımı kadar can alıcı bir rol üstlenmektedir.

2.2.1. İşletim sistemi

Tüm yeni nesil akıllı kartlarda üreticisine özel bir işletim sistemi bulunur. Her şeyden önce işletim sisteminin ana görevi kartın bir okuyucu ile ISO 7816-1, 2, 3 standartlarında haberleşmesini sağlamaktır. Kartın komut kümesi ise ISO 7816-4 standartları göz önüne alınarak oluşturulur. Ancak bu standart ancak okuma, yazma gibi temel komutları kapsamakta, karta ve üreticiye özel komutlar ise açık bırakılmaktadır. Bu nedenle, kartın dokümanları geliştirme yapılırken standarttan daha yararlı olacaktır.

İşletim sisteminin diğer önemli görevi, dosya yapılarının oluşturulması ve bunların korunmasıdır. Bu noktada "güvenlik birimi" işletim sistemi ile sıkı ilişki içindedir. Zira herhangi bir dosya oluşturulurken aynı zamanda o dosyaya erişim hakları da belirlenir. Bu erişim haklarını şöyle sınıflayabiliriz:

- Okuma Hakkı
- Yazma Hakkı
- Değiştirme Hakkı
- Silme Hakkı

"Yazma" ve "Değiştirme" hakkı aslında aynı işi yapıyor gibi görünebilir ancak akıllı kartlar bir alanda daha önceden veri yazılıp yazılmadığını anlayabilecek yetenektedirler. Bu yeteneğe sahip olmayan kartlarda bu erişim yetkisi zaten bulunmamaktadır. Silme hakkı ise bir dosya ya da klasör (DF- Dedicated File) silinmesi için gerekli erişim izni olarak tanımlanabilir. Bu erişim hakları genellikle yaratılma esnasında belirlenen bir anahtarın doğrulanması şeklinde kullanılır.

Gelişen ihtiyaçlar nedeni ile, akıllı kartlar birden fazla uygulamayı üzerlerinde taşıyabilecek yapılara sahip olabilecek şekilde esnek işletim sistemlerine ihtiyaç duymaktadır. Bu nedenle bir çok firma "açık platform" (Open Platform), JavaCard, MulTOS gibi yazılım ile özellikleri geliştirilebilen kartlar çıkartmaya başlamışlardır. Bu tip kartlarda temel, çekirdek bir işletim sistemi hali hazırda mevcut olmakta, ancak çeşitli yazılımlarla, ki popüler olarak JavaCard'lar için Java dili kullanılarak, kartlara istenilen

ek özellikler kazandırılabilmektedir. Ancak halen bu desteğe sahip akıllı kartlar maliyet açısından üst noktalarda bulunmaktadır.

İşletim sisteminin önemli olan diğer bir görevi ise anahtar bilgilerini saklamaktır. Hiç bir koşul altında anahtarlar dış dünyaya verilmemelidir. Bu nokta tezin can alıcı noktasıdır. Doğrulama ve şifreleme her zaman iki kart arasında, kullanıcının ve sistem yetkilisinin dahi bilmediği bir şifre ile yapılacaktır. Böylelikle şifrenin 3. şahısların eline geçme olasılığının olabildiğince azaltılması hedeflenmektedir.

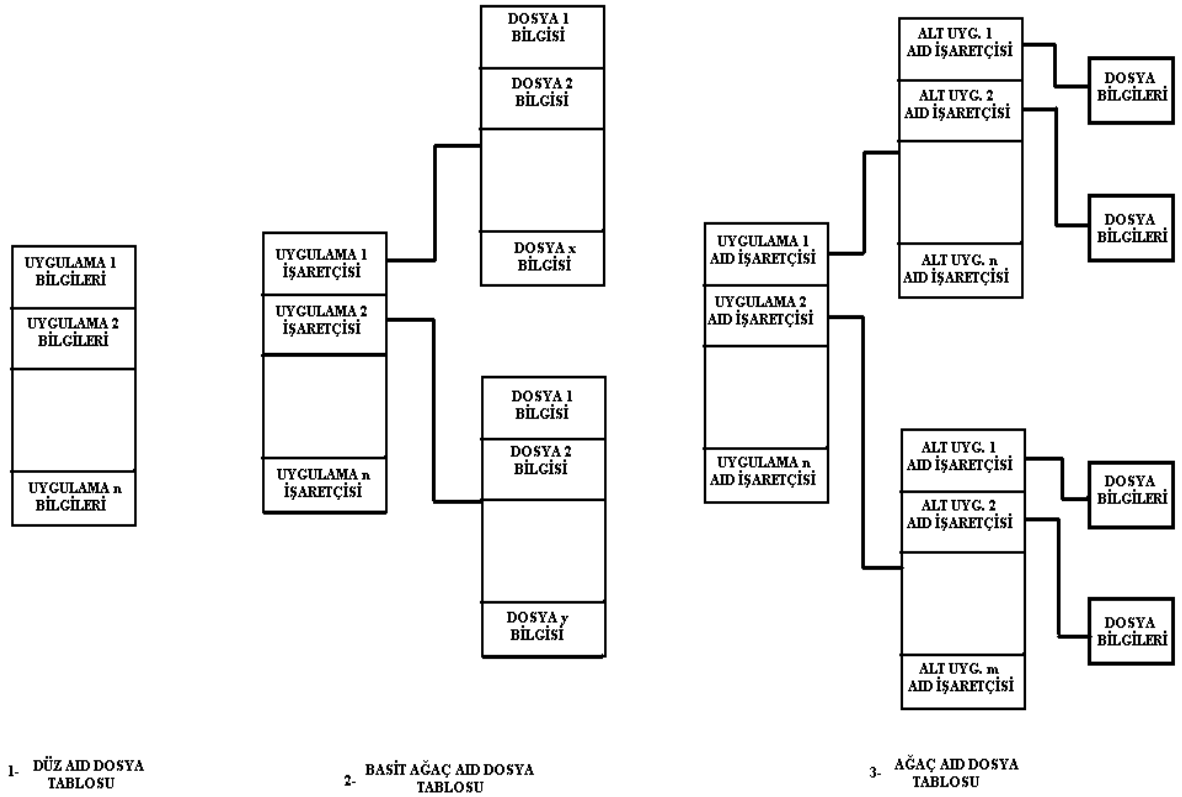
2.2.2. Dosya sistemi

Akıllı kartlar, değişik uygulamaları bünyesinde barındırabilmek için her uygulamaya özel bir klasör kullanırlar. Her akıllı kart kullanıcının veri depolama alanı olarak görebileceği bir ana dosyaya (MF – Master File) sahiptir. Uygulamalar bu dosya altında değişik dosyalar altında bulunabilirler. Peki bir kartın altında hangi uygulamaların olduğu nasıl anlaşılacaktır? Bir uygulama listesi, AID (Application Identifier), bu konuda yardımcı olabilir. Temel çalışma prensibi özetlenirse: Ana AID dosyası MF altında bulunur ve sadece okuma erişim hakkı sınırlandırılmamıştır. Karta erişen bir uygulama ilk önce bu dosyayı okuyarak kart üzerinde desteklenen uygulamalar hakkında fikir sahibi olur. Bu noktadan sonra yöntem üç şekilde devam ettirilebilir.

- Kartı okuyan uygulama kendisine tanıdık gelen uygulamaya ait tüm alt dosya adlarını bilmektedir ve uygulama seçimi bu noktada biter(Düz AID Dosya Tablosu).
- AID dosyası tüm alt dosyaların adlarını ve erişim şartlarını ilgili dosyanın içeren dosyanın adını içerir. Bu dosya okunarak uygulamanın tüm dosyaları hakkında bilgi sahibi olunabilir (Basit Ağaç AID Dosya Tablosu).
- Her uygulama ve her uygulamanın alt uygulamalarının dosyalarının altında da birer AID dosyası bulunur ve ana AID dosyası uygulamanın AID dosyasına o da ilgili alt uygulamanın AID dosyasına işaret eder. Böylece bir ağaç yapısı oluşturulur. Tüm AID dosyaları erişim bilgilerini de içerir. Ayrıca bir bayrak AID dosyasının ağacın bir dalı ya da yaprağı mı olduğunu ifade eder. Yaprak

konumunda bulunan AID başka bir AID'yi değil bir dosyayı işaret etmektedir (Ağaç AID Dosya Tablosu).

İlk yöntem gerçekleştirmesi son derece kolay ancak esnek olmayan bir çözümdür, bellek kullanımı ise son derece makul olacaktır. Yeni bir uygulama eklemek ise kolay ancak kartı okuyan uygulamaya da müdahaleyi gerektirir. Son yöntem ise son derece esnek, ancak kullanımı diğer yöntemlere nazaran daha fazla zaman alıcı ve gerçekleştirilmesi daha zor bir yöntemdir. Yeni bir uygulama ya da alt uygulama eklemek son derece kolaydır. 2. yöntem ise ana dosya altındaki dosya sayısı sınırı çok küçük olmadığı sürece kullanılabilir. Ancak bir alt uygulama eklemek mümkün değildir. Şekil 2.2'de AID'ye ilişkin yöntemler görülebilir.



Şekil 2.2 AID oluşturma yöntemleri.

3. VERİ ŞİFRELEME – DOĞRULAMA METODLARI

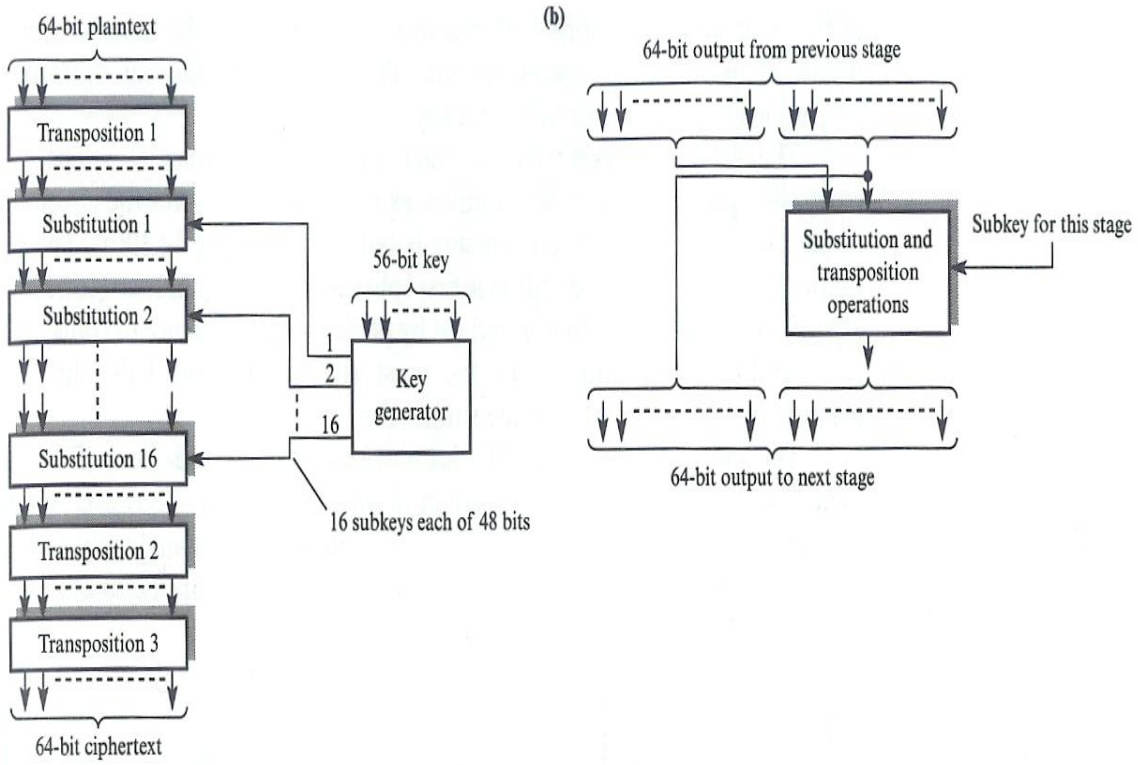
Günümüzde çok çeşitli veri şifreleme yöntemleri kullanılmasına karşın, kullanılan bu yöntemlerin temelinde DES (Data Encryption Standart) ve RSA (Rivest, Shamir and Adleman Algorithm) yöntemleri ve bu yöntemlerin mutasyonları başrolü oynamaktadır.

3.1 DES (Data Encryption Standart) Yöntemi

DES algoritması bir blok şifreleme algoritmasıdır, yani algoritma sabit sayıda sekizlilerden (8 sekizlik) oluşan bloklar ile çalışabilir. Bu nedenle algoritma uygulanmadan önce veri yığını 64 bitlik parçalara bölünür ve gerekiyorsa veri bloğu 64 bitin bir katına tamamlanır. 56 bitlik bir anahtar ile 64 bitlik bu bloklar şifrelenir. Şifrelenmiş veriyi alan taraf da aynı anahtarı kullanarak çözme işlemini uygular. Şekil 3.1'de DES algoritmasının çalışmasına ilişkin blok diyagram görünmektedir. Şifrelenmemiş bloğa ait bilgiler bir yer değiştirme işlemine tabi tutulduktan sonra 56 bitlik anahtardan üretilen 16 adet 48 bitlik alt anahtar, sonuç veriye çıkartma işlemi olarak uygulanır. Daha sonra sonuç bloğunun en anlamlı 32 bitlik kısmı ile düşük anlamlı 32 bitlik kısmı yer değiştirilir. Son adımda ise elde edilen bloğa birinci adımda uygulanan yer değiştirme kuralı uygulanarak nihai sonuç elde edilir.

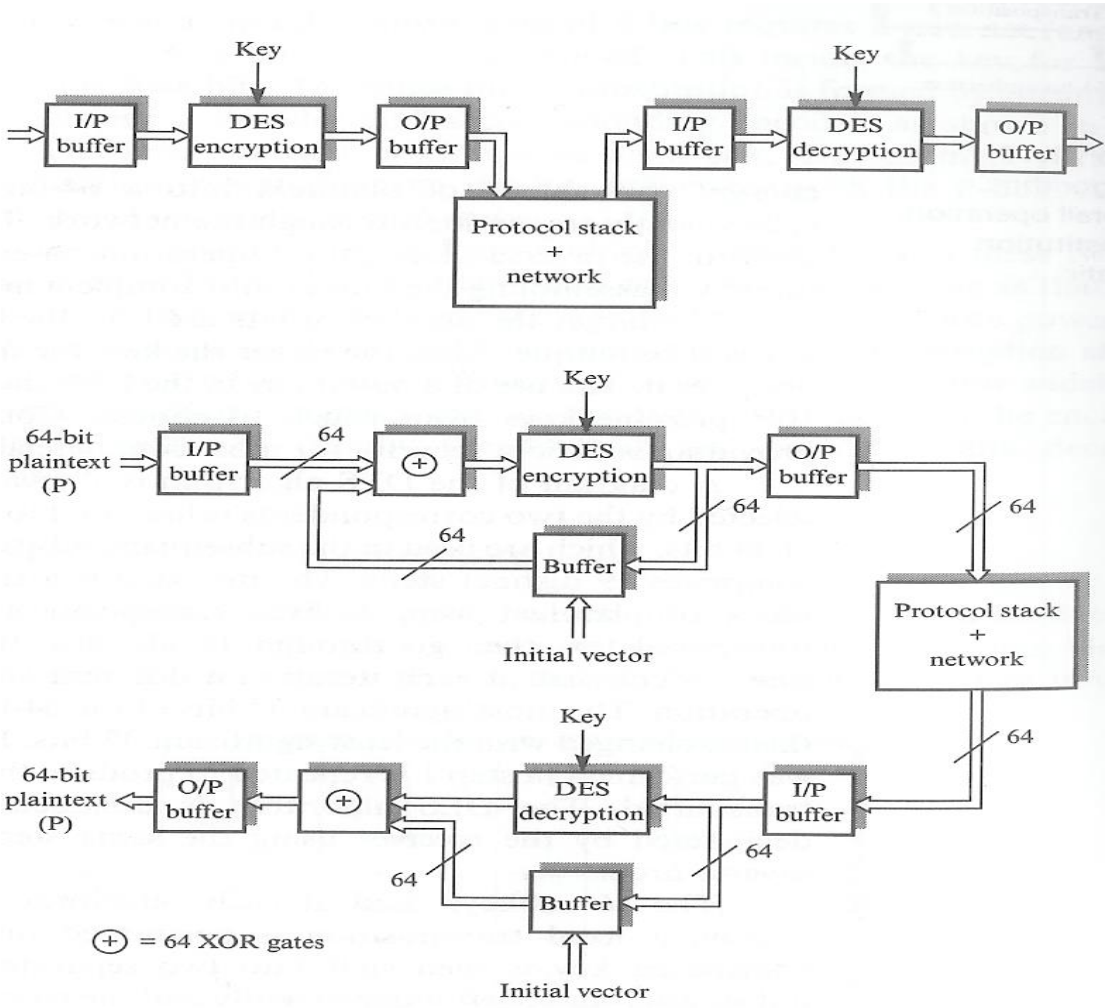
16 adet alt anahtar ise şöyle oluşturulur; ilk önce 56 bitlik anahtara bir yer değiştirme algoritması uygulanır, sonuç iki adet 28 bitlik bloklara bölünür. Elde edilen 28 bitlik bloklar bağımsız olarak alt anahtar numarası kadar sola döndürülür ve tekrar birleştirilir. Elde edilen son bloğa yeniden yer değiştirme algoritması uygulanarak sonuç alt anahtar elde edilir.

Açıkça görüleceği üzere aynı anahtar kullanılarak işlemler ters olarak uygulanırsa şifrelenmiş veriyi çözmek mümkündür.

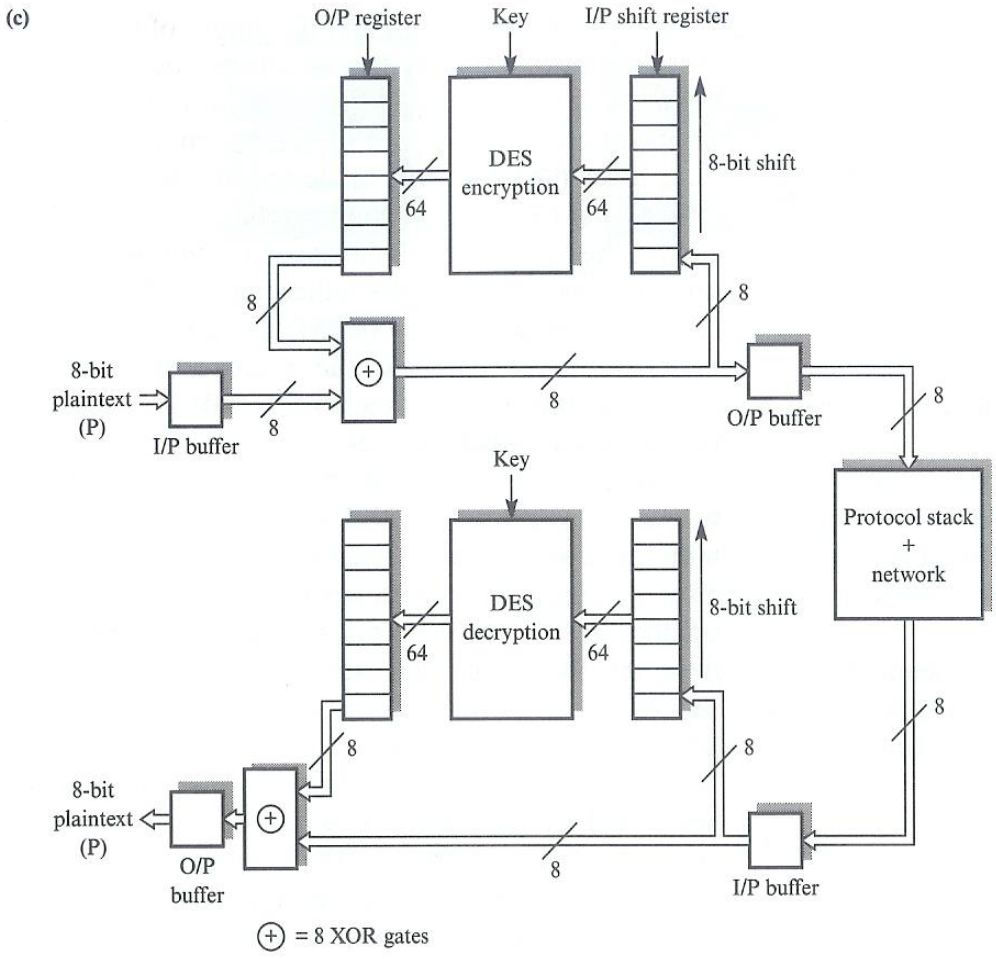


Şekil 3.1 DES algoritmasının temel çalışma mantığı

DES'in bu şekilde çalıştırılmasına ECB (Electronic Code Book) adı verilir. Tüm 64 bitlik alanlar birbirinden bağımsız olarak şifrelenir. Diğer bir DES türevi olan CBC (Chained Block Cipher) yönteminde ise şifrelenen her blok bir sonraki şifrelenmemiş 64 bitlik blok ile dar veya işleminde tabi tutulur. Böylelikle verinin tümü birbiri ile ilişkilendirilmiş olur. İlk blok ise her iki taraf tarafından ortaklaşa belirlenen bir ilklendirme vektörü ile dar veya işlemine tabi tutulur. CFM (Cipher Feedback Mode) yönteminde ise veri 64 bitlik değil 8 bitlik yığınlar halinde ele alınır. Yöntem temel olarak CBC yöntemine benzer ancak her işlemde 8 bit ötelenen iki adet tampon kullanılmıştır. Şekil 3.2 ve Şekil 3.3'de bu yöntemlere ait blok diyagramlar görülebilir[10],722-726.



Şekil 3.2 Sırası ile DES ECB ve CBC Modları



Şekil 3.3 DES CFM modu.

3.2 RSA (Rivest, Shamir and Adleman Algorithm) Yöntemi

RSA yönteminin en önemli özelliği verinin şifrelenmesi için kullanılan anahtar ile çözülmesi için gerekli anahtarın aynı olması gereğinin ortadan kalkmasıdır. Bu yöntemde bir adet açık anahtar (şifreleme işlemi için) bir adet de gizli anahtar (çözme işlemi için) kullanılır.Yöntem bir dizi üs alma ve modüler aritmetik işleminden oluşur. Bir örnek ile açıklanacak olursa :

Açık anahtar için :

- İki adet pozitif asal sayı alınır (P = 7, Q = 17)
- X = (P-1) (Q-1) hesaplanır (X = 96)
- X ile aralarında asal olan bir E sayısı alınır (E = 5)
- N = P Q hesaplanır (N = 119)
- Açık anahtarımız 119, 5 ikilisidir. (Kp, Ep)

Gizli anahtar için

- MOD (D*E, X) = 1 olacak şekilde D hesaplanır (MOD (D*5,96) = 1 → D=77)
- Gizli anahtarımız 119,77 ikilisidir. (Ks, Es)

a = 19 verisini şifrelersek : a' = MOD (a^{Ep}, Kp) → a' = MOD (19^{5}, 119) = 66

a' = 66 verisini çözersek : a = MOD (a'^{Es}, Ks) → a = MOD (66^{77}, 119) = 19

olarak buluruz.

Yöntemin can damarı olan P ve Q asal sayıları her zaman saklı kalmalıdır. Bu nedenle N sayısı olabildiğince büyük seçilmelidir ki herhangi bir bilgisayar sistemi ile dahi çözümleme işlemi kabul edilemez derecede uzun sürsün. Yapılan ölçümlere göre standart bir sistem ve N değerleri için Tablo 3.1 deki sonuçlar elde edilmiştir[10],728.

N	**Hesaplama Süresi**
100 hane	~ 1 hafta
150 hane	~ 1000 yıl
>200 hane	~ 1 milyon yıl

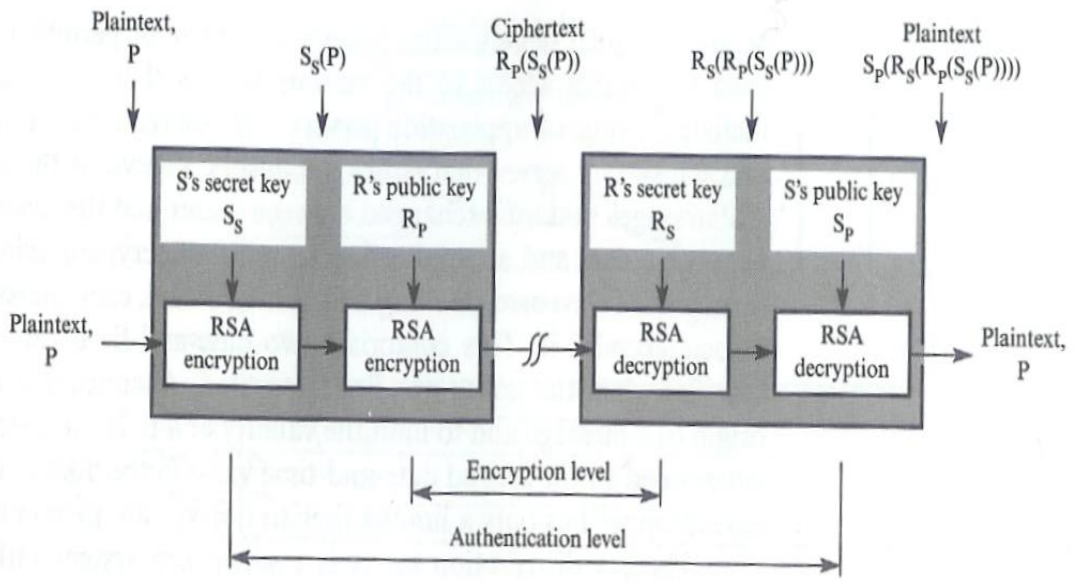
Tablo 3.1 RSA için anahtar uzunluğuna göre anahtarın kırılma zamanı

3.3 Veri Şifreleme Algoritmalarına Genel Bir Bakış

Şifreleme sistemlerinde genel olarak kullanılan iki algoritma da basitçe ele alındıktan sonra, DES yönteminde göze çarpan en büyük eksiklik anahtarın her iki tarafça da bilinmesi gerekliliğidir. Böyle bir sistemde anahtarı ele geçiren kişi her iki tarafın bilgilerine de ulaşma yeteneğine kolaylıkla sahip olabilmektedir. Veri şifreleme için kullanılan ve çıkışları saniyede 1 Megabit'e ulaşılan entegrelerin piyasada bulunduğunu göz önüne alırsak, bilgi toplamaya çalışan kişinin tek yapması gereken orijinalini bildiği bir mesajın şifrelenmiş halini ele geçirmek olacaktır. Bu nedenle DES algoritması kullanılırken genellikle değişik kombinasyonlar kullanılarak algoritma farklılaştırılmaya çalışılır örneğin mesajın şifrelenmemiş bölümünde geçen ya da her iki tarafça da bilinen bir veri (örneğin seri numarası) ile veri çeşitli işlemlere tabi tutulabilir, ek olarak DES yerine art arda 3 defa (şifreleme-çözme-şifreleme şekline) DES uygulanan TDES (3DES – Tripple DES) yöntemi kullanılabilir. Kötü olan kısım ise anahtarın yüzlerce uç birimin olduğu bir ağda herkesçe bilinmesi gerekliliğidir[11].

RSA yönteminde ise gizli anahtarı sadece alıcının bilmesi yeterlidir ve diğer uç birimlerin açık anahtarları farklı olabilir. Ancak, bu noktada iki temel sorun belirmektedir. Birincisi, açık anahtar herkesçe bilindiğine göre bir mesajın doğru kaynaktan gelip gelmediği nasıl anlaşılacaktır. İkincisi de tüm uç noktaların veri gönderip alabildiği Ethernet modelindeki bir ağ yapısında durum ne olacaktır. İlk sorunun çözümü için önerilen bir kaç metot vardır. Bunlar :

- Mesaj Doğrulama Sistemi : Temel olarak göndercinin kendi gizli anahtarı ile şifrelediği mesajı tekrar alıcının açık anahtarı ile şifrelemesi, alıcının da kendi gizli anahtarı ile mesajı çözdükten sonra göndericinin açık anahtarı ile mesajı çözebilmesine dayanır. Eğer mesaj gerçekten çözülebilirse mesajın doğru kişiden geldiği anlaşılır. Şekil 3.4'de sisteme ait blok diyagram görülmektedir.



Şekil 3.4 Mesaj doğrulama protokolü blok diyagramı.

Ancak mesajın gerçekten doğru olarak çözüldüğünü anlamak için de ek bir bilgiye ihtiyaç vardır. Diğer yandan RSA, hesaplama süreleri bakımından oldukça maliyetli bir algoritmadır[10],726-729.

- Dijital İmza : Bu yöntemde verinin şifrelenmesi yerine, sadece mesajın sonuna göndericinin gizli anahtarı ile şifrelenmiş bir blok eklenir. Eğer alıcı bunu göndericinin açık anahtarı ile çözebilirse, mesaj doğrulanmış olur. Ancak verinin açık bir şekilde iletilmesine çözüm getirmez[10],730.
- Güvenilir Sunucu Kullanımı : Bu sisteme bir örnek olarak Kerberos sistemini gösterebiliriz. Sistem sadece gizli anahtarların kullanımına yönelik hazırlanmıştır, ve çeşitli birimlere ulaşım hakları için izin jetonları kullanır. Her bir izin jetonu üzerinde geçerlilik süresini belirten bir tarih zaman bilgisi bulunur. Sonuç itibarı ile sisteme girmeye çalışan kişinin çözme işlemini tamamlayabilmesi için sınırlı zamanı vardır. Sistemde herhangi bir uç birime bağlanmak isteyen bir birim, tüm Kerberos sunucularını koruyan izin sunucusuna bağlanmak ve onay kodu ve

bağlanmak istediği sunucuya ilişkin kullanabileceği bir anahtar almak zorundadır. Bu ikili, bahsedildiği üzere belli bir zaman zarfı için geçerlidir[10], 730-732, [11]. Yani, izin sunucusu aslında uygulama sunucusu ile bağlanmak isteyen uç birimi kısıtlı bir süre için birbiri ile konuşturmaktadır. Ancak, bu geçici anahtarlar gizli kalmaktadır. Sonuç olarak, izin sunucusuna bağlanmak için gerekli olan şey kullanıcı adı ve parolasıdır ki bu bilgiler 3. şahıslar tarafından kolaylıkla ele geçirilebilecek bilgilerdir. Kerberos, ağ alt yapısını ve şifrelemeyi kullanıcılardan gizlemeyi başarmış ancak uç birimi korumayı başaramamıştır.

Görüldüğü üzere DES ya da RSA kullanılmasından ziyade veri güvenliğinin sağlanması gereken sistemlerde önemli olan nokta anahtarların gizliğinin korunması ve mümkün ise karşılıklı kullanılan anahtarların dinamik olmasıdır[11-13]. Yüzlerce uçbirimin aynı anahtarı kullandığı bir ağda 3. şahıslar için anahtar ya da anahtarlara yönelik örnek toplama şansı çok yüksek olacaktır. Bunun yerine, her oturum kurulduğunda değişecek bir anahtar kullanılması çok faydalı olacaktır.

4. İŞLETİM SİSTEMLERİ – GÜVENLİK ÖNLEMLERİ ve AÇIKLARI

Modern her işletim sistemi, üzerinde çalıştığı bilgisayar ve aynı zamanda ağı korumak için bir çok güvenlik desteğine sahiptir. Ancak halihazırda mevcut bulunan bu güvenlik duvarlarının yetersiz kaldığı, hemen her türden sisteme yapılan saldırılar ile belirgin biçimde görülmektedir. İşletim sistemlerinin güvenlik için kullandığı yöntemler iki gruba ayrılabilir. Bunlar; üzerinde çalışılan bilgisayarı yetkisiz kişilerin erişiminden koruma, ki bu "kullanıcı koruması" olarak adlandırılabilir ve bir ağ ortamındaki kaynakların sadece yetkili kullanıcılar tarafından kullanılmasını sağlayan paylaşım ve ağ koruması olarak adlandırılabilir.

4.1 Kullanıcı Koruması

İşletim sisteminin temel fonksiyonlarından biri yetkisiz bir kişinin ilgili bilgisayarı kullanmasını önlemektir. Hemen hemen tüm işletim sistemlerinde, kurulum esnasında bir yönetici hesabı oluşturulur. Bu hesap yardımı ile daha sonra sisteme kullanıcı eklemek, çıkarmak yada herhangi bir kullanıcının yetkilerini sınırlandırmak gibi işlemler yapılabilir. Yönetici hesabının işletim sistemi üzerinde tüm yetkileri mevcuttur ve genellikle bu yetkiler başka bir kullanıcıya tam olarak verilmez.

İşletim sistemi açılırken, sisteme bağlanmak için bir kullanıcı adı ve şifresi talep edilir. Girilen kullanıcı adı işletim sisteminin kullanıcı listesinde yer alıyor ise şifre kullanıcı listesindeki şifre ile karşılaştırılır. Şifre doğrulanır ise kullanıcı yine bu listedeki erişim hakları ile sisteme bağlanmış olur. Erişim hakları kısıtlamaları ile kullanıcını aynı sistem üzerinde tanımlı diğer kullanıcılara ait dosyaları okuması, bu dosyalara yazması, sisteme yeni programlar kurması, sistemin gerek iç dinamiklerini gerekse de ağ ortamındaki dinamiklerini değiştirebilme izinleri gibi özellikler sınırlandırılabilir.

Bu yöntemin esas problemi kullanıcı listesinin tutulduğu kaydın korunması gerekliliğidir. Şüphesiz ki bu liste gizli bir yerde ve şifrelenmiş bir şekilde tutulmaktadır. Ancak bu yöntem yine de özellikle sistem içindeki bir kullanıcının diğer kullanıcıların şifrelerini öğrenebilmesini engellemekte yetersizdir. Bunun için farklı yöntemler kullanılabilir. Deneyimli bir kullanıcı, bu listenin tutulduğu yeri ve şifrelenmiş bilgiyi çözme yolunu rahatlıkla bulabilir. Zira elinde bildiği bir kullanıcı adı-parola çifti bulunmaktadır. Herhangi bir kullanıcı, sistemde, basılan tuşları kaydeden bir program parçacığı çalıştırabilir. Bu program sistemdeki tüm kullanıcıların şifrelerini ve daha önemlisi dosya şifreleri gibi özel bilgileri de sahibine aktarabilir. Bu tip bir program, 3. şahıslar tarafından ağ ortamından da bir bilgisayara kurulabilir. Mevcut işletim sistemlerini yönetici yetkileri ile açabilen bir çok program da piyasada bulunmaktadır. Bu tür yasadışı programların sayısı ve içeriği işletim sisteminin popülerliği ile doğru orantılı olarak artmaktadır. Yaygın kullanım, işletim sistemin kullandığı yöntem ve anahtarların deşifre edilmesinde önemli rol oynamaktadır.

Kullanıcı adı – parola çiftinin önemli problemlerinden biri de kullanıcının sistemden uzaklaşması durumunda, sistemi bir şekilde kilitlemediği müddetçe kaynakların 3. şahıslara tamamen açık durumda kalmasıdır. Bunun için, belirli müddet içinde hiç bir işlem yapılmaması durumunda devreye giren şifre korumalı "ekran koruyucu" adlı programlar üretilmiştir. Şüphesiz ki, bu programların devreye girebilmesi için gereken süre ve kullandığı basit şifre algoritmaları güvenlik açığını kapatamamaktadır yani "kötü" bir çözümdür [14].

Anlaşıldığı üzere, basit bir kullanıcı adı-parola ikilisi bireysel bir sistemi korumakta yetersiz kalmaktadır. Bireysel bir sistem, ağda bir yaprak olabileceği için, içerdiği güvenlik açıkları ağın güvenlik açıkları sayılabilir. Sonuç itibarı ile sisteme bağlanmayı sağlayacak ve sistemin güvenliği sadece bağlanılırken değil sürekli sağlayacak bir yönteme ihtiyaç bulunmaktadır[15,16]

4.2. Paylaşım ve Ağ Koruması

Hemen hemen tüm işletim sistemleri bir ağa bağlanmak için ya kullanıcının sisteme bağlanırken kullandığı kullanıcı adı-parola ikilisini, ya da sırf ağ için düzenlenmiş farklı bir ikiliyi kullanır. Ancak bu denetim sadece bağlantı esnasında yapılırken, veri iletişimi esnasında kontrol genellikle uç noktalara atanan dinamik ya da statik adreslerin kontrolü şeklinde olmaktadır.

Uç noktalar üzerindeki işletim sistemleri, barındırdıkları hizmet ve dosyalardan hangilerinin, hangi kullanıcılar tarafından kullanılabileceğine dair bir tablo tutarlar ve bu kurallar topluluğuna göre erişim hakları sınırlandırılır. Çok kullanıcılı bir ağda her kullanıcıya erişim hakkı tanımlamak zahmetli bir hal aldığından dolayı ağ üzerindeki kullanıcılar belirli seviyelerdeki kullanıcı gruplarına ayrılır.

Ağ üzerinde haberleşme uç noktalar arasında adres kontrolü mantığına dayanır ve en yaygın kullanılanı IP modelidir. Bu sistemde her uç noktanın belirli kurallara göre düzenlenmiş bir IP adresi bulunur, bu adres uç birimlerde kalıcı bir şekilde oluşturulabileceği gibi, uç birimler açılırken bir sunucu tarafından atanacak şekilde de işletilebilir. Her iki durumda da sunucu ağ üzerindeki tüm uç noktaların adres bilgilerini bilmek durumundadır. Yapı aynı zamanda alt ağ tanımlamalarına da izin verir. Bu şekilde istek gönderen uç birimin hangi ağa mensup olduğu da kolaylıkla anlaşılabilir.

IP modelinde veri iletişimi uç birimler arasında mantıksal iskeleler (PORT – SOCKET) arasında yapılır. Sunucu terminal farklı hizmetler verebilmek için farklı soketlerden istek bekler. Normalde soketler arasında bilgi şifrelenmemiş bir şekilde taşınır, elbette ki bir şifreleme algoritması yardımı ile hattaki verilerin şifrelenmesi mümkündür. Aynı zamanda sunucuya bağlanma istekleri de bu protokole uygun yapılabilir. İstemci ağ açılış bilgiler uç birimler arasında şifrelenmemiş (plaintext mode) şekilde iletilebileceği gibi, şifrelenerek de gönderilebilir. Ancak bu şifreleme genellikle işletim sistemine has bir şekilde yapılır.

Uç birimler arasında iletilen bilgileri korumak için sıkı kuralları olan yüksek şifreleme özellikli protokoller de kullanılmaktadır. Örneğin SSL ya da SSL2 teknolojileri RSA

şifreleme mantığını kullanarak soketler arası bilgi gizlemesini sağlar. Ancak RSA algoritmasının getirdiği yoğun iş yükü nedeni ile genellikle kredi kartı numarası, banka hesap şifreleri gibi kısa bilgi parçacıklarını şifrelemek için ve genellikle tek yönlü bilgi şifrelemek amacı ile kullanılır. Günümüzde RSA algoritmasını hızlandırabilmek için yoğun çalışmalar yapılmaktadır.

4.3 Güvenlik Açıkları

Bir ağ ortamındaki en büyük güvenlik açıklarından biri uç birimlerdeki işletim sisteminin kullanıcısının kendisinden uzaklaştığını hızlı bir şekilde algılayamamasıdır. Bir sisteme bağlanmış uç birim, kendisini kullanan kişi hakkında bir fikre sahip değildir. Bağlanma işlemi gerçekleştikten sonra tüm sorumluluk kullanıcıya bırakılmakta bu da insan hatalarından kaynaklanan açıklar için büyük potansiyel oluşturmaktadır.

Ağ üzerindeki kullanıcılar ve veri, parolalarla korunulmaya ve yetkisiz kişilerin erişiminin önüne geçilmeye çalışılmaktadır. Ancak ağı dinleyen bir ağ dışı kullanıcı kolaylıkla sistemdeki adresleri öğrenebilir ve kendi makinesinin bu adreslerden birine sahipmiş gibi davranmasını sağlayabilir. Bu işleme "IP Spoofing" adı verilmektedir. Her ne kadar bu durumdan kurtulmak için ağ kartlarının donanımsal adres bilgileri kaydedilse de, ağ dışı kişinin standart bir sürücü yerine geliştirilmiş bir ağ kartı sürücüsü kullanarak bu bilgileri değiştirebilmesi mümkündür. Bu şekilde bir dinleyici sistemde isteği uç birimden, yetkisinin olduğunu anladığı, o an bağlı bulunan başka bir uçbirimin adresini kullanarak istediği bilgileri çekebilir. Sonuç olarak ağa bağlantı ne kadar güvenli yapılırsa yapılsın, sürekli uç birimlerinin geçerliliğinin denetlenmesi gerekir[15,16].

Görülüyor ki ağlardaki güvenlik açıklarının temeli gerek uç birimin kullanıcısını, gerekse de uç birimlerin birbirlerinin geçerliğini sınayamaması, ya da bu sınama için çok zayıf, kolay yıkılabilir yöntemler kullanmasıdır. Eğilinmesi gereken ana tema bu olmalıdır. Uç birim kullanıcısını tanıyabilmeli, uç birimler kendi aralarında da sürekli birbirlerini denetleyebilmelidir. Bunu yapmak için Kerberos sistemindeki gibi bir erişim sunucusu kullanılabilir[6]. Ancak bu yöntem çok kullanıcılı ve iletimin sadece sunucu-uç birim arasında olmadığı ağlarda performansı belirgin bir şekilde düşürecektir. Bunun yerine, her uç birimin mümkün olduğunca bir erişim sunucusu gibi çalışması, mümkün olmadığı

durumlarda erişim sunucusuna eriştiği bir sistem çok daha yüksek performanslı çalışabilecektir. İşleyiş tasarım bölümünde detayları ile verilecektir.

5. TASARIM

Tezin bu bölümünde, önceki bölümlerde verilen temel fikirlerden yola çıkılarak, güvenlik konusunda belirtilen açıkların önüne geçmeye yönelik bir sistem tasarlanmaya çalışılacaktır. Her mühendislik çalışmasında olduğu gibi tasarımı etkileyen temel faktörler performans, esneklik, yeniden kullanılabilirlik, sağlamlık ve tabii ki maliyet olacaktır.

5.1 Akıllı Kart

Bu tezde akıllı kartlar kullanılarak bir güvenlik sistemi tasarlanması hedeflendiği üzere, sistemin belkemiğini oluşturan akıllı kart özelliklerinin seçimi önemli bir rol üstlenmektedir.

5.1.1 Akıllı kart seçimi

Kartın seçiminde dikkat edilmesi gereken ilk nokta sağladığı güvenlik koşullarıdır. Bu noktada ihtiyaç duyacağımız şifreleme algoritmasına karar vermek gerekliliği baş göstermektedir.

5.1.1.1 Şifreleme-güvenlik yetenekleri

İncelenen şifreleme algoritmalarını göre kullanabileceğimiz iki temel algoritma var. İlki nispeten küçük anahtarlarla kullanılan, ek işlem yükü nispeten az ancak anahtarların her noktada bilinmesini gerektiren DES algoritması, diğeri ise büyük anahtarlar ile güvenilirliği belirgin bir şekilde artan, işlem yükü çok fazla ve anahtarların her uç birimce bilinmesini gerektirmeyen RSA algoritması.

RSA algoritması, gerçek zamanlı sistemlere yakın bir sistem üretmek için çok uygun bir algoritma değildir. Şifrelenecek verinin büyüklüğü arttıkça getirdiği iş yükü doğrusal olmayan bir şekilde artmaktadır. Sakıncalı diğer bir nokta ise RSA algoritmasının sahip

olması gerektiği büyük anahtarlardır. Bu özellikle birden fazla anahtara sahip olması planlanan bir kart için giderek problem çıkartmaya başlayacaktır. Eğer periyodik olarak anahtarların güncellenmesi planlanıyor ise büyük anahtarları korumak ve değiştirmek zaman ve iş yükü açısından pahalıya mal olacak bir durumdur. RSA algoritmasının iş yükünü fazlaca arttırması ve bir çok işlemin kartın içerisinde kalması gerekliliği RSA koşan bir kartın performansını olumsuz yönde etkileyecektir. Sonuç itibarı ile eğer ağa çıkacak tüm veriler RSA algoritması ile kart tarafından şifrelenecek olur ise ağ performansı çok düşük değerlerde olacak, paket tekrarları istenmeyen oranlarda artmaya başlayacaktır. Ağa çıkan verilerin şifrelenmesinde RSA kullanılması planlansa bile bunu kart yerine daha güçlü bir sistem olan bilgisayarın yapması daha doğru olacaktır. Bu noktada gizliği sağlayacak önemli nokta kullanılacak oturum anahtarlarının kart tarafından üretilmesi olacaktır. RSA konusunda söylenebilecek son nokta bu algoritmaya sahip olan kartların DES koşturabilen kartlara oranlar 10 kat mertebesinde pahalı oluşlarıdır.

DES algoritmasında beğenilmeyen iki özellik bulunmakta, bunlar verinin 64 bitlik bloklar halinde kullanılması ve anahtarların tüm uç birimlerce bilinmesi gerekliliğidir. CFM metodu ile verinin 64 bitlik bloklar halinde iletilmesi problemine bir çözüm getirmek mümkündür. Anahtarların her uç nokta tarafından bilinmesi problemini akıllı kartlar ile kolaylıkla aşabilir.Konuya tasarım bölümünün ileri safhalarında ayrıntıları ile değinilecektir.

Böylelikle kullanılması düşünülen akıllı kartın temel şifreleme özellikleri belirlenmiş olmuştur. DES işlemini yürütebilen, bu sayede sahip olduğu anahtar kümesinden yeni anahtarlar üretebilen bir kart, performans, güvenlik ve maliyet açısından en uygun kriterleri sunmaktadır.

5.1.1.2 Akıllı kart güvenliği ve esnekliği

Seçilecek olan kart, her şeyden önce hiç bir koşul altında anahtar kümesini dış dünyaya aktarmayacak, ancak bu anahtarların belirli şartlar altında kullanılmasına izin verecek yapıda olmalıdır. Ek olarak, kart üzerinde değişik görevlerde dosyalar kullanılacaktır. Bu dosyalarda kişiye özel bilgiler bulunabileceği gibi, herkes tarafından açıkça görülebilecek bilgiler de bulunabilecektir. Bu nedenle, kullanılacak kart esnek bir erişim denetleme

mantığına sahip olmalıdır. Kart üzerindeki değişik uygulamalar ve bunlara bağlı ayrı dosyalara, bağımsız olarak erişim şartları tanımlandırılabilmelidir. Ancak açık platform desteğindeki kartları tercih etmek maliyet açısından pahalı olacak ve yeni olmalarından kaynaklanan hataları da beraberinde getirecektir. Bu nedenle, bağımsız yetkilendirme özelliklerine sahip, dosya yaratma ve biçimlendirmeye işleyiş esnasında da izin verebilen bir kartta çoklu uygulama desteğini sağlamaya yönelik bir yapı oluşturulmaya çalışılacaktır.

5.1.1.3 Akıllı kart veri saklama kabiliyeti

Seçilen kart uygulama gereksinimleri de göz önünde bulundurularak, yeterli veri depolama kapasitesine sahip olmalıdır. Tasarlamayı düşündüğümüz sistem bir "açık sistem" tasarlanacağına göre eklenebilecek yeni uygulamalar – ki bunlara kart üzerinde bulunan "servisler" denilebilir, göz önüne alınarak kart kapasitesi mevcut servislerin ihtiyaçlarından daha fazla olarak düşünülmelidir. Fazlalık kısmın miktarı eklenmesi öngörülen servislere göre değişecektir.

5.1.1.4 Akıllı kart iletişim tipi

Önceki bölümlerde belirtildiği üzere akıllı kartları kontaklı ve kontaksız olarak iki ana bölüme ayırmak mümkündü. Kontaksız akıllı kartlar maliyet bakımından, kontaksızlara oranla maliyet açısından çok daha uygundur.Zira kontaksız bir kartın okuyucusu da kontaklı bir kartınkine oranla çok daha masraflıdır. Tasarlayacağımız sistemde kullanılacak akıllı kart kapalı ortamlarda kullanılacağı üzere kontaklı bir kart yeterli olacaktır. Oysa kullanılabilecek bir hibrit kart, kart ile yapılması planlanan diğer işlemlerde büyük kolaylıklar sağlayabilecektir (örneğin bir kapının açılması). Yine de maliyet ölçütleri de göz önüne alınarak kontaklı bir kart tercih edilecektir.

Bahsi geçen özellikleri sağlaması bakımından projede Schlumberger Payflex Mk5 2K SAM[17] kartları kullanılacaktır. Bu kartın seçilmesindeki temel neden, tezi yazanın kart üzerindeki tecrübesi ve temin kolaylığıdır[17]. Bu akıllı kart, T0 protokollü, kontaklı ve belirtilen kriterleri sağlayabilen bir akıllı karttır.

5.1.2 Akıllı kart dosya yapısı

Güvenliği sağlayacak kartın seçimi kadar kart üzerinde oluşturulacak dosya sistemi de güvenlik sorununun çözülmesinde ayırım noktalarından biridir. Hazırlanacak dosya sistemi, tez içerisinde savunulan güvenlik, esneklik ve sağlamlık kriterlerini sağlamak zorundadır. Bu kriterlerin gerekliliklerini gözden geçirecek olursak sağlanması gerekli özellikler şu şekilde özetlenebilir:

- Anahtarların gizli tutulması ve hiç bir koşul altında dış ortama aktarılmaması,
- Çoklu uygulama ve uygulama eklenebilme özelliği,
- Her uygulamaya has ve birbirinden bağımsız anahtar kümesi,
- Özel bilgileri korunmasına yönelik güçlü ve sağlam bir erişim sınırlandırma mantığı,
- Performansı ve kullanımı kolaylaştırıcı dosya ağaç yapısı,
- Kart üzerinde bulunan uygulamaların tanınmasını sağlayacak bir kart tanıtma sistemi.

5.1.2.1 Çoklu uygulama desteği ve akıllı kart uygulama tanıtım metodu

Nasıl her sabit sürücü kullanılmaya başlanmadan önce bir biçimlemeye ihtiyaç duyuyorsa, her akıllı kart da ilk kullanılma aşamasına geçmeden önce bir biçimlendirmeye ihtiyaç duyar. Bu aşamaya "personalizasyon" aşaması adı verilir. Personalizasyon esnasında kartın temel dosya sistemi oluşturulur, bu dosyalara ait erişim hakları belirlenir ve tanımlanan asgari anahtar kümesi karta yüklenir. Bu safhanın güvenliği kesin sağlanmak zorundadır. Özellikle anahtar yükleme işleminin gizliliği önemlidir.

Personalizasyon aşamasında kartın MF'i altında bir adet anahtar kümesi seti oluşturulma zorunluluğu vardır. Bu dosyaya "Yönetimsel Anahtar Kümesi" adı verilebilir. Her akıllı kart fabrika çıkışında personalizasyon aşamasında kullanılmak üzere bir ana anahtar (Transport Key) ile kodlanır. Bu anahtar kart üzerinde dosya yaratmak için doğrulanmak zorundadır. "Yönetimsel Anahtar Kümesi"'ni oluşturmaktaki amaç, ana anahtara ihtiyaç

kalmadan diğer dosya işlemlerini yürütebilmektir. Bu dosya yaratıldıktan sonra dosya işlemleri ile ilgili haklar bu anahtar kümesine atanacaktır.

Dosya sistemi oluşturma safhasında ise kartın MF'inde her uygulamaya ait bir DF oluşturulmalıdır. MF'in altında oluşturulacak son dosya ise okuyucu programa kartın uygulamalarını tanıtacak AID dosyasıdır.

AID dosyasının yapılandırılmasında ağaç modeli kullanılacaktır. Buna göre tüm AID dosyaları bağlı bulundukları uygulamalara ait tüm alt uygulamalara ait bilgilere işaretçiler içerecek, yaprak konumunda bulunan AID'ler ise bağlı bulundukları uygulama ya da alt uygulamalara ait dosyalara ilişkin bilgileri tutacaktır. MF altındaki AID dosyası da doğal olarak kart üzerinde bulunan tüm uygulamaların AID dosyalarına birer işaretçi içerecektir. Okunacak bir AID dosyasının bir dala ya da bir yaprağa ait olması durumunu ayırt etmek için AID dosyası kayıt yapısında bir belirteç tutulacaktır. İlk kaydın bir dal olduğunu gösterir işaretçi içermesi müteakip kayıtların da birer dal olduğunu, tersi ise müteakip kayıtların birer dosya belirteci olacaklarını bildirecektir. Bu belirteci bir sekizlik veri olarak düşünülürse geri kalan 7 bit ile de uygulama tipi belli edilecektir.Uygulama tipleri Tablo 5.1. de belirtilmiştir. AID kayıt yapısındaki diğer bir alan dosyanın tipini belirtmektedir. Dosya tipleri Tablo 5.2 de gösterilmiştir. AID kaydındaki son alan ise bildirilen uygulamanın aktif olup olmadığını gösterecektir.Her seviyedeki AID dosyasının belirteci aynı olacaktır Kullanılacak AID yapısı Şekil 5.1'de gösterilmiştir.

Uygulama Adı	**Tip Bilgisi**
Güvenlik Uygulaması –Bağlanma	000 0001
Güvenlik Uygulaması – Ağ Güvenliği	000 0002
Kişisel Bilgiler Uygulaması – Personel Bilgileri	001 0001
Kişisel Bilgiler Uygulaması – Acil Bilgiler	001 0002
E-Cüzdan Uygulaması – E-Cüzdan	002 0001

Tablo 5.1 AID kayıtlarındaki uygulama tipleri.

Dosya Tipi	**Tip Bilgisi**
Klasör (DF)	00
PIN Dosyası	01
AID Dosyası	02
Anahtar Küme Dosyası	03
Bilgi Dosyası	04
Elektronik Cüzdan Dosyası	05

Tablo 5.2 AID kayıtlarındaki dosya tipleri.



a) Dal Konumunda AID yapısı



b) Yaprak Konumunda AID yapısı

Şekil 5.1 Kullanılacak AID yapısı

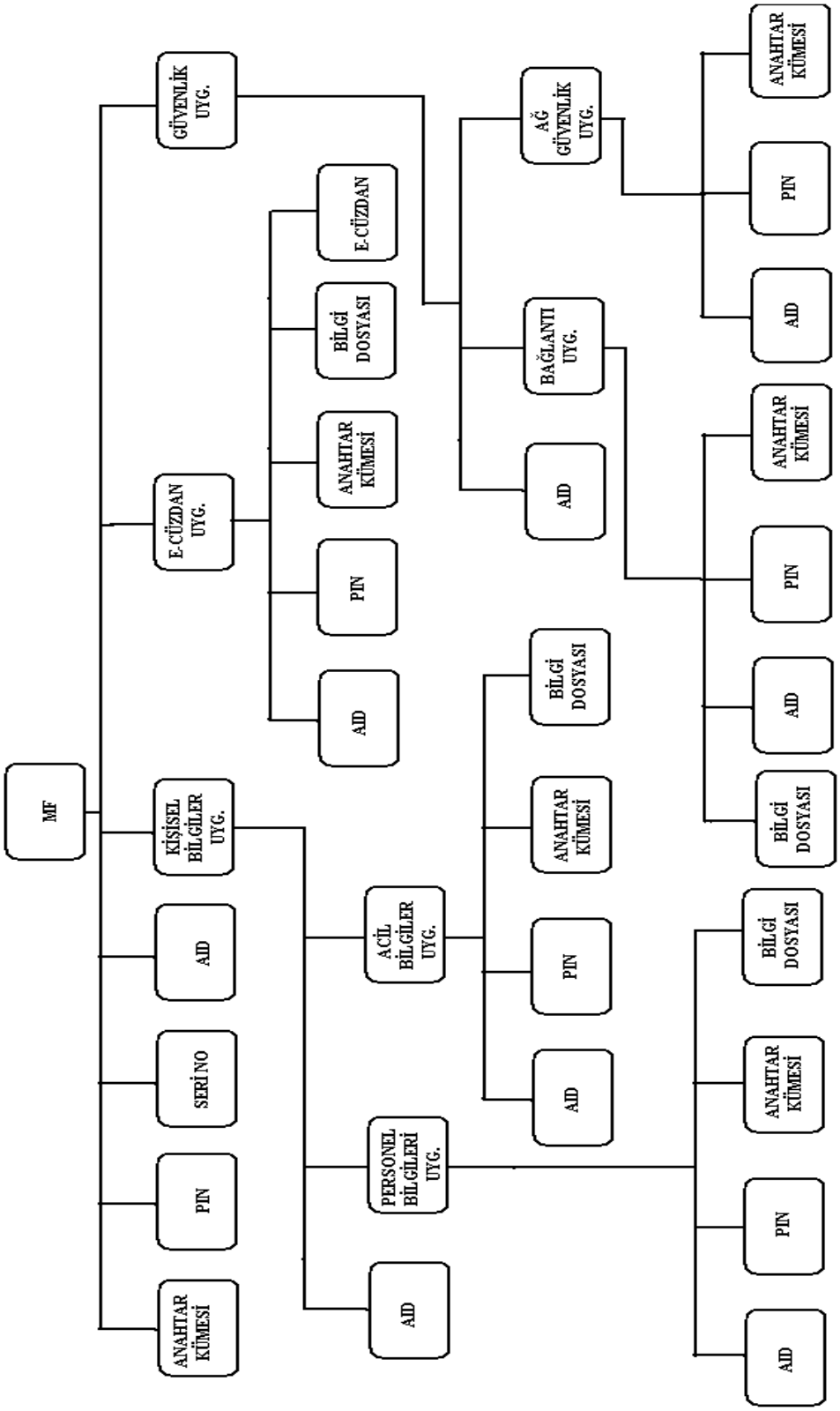
Kartın üzerindeki dosyalar yaratılırken bir azami kayıt sayısı belirlenmesi gerekliliğinden dolayı her AID dosyası 5 adet uygulamaya işaret edecek şekilde oluşturulacaktır. Bu sayı kartın kapasitesine göre arttırılabilir ancak projede 5 adet ana uygulama ve buna bağlı uygulamalar dizisi fazlası ile yeterli olacaktır.

5.1.2.2 Uygulama dosya yapısı

Her uygulama DF'i tanımlanan AID yapısına uyması açısından bir AID dosyası içermek zorundadır. Uygulamanın yaprak durumunda olması durumunda ise her uygulama bir parametre dosyası, bir anahtar kümesi dosyası , bir PIN dosyası ve gerekli olduğu taktirde bir e-cüzdan dosyasına sahip olmak durumundadır. Tüm dosyalar bağımsız olarak uygulamanın sahip olduğu anahtar kümesine göre erişim haklarına sahip olmalıdır ve bu erişim hakları dosyalar yaratılırken belirlenir. Projede kart üzerinde olması düşünülen uygulamalar :

- Kişisel bilgiler uygulaması,
    - Çalışan bilgileri uygulaması,
    - Acil durum bilgileri uygulaması,
- Güvenlik Uygulaması,
    - İşletim sistemi ve ağ bağlanma uygulaması,
    - Ağ güvenlik uygulaması,
- E-Cüzdan Uygulaması.

Projede kullanılması planlanan kartın kısıtlamaları nedeni ile iç içe en fazla 3 seviyeye izin verilmiştir[17], bu nedenle dosya yapısı derin bir ağaç yerine yayvan bir ağaç formunda tasarlanmıştır.Düşünülen uygulamalara ilişkin oluşturulması gereken kart yapılandırması Şekil 5.2 de gösterilmiştir.



Şekil 5.2 Akıllı kartta oluşturulacak dosya yapısı.

5.1.2.3. Dosya erişim hakları ve anahtar küme dosyaları yapısı

Kart üzerindeki tüm uygulamaların ve bağlı bulunan dosyaların güvenlik gereksinimleri farklı olacaktır. Bu dosyalardan okuma, yazma, değiştirilme erişim denetimleri bağlı bulunacakları uygulamanın anahtar kümesi ile ilgili olacaktır. Bu nedenle gerek uç birimlerde, gerek kullanılabilecek ek cihazlarda koşan yazılımların dosyalara erişmek için sağlanması gerekli koşulları önceden bilmek durumundadırlar. Silme işlemi için izin tanımlama ise kullanılan kartın özellikleri bakımından sadece bağlı bulunulan DF'e verilebilir. Bunun anlamı bir DF'e verilen silme hak tanımı altındaki bütün elementer dosyalar için geçerlidir. Kullanılan kartın diğer bir sınırlaması ile dosyaların yaratılma sırası ile silinme sırası ters olmak zorundadır. Yani bir DF altında en son yaratılan dosya silinmeden, başka bir dosyayı silmek mümkün değildir[17].

Akıllı kartlarda bir dosya okunmak istendiği zaman "SELECT" komutu aracılığı ile önce dosya seçilmelidir. Bu komuta kart herhangi bir sınırlandırma getirmez. Yani kart yapısını ve komut setini bilen bir kişi istediği dosyayı seçebilir. Ancak erişim kıstasları sağlanmadıkça bu dosyaları seçmek haricinde bir işlem uygulamak mümkün olmayacaktır.

Tüm anahtar küme dosyaları hiç bir zaman okunamaz şeklinde ayarlanmalıdır. Bu şekilde anahtarların azami düzeyde korunmuş olması sağlanacaktır.Ayrıca tüm anahtarlara yönetici tarafından belirlenecek sayıda deneme hakkı tanınmalıdır.Örneğin bu sayı 5 olarak belirlenmiş ise 5 hatalı denemeden sonra anahtar bloke olur ve tekrar kullanılamaz.Bu kartın kullanılamayacağı anlamına gelmez, sadece kart uygulaması kısmen kullanılamaz (tüm anahtarları bloke olursa uygulama artık fonksiyonel değildir). Bu sayede sisteme olan saldırıların önlenmesi, pahalıya patlasa bile, önlenebilecektir. PIN dosyasının anahtarı ise bloke olduktan sonra bir PUK anahtarı ile açılabilir, bu bilgi sadece sistem yöneticisinde olmalı ve gerektiğinde PIN sadece onun tarafından açılmalıdır. PUK anahtarı da belli bir deneme sayısından sonra kilitlenir, bu durumda da kart uygulaması kullanılamaz hale gelir.

Tasarlanan kart yapısında dosya erişimleri şöyle belirlenmiştir :

- AID dosyaları eşdeğer seviyede bulundukları anahtar küme dosyasının ilgili anahtarı doğrulanarak yazılabilir. Okuma işlemi için bir doğrulama gerekmez,
- SERİ NO dosyası hiç bir zaman yazılamaz, ancak eşit seviyedeki PIN doğrulanarak okunabilir,
- Hiç bir anahtar kümesi okunamaz ancak kendi üzerinde tanımlanan bir anahtarın doğrulanması ile yeni anahtar eklemeye izin verir,
- E-Cüzdan dosyası özel bir dosyadır ve kredilendirme ya da bakiye düşme işlemlerini otomatik olarak yapabilir. İlgili komutlar için eşdeğer seviyedeki anahtar kümesindeki, kredilendirme ya da bakiye düşme anahtarı doğrulanmalıdır,
- Bilgi dosyalarının erişimi tezi yorumlayana bırakılmıştır. Ancak, "ağ bağlanma" uygulaması altındaki bilgi dosyası kullanıcı adı ve şifresini içereceğinden en azından PIN ile korunması daha mantıklı olacaktır.

Bu şekilde bir tasarım ile "open platform" yapısında olmayan bir akıllı kartın çoklu uygulama desteği belirli kısıtlamalar ile sağlanmış bulunmaktadır. Bu kısıtlamaların temeli şifreleme algoritmalarına dayanmaktadır. Kullanılan kart DES işlemlerine izin verdiği için herhangi bir uygulama için RSA yöntemi kullanılamaz. Ancak uygulamaların kullandığı şifreleme yöntemini çeşitlendirmek mümkündür. Kullanılan akılı kart iki adet şifreleme çeşitleme bayrağına dolayısı ile dört farklı yönteme sahiptir. Bu bayraklar :

- DES – TDES mod bayrağı,
- Normal şifreleme - S-Mode şifreleme bayrağıdır.

Uygulamaların şifreleme özellikleri bu kısıtlamalar altında çeşitlendirilebilir[17].

5.1.3 Okuyucu seçimi

Sistemde akıllı kart kullanılacağına göre uygun bir akıllı kart okuyucu da seçilmelidir. Mevcut standartlar çerçevesinde en yaygın olarak desteklenen PC/SC uyumlu bir akıllı kart okuyucusu yeterli olacaktır. Bu tip okuyucuların avantajı, T0 ve T1 uyumlu kartları

desteklemeleri, yazılım ile okuyucunun sürücülerinin güncellenebilmesidir. Böylelikle ileri teknolojilerin desteklenme olasılığı ve terfi olanağı hayli arttırılmış olacaktır. Önemli bir nokta da okuyucunun kullanacağı arabirim olacaktır. Günümüz bilgisayarlarında seri iletişim birimleri artık nadiren kullanılır olmuştur. Seri arabirime sahip okuyucular genellikle güç gereksinimlerini fare ya da klavye bağlantı noktasından sağlamaktadır. Bu ise bazı donanımlar için sorun çıkarabilecek bir yöntemdir ve kişisel tecrübelerle sabittir. Güç gereksinimleri ve seri arabirimin neden olduğu performans kayıplarının önüne geçilmesi ve "tak çalıştır" özelliğinden azami şekilde faydalanılabilmesi için USB arabirimine sahip bir okuyucunun kullanılması daha mantıklı görünmektedir. Teknolojideki ilerlemeler sayesinde artık USB arabirime sahip okuyucular ile seri arabirime sahip okuyucular arasındaki fiyat farkı makul ölçütlere inmiştir.

Kullanılacak kart kontaklı bir kart olacağı üzere, okuyucunun da kontaklı kartları destekleyen bir özellikte olması yeterli olacaktır.

5.2 Sunucu Güvenlik Sistemi (HSM – Host Security Module)

Tasarlanan güvenlik sisteminde, tüm uçbirimlerin ağa bağlanmasını sağlayacak bir sunucuya ihtiyaç olacaktır.Bu sunucunu görevleri:

- Ağa bağlanmaya çalışan uç terminali doğrulamak,
- Kilitli durumda ya da kullanıcısı başında olmayan bir uç birime ulaşılmaya çalışılması durumunda, istekte bulunan uç birimi doğrulayıp, aralarında kullanılacak anahtarı belirlemek,
- Ağda yakalanan yetkisiz bir müdahaleyi saptayıp, adresini ve ilgili bilgileri ağ yöneticisine raporlamak ve ilgili uç birimi yönetici izin verene kadar bloke etmek üzere diğer uç birimlere yayın yapmak,
- Sunucu üzerinde bulunan dosyaları isteğe bağlı olarak anahtarı ve yöntemi sadece HSM modülünce bilinecek bir şekilde şifrelemek. Ve isteğe bağlı olarak bu dosyaların şifresini açmak olacaktır[18,19].

HSM sisteminde de karşılıklı doğrulamayı yapacak ve oturum anahtarlarını oluşturacak bir akıllı kart ve gerekli iletişim donanım kullanılacaktır.

5.2.1 HSM akıllı kartı

Güvenlik sunucusunda da uç birimlerde olduğu gibi bir akıllı kart kullanılacaktır. Ancak bu görevi nedeni ile sadece anahtar kümesini kullanmakla kalmayıp istekte bulunan istemcilere yeni anahtarlar üretebilecek kapasitede olması gerekir. Bu tip kartlara "FULL DES SAM" adı verilir. FULL DES SAM kartlar, sıradan akıllı kartların bellek kapasiteleri ve şifreleme özellikleri arttırılmış versiyonları olmasından başka bir farkı bulunmamaktadır. Sıradan akıllı kartlar anahtarlarını korurken, bir şekilde türetilmiş anahtarları da (derived key) dış dünyaya vermez, ancak türetilmiş anahtarlar kullanılarak doğrulama yapılmasına izin verir. Karta gönderilen bir doğrulama (authentication) komutu cevabı normal bir akıllı kartta sadece doğrulandı ve ya doğrulanmadı şeklinde olurken, SAM kartlar bu komuta doğrulanması durumunda türetilmiş anahtardan oluşturulan bir sayı döndürür. FULL DES SAM kartlar ise türetilmiş anahtarın üretilmesine izin verir, ve tüm sistemde yer alacak kartların anahtarları bu kart kullanılarak belirlenecektir. Bu özelliğin oturum anahtarlarının oluşturulma aşamasında kullanılması zorunluluğundan HSM kartı olarak bu tür bir kart kullanılması gereklidir. Ancak bu seçim, daha önce belirlenen kart seçimine etki etmeyecektir. Zira gerek FULL DES SAM gerekse de uç birim akıllı kartları (SAM) temel özellikler olarak aynıdır. Mevcut koşullar bu tür varyasyonları desteklemekte olduğundan sıkıntı çekilmeyecektir.

HSM kartının dosya yapısı uç birimlerde kullanılacak kartınki ile benzer olacaktır. Ancak bu kart sadece ağ için kullanılacağından üzerinde sadece güvenlik uygulamasının bulunması yeterli olacaktır.

5.2.2 HSM iletişim donanım ve yazılımı

HSM, sistemde kullanılacak tüm anahtarları barındıracak bir anahtar değiştirme-dağıtım sunucusu gibi davranacağı göz önünde bulundurulursa, sıradan bir bilgisayar kullanılmasının sakıncalı olacağı rahatlıkla görülebilir. Zira anahtarların oluşturulma algoritmaları, yeni anahtarlar için giriş ve ara çıkışlar kesinlikle gizli tutulmalıdır. Bu nedenle, HSM yazılımını bir masaüstü ya da sunucu bilgisayarda çalıştırmak ve akıllı kart ile bağlantıyı bir kart okuyucu aracılığı ile yapmak yerine, sadece ağdan gelen istekleri alıp bu isteklerdeki temel türetme bilgilerini (girişlerini) HSM donanımına aktaracak ve nihai çıkışı da paketleyip ağda istemde bulunan uç birime gönderecek "Ağ

İletişim Arabirimi"ni uç birimde bırakmak daha mantıklı olacaktır. HSM sisteminin iç işlemlerini yürütmek için daha güvenli bir cihaz kullanılabilir[16].Örneğin bir POS terminali bu iş için kullanılabilir. Modern bir POS barındırdığı bilgileri ve kaynak kodunu dış dünyadan saklı tutar ve kasasının açılması halinde içeriğini silecek şekilde ayarlanabilir.

HSM modülünün sunucu üzerinde koşan ağ iletişim arabirimi, TCP/IP protokolünü destekleyecek şekilde çalışacaktır. Şu an için başka bir protokol ve buna bağlı farklılıkları içeren destek düşünülmemektedir. Ağ iletişim arabirimi açık bir TCP/IP portundan gelen istekleri karşılayabilecektir. Bu portlardan kendisine gelen ağ bağlanma, ya da başka bir uç birime bağlantı için geçici anahtar isteklerine cevap vermek üzere sonsuz bir döngüde dönecek şekilde tasarlanacaktır. HSM modülü aynı zamanda uç birimlerde çalışacak TSM modülünün de tüm görevlerini yapmak durumundadır, bu sayede ağa Kerberos sisteminde olduğu gibi ayrı bir anahtar dağıtıcı ya da doğrulayıcı sunucu kurulmasına gerek kalmayacaktır.Ancak bir anda birden fazla istek gelebilmesi olasılığına karşı, program aynı zamanda proses mantığında olacaktır.

5.3 Uç Birim Güvenlik Modülü (TSM – Terminal Security Module)

Uç birim güvenliği için belirlenen SAM cinsi akıllı kartlar kullanılacaktır. Ancak oturum açma esnasında HSM modülü ile yalnızca türetilmiş anahtarlar aracılığı ile doğrulama yapılacağından kart ile ilgili işlemlerin ayrı bir cihazca yapılmasına gerek bulunmamaktadır. Bu nedenle ağ ile bağlantı kuran ve kart ile ilgili işlemleri yürüten yazılımımız uç birimde koşan bir yazılım olacaktır. Bu modülün görevleri:

- Uç birime bağlanmak isteyen kullanıcıyı akıllı kart aracılığı ile doğrulamak,
- Kullanıcının geçerliğini akıllı kartı sürekli denetleyerek sağlamak, kullanıcı uç birimden ayrıldığı anda belirli bir süre sonra uç birimi kilitlemek ya da kapatmak,
- Kullanıcı talebi üzerine kullanıcının PIN bilgisini değiştirmek,
- Başka bir uç birim (sunucu da dahil) ile kurulacak iletişim esnasında verilerin şifrelenmesi ve çözülmesini sağlamak,

- Sıradan erişimler için paylaştırılmamış ancak güvenli bağlantı için paylaştırılmış dosyaların transferini sağlamak,
- Uç birim üzerinde bulunan dosyaları isteğe bağlı olarak, anahtarı ve yöntemi sadece TSM tarafından bilinecek şekilde şifrelemek ve arzu edildiğinde şifrelemeyi kaldırmaktır.

TSM, kullanıcıların uç birime ve dolayısı ile ağa bağlanmasını sağlarken bir akıllı kart kullanacağına göre, işletim sisteminin standart bağlanma arayüzünü devre dışı bırakmak durumundadır. Yeni nesil işletim sistemlerinde kullanıcı bağlama (logon) sağlayan modüller çekirdek (kernel) ile yakın ilişkide bulunduğundan dolayı bu tür modülleri devre dışı bırakmak işletim sisteminin açılış ve çalışma esnasında problemlere neden olacaktır. O halde TSM'in yapması gereken bir şekilde bu bağlanma arayüzünün önüne geçerek ancak doğrulama sağlandıktan sonra işletim sisteminin bağlanma arayüzünü çalıştırmaktır.

"Microsoft Windows" tabanlı sistemleri örnek alınacak olursa, bu tür sistemler bağlanma arayüzü olarak "LogonUI" ismi verilen bir API kullanır. LogonUI API'si kullanıcı adı ve şifresini bir görsel metin penceresi vasıtası ile aldıktan sonra bu bilgileri kullanarak işletim sistemini başlatmak için "userinit.exe" programını başlatır. TSM'in yapması gereken LogonUI API'sinde gereken değişiklikleri yaparak "userinit.exe" programı yerine kart doğrulamasını yapan programı çağırmalı, doğrulama gerçekleşir ise "userinit.exe" programına karttan okuduğu kullanıcı adı ve şifre ikilisini geçirmelidir. Böylelikle, güvenli bir bağlantı yanında kullanıcı adı ve şifresini de gizlenmiş olunacaktır[20].

TSM modülü de HSM modülünde olduğu gibi TCP/IP protokolünü destekleyecektir ve şu an için başka bir protokolü desteklemesi düşünülmemektedir. Program, aynı anda birden fazla bağlantının yapılabilmesi için proses mantığında tasarlanacaktır.

İki uç birim arasındaki güvenli mesajlaşma esnasında verilerin şifrelenmesi ve çözülmesi TSM'in görevidir. Verilerin şifrelenmesi, kart tarafından da gerçekleştirilebilir ancak bu performansı olumsuz yönde düşüreceği için şifreleme işleminin uç birimde koşan TSM tarafından yapılması daha uygun görülmüştür. Şifreleme esnasında kullanılacak anahtar kesinlikle, uygulama içinde tutulmayacak ve şifreleme işlemi bittiği anda bellekten

silinecek şekilde bir çalışma mantığı izlenecektir. Buna tek bir istisna, TSM'in sunucuya erişememe durumunda kullanacağı "yarı erişim anahtarı"'dır. Bu anahtar her kart için farklı olacak şekilde üretilecektir.

TSM modülleri birbirleri ile konuşurken aynı anahtarları kullanma zorunluluğu yoktur. Bir anahtar ile şifrelenmiş mesaja yanıt başka bir anahtar ile şifrelenmiş olarak gönderilecektir. Art arda gönderilmesi gereken mesajlarda da, anahtar kümesi yettiği müddetçe mümkün olan farklı anahtarlar kullanılarak, hatta iki uç birim arasında mümkün olduğunca farklı anahtarla şifrelenmiş mesajların dolaşması sağlanacaktır.

Şüphesiz ki ağdaki her paketin şifrelenmesi ağ performansını düşürücü bir rol oynayacaktır. Bu neden ile güvenli bağlantı isteği TSM programında kullanıcı tarafından belirtilecek ve kullanıcı tarafından sonlandırılıncaya dek belirlenen uç birimle mesajlaşma şifrelenecektir.

5.4 Sistemin Çalışma Şekli

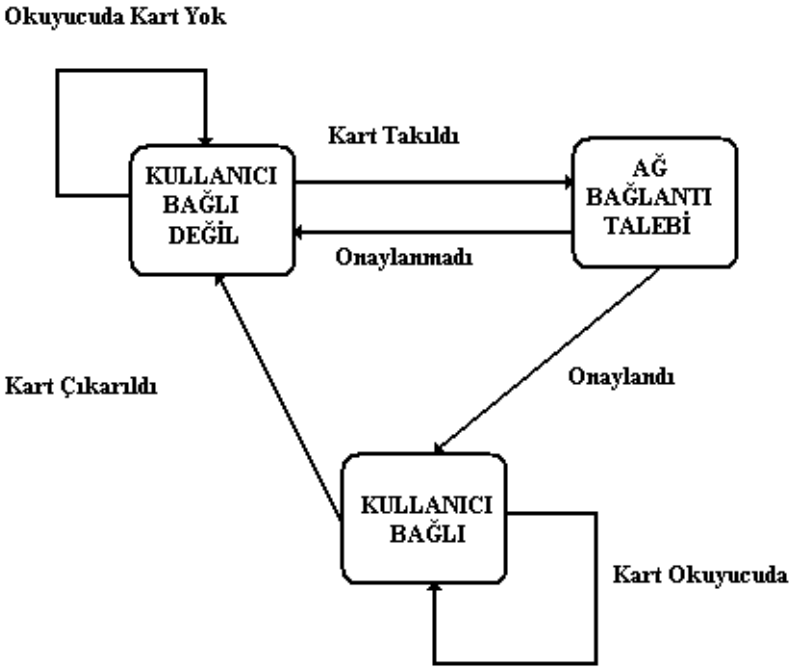
Sistemin çalışmasını ağa bağlantı ve uç birimler arası haberleşme şeklinde iki kısma ayırmak mümkündür.

5.4.1 Ağa bağlantı

Ağa bağlanmak asıl itibarı ile bir kullanıcının uç birimi açması ile gerçekleşir ancak her zaman ağa bağlantıyı sağlayan sunucunun dolayısı ile HSM sisteminin çalışması garanti edilemez. Bu gibi durumlarda da kullanıcıya en azından kendi uç birimini kullanabileceği şekilde bir yöntem düşünülmek zorunluluğu bulunmaktadır.

5.4.1.1 Akıllı karttan kullanıcı bilgilerinin alınması

Uç birim kullanıcısı sistemini açtığı anda, klasik bağlanma ekranı yerine, akıllı kartını takması yönünde bir mesaj alacaktır. Akıllı kart takılması ile beraber TSM modülü, kullanıcıdan PIN kodunu girmesini isteyecektir. PIN kodunun doğrulanmasının ardından kart üzerindeki, kullanıcı ve parola bilgileri alındıktan sonra, güvenli mesaj protokolü kullanılarak sunucudan bağlanma isteği talep edilecektir. Şekil 5.3'te kullanıcı bilgilerinin alınmasına dair işlemin durum diyagramı gösterilmektedir.



Şekil 5.3 Bağlanma işlemi durum diyagramı.

5.4.1.2 Ağa bağlantı sağlanması

Sunucunun çalışır durumda olması halinde, sunucu HSM modülü ile TSM modülü karşılıklı doğrulama (Cross Authentication) yapacaklar ve işlemin başarı ile bitmesi durumunda kullanıcı yerel işletim sistemi ve ağa bağlanmış olacaktır. TSM'in sunucuya erişememesi durumunda yarı erişim anahtarını kullanarak kullanıcıya sadece uç terminalini açacak ancak hiç bir ağ bağlantısına izin vermeyecektir. Kullanıcı istediği zaman TSM'i kullanarak sunucudan ağ bağlantısı talep edebilecektir.

Her iki durumda da TSM uç birim açık kaldığı müddetçe kullanıcı kartının takılı olduğunu kontrol edecek, kartın çıkarılması ile beraber, ekrana kullanıcıyı uyarıcı bir mesaj çıkartacaktır. Makul bir süre sonunda kartın tekrar takılmaması halinde, işletim sisteminin izin verdiği ölçüde uç birimi kilitleyecek, kullanıcı bağlantısını kesecek

(logoff) ya da terminali kapatacaktır. Kilitli bir terminalin açılması ancak, doğru kullanıcı kartının tekrar takılması ile mümkün olacaktır. Bu tür durumlarda TSM, beklediği kartın özellikleri ile takılan kartın özelliklerini karşılaştıracak ve kartın değişip değişmediğini kontrol edecektir. Kartın tekrar takılması ile bağlanma işlemleri yeniden başlayacaktır. Terminal kilitli konumda ise ve bağlanma olumsuz sonuçlanmış ise mevcut kullanıcının derhal bağlantısı kesilecektir.Kilitli durumdaki terminal belli bir zaman aşımı sonunda kapatılacaktır. Uç birime, yetkili başka bir kullanıcı kartı takılması durumunda ise eski kullanıcının işletim sistemi ile bağlantısı koparılarak yeni kullanıcı bağlanma işlemine tabi tutulacaktır.

Ağa bağlanmak için TSM ve HSM kartlardaki "işletim sistemi ve ağ güvenlik uygulaması"'na ait anahtar küme dosyasındaki anahtarları kullanarak birbirlerini doğrularlar. Her doğrulama işleminde gerek HSM gerek de TSM bu anahtar kümesinden rasgele birer anahtar seçer ve karşı tarafa seçtiği anahtarı bildirir[21-23].Bu işleme ait daha açıklayıcı bilgi gerçekleme bölümünde verilecektir.

5.4.2 Uç birimler arası haberleşme

Başka bir uç birime bağlantı ise iki şekilde olabilir:

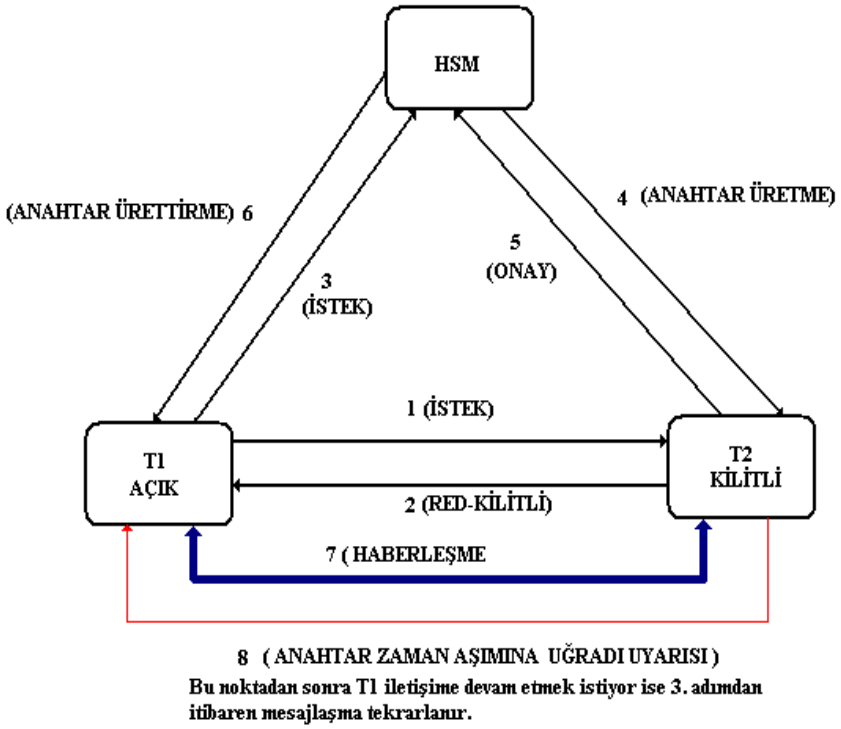
- Karşı uç birimin bir kullanıcı tarafından kullanılıyor -"açık"- olması,
- Karşı uç birimin kullanıcı kartının çıkarılmış - "kilitli"- olması.

5.4.2.1 Karşı uç birimin açık olması

Bu durumda iki uç birimin TSM modülü birbirini karşılıklı olarak doğrulayarak iletişime başlayabilirler. Karşılıklı doğrulama mantığı ağa bağlanmada kullanılan yöntem olacaktır. Doğrulama gerçekleştikten sonra, her paket gönderilirken ilgili TSM rasgele olarak kartın ağ güvenlik uygulaması anahtarlarından birini veri şifrelemek için seçer. Bu anahtardan, bir dizi dinamik veriyi de kullanarak yeni bir anahtar üretir ve göndereceği paketin başına kaçıncı anahtarı kullandığı ve kullandığı dinamik değişkenleri yerleştirir. Konu ayrıntılı olarak gerçekleme bölümünde anlatılacaktır.

5.4.2.2 Karşı uç birimin kilitli olması

İkinci durumda karşı uç birim kilitlenmiş olabilir, bu durumda karşı uç birimin TSM modülünün kullanabileceği bir kart yoktur. Karşı uç birim TSM'i durumu alıcı tarafa bildirir. İstekte bulunan uç birim ise bağlantı isteğini hangi uç birim olduğunu da belirterek HSM'e iletir. HSM, kilitli uç birime yarı erişim anahtarı ile çözebileceği bir dizi dinamik veriyi, bu yolla haberleşebileceği uç birimin adresini ve zaman sayacını yollar. Kilitli uç birim bu bilgileri kullanarak yeni bir anahtar üretir ve bundan sonra ilgili uç birim ile haberleşmek için bu türetilmiş anahtarı kullanır. Önemli olan nokta HSM'in bu bilgileri kullanırken, istemde bulunan TSM'in kartının anahtarlarını biliyor olmasıdır. Üretilecek anahtarın asıl anahtarı TSM'de bulunan akıllı kart üzerindeki anahtardır. Kilitli uç birim paketi aldığını ve işlemi gerçekleştirdiğini HSM'e bir cevap atarak bildirir, bağlantıyı belirli koşullara göre reddetmesi de mümkündür.Eğer bağlantı kabul edilmiş ise HSM, aynı anahtarı oluşturacak şekilde dinamik bilgiyi ve anahtar numarasını istemci uç birime bildirir. İletişimin esaslarına, kilitli uç birimin TSM modülü karar verir, dosya kopyalama, çalıştırma gibi haklar kilitli uç birimin kullanıcısı tarafından belirlenecektir. Kilitli uç birim TSM'i sürekli zaman sayacını kontrol eder, eğer zaman sayacı sıfırlanmış ise türetilen oturum anahtarı geçersizdir ve bir sonraki gelen isteğe, uç birimin kilitli olduğu mesajı geçilir. Bu durumda istemci terminal bahsedilen işlemleri tekrar başlatmak zorundadır. Zaman sayacının çok uzun tutulması oturum anahtarlarının güvenilirliğini azaltma yönünde, çok kısa tutulması ise ağ performansının düşmesi yönünde olumsuz etkiye neden olacaktır. Bu süre isteğe bağlı olarak ağ yöneticisi tarafından HSM üzerinde belirlenebilir. Sürenin sıfır olması bu tip bağlantının yasaklanması olacaktır. Kilitli uç birim açıldığı oturum anahtarını geçersizleştirecek ve gelen mesaja uç birimin açıldığı, karşılıklı doğrulama isteğinin gerektiğini belirten bir mesaj atacaktır. Basitleştirilmiş bir akış Şekil 5.4'da gösterilmiştir.



Şekil 5.4 Uç birimlerden birinin kilitli olması halinde haberleşme.

Her TSM bağlantı kurduğu uç birimi bir zaman bilgisi tutarak listesinde tutacak ve bu zaman aşımı ya da uç birimlerden birinin bağlantıyı kesme isteğine kadar tekrar karşılıklı doğrulama işlemine gerek olmayacaktır. Tutulacak zaman bilgisi sistem saatinden alınmayacak (tarih ile oynanması ihtimaline karşı), proses zamanı esas alınacaktır. Uç birimlerden birinin kilitli olması durumundaki zaman sayacı işlenmesi de bu şekilde olacaktır. Uç birim her açıldığında liste tamamen temizlenecektir. Bağlantı kurulduktan sonra, veriler şifrelenerek ilgili uç birime aktarılacaktır.

HSM ya da TSM'lerden herhangi biri doğrulama adımlarının bir safhasında üst üste iki defa anahtar hatasından doğan bir hata alırsa, eğer TSM ise HSM'i durumdan haberdar edecek ve istekte bulunan uç birimi kara listesine alacaktır. HSM böyle bir bilgiyi aldığı anda ilgili uç birimi kara listesine alır ve tüm ağa bir yayın ile bunu bildirir. Yayını alabilen tüm TSM'ler ilgili uç birimi kara listelerine dahil ederler. Bir uç birimin kara listeden çıkartılması ancak ağ yöneticisinin uç birimi HSM üzerinde kara listeden

çıkartması ile olur. Bu durumda HSM ağa uç birimin kara listeden çıkarılması yönünde bir yayın yapar. Eğer yönetici kara listeyi tamamen silerse, HSM'in yayını kara listenin silinmesi şeklinde olur. Ağ yöneticisi, şüpheli gördüğü bir durumda ağı bloke edebilir. Bu durumda HSM ağa "ağ bloke" yayınını yapar ve yayını alan tüm TSM'ler iletişimi durdurur ve "ağ açık" yayınını alana dek iletişime başlamazlar. Ağ bloke durumunda HSM hiç bir isteğe yanıt dönmez.

5.4.3 Uç birim dosyalarının güvenliği

TSM modülünün bir görevi de uç birim üzerinde yetkisiz kişilerce açılması istenmeyen dosyaların şifrelenmesidir. Her ne kadar yetkili bir akıllı kart olmadan sisteme bağlanmak mümkün olmasa da uç birimde bulunan veri saklama birimi sökülerek başka bir bilgisayarda kullanılabilir. Bu nedenle kullanıcını belirlediği dosyalar TSM tarafından şifrelenebilmeli ve çözülebilmelidir.

Kullanıcı talebi karşısında TSM mevcut akıllı karta rasgele bir anahtar numarası ve kartın seri numarası ile bir anahtar ürettirir ve bu şifreyi kullanarak ilgili dosyayı şifreler. Şifrelenmiş veriyi kaydederken kullanılan rasgele anahtar numarasını da dosyanın özel bir yerine yazar.

Şifrelenmiş bir bilgi çözülürken dosyadaki seçilen anahtar numarası öğrenilerek tekrar karta aynı anahtar oluşturularak çözme işlemi gerçekleştirilir.

Görüldüğü üzere açıklanan sistemde şifrelenen dosyayı yalnızca şifreleyen kullanıcı açabilmektedir. Ancak bir kullanıcı şifrelediği dosyanın belirli başka kullanıcılarca da görülmesini isteyebilir. Bu durumda bu uygulama için, seri numarası yerine kartta yaratılacak kart numarası dosyası kullanılabilir. Bu durumda TSM, erişimine izin verilen tüm kullanıcılara ilişkin verileri (kullanıcıların kart numaraları da alınmak zorundadır) aldıktan sonra dosyayı şifrelemek için yarı erişim anahtarından dinamik bir bilgi ile üretilmiş anahtarını kullanır ve dinamik bilgileri kaydedeceği dosyaya dağınık bir şekilde yerleştirir. Anahtar oluşturulurken girilen erişim bilgileri de hesaba alınarak bir anahtar oluşturulur. Buna göre, anahtarın girilen kart numarası ile bir işleme tabi tutulması kullanıcını yetkisini verecektir. Örneğin bu işlem bir modül alma işlemi olabilir. TSM

erişimine izin verilmeyen tüm kullanıcılara ‘0’, okuma izni olanlara ‘1’, değiştirme izni olanlara ‘2’ kalanını verecek şekilde bir anahtar oluşturabilir[13].

5.5 Ek Özellikler

Kullanıcı kartlarında “kişisel bilgiler” ve “E-Cüzdan” başlıkları altında ek uygulamalar da yaratılmış bulunmaktadır, ayrıca bir “Geçiş Güvenlik Uygulaması” da bu uygulamalara eklenebilir. Bu uygulamalara kısaca değinilecek olunursa :

5.5.1 Kişisel bilgiler uygulaması

Bu uygulamanın dosyaları ile kullanılan kartı bir nüfus cüzdanı gibi kullanmak, acil durumlarda ulaşılabilecek, ya da kullanıcının kendi rehberini oluşturabilecek aynı zamanda sağlık sisteminde kolaylık sağlayabilecek bir araç olarak da kullanmak mümkün olacaktır. Bu uygulamaların da erişim hakları isteğe bağlı olarak değiştirilebilir ancak, nüfus bilgilerinin değiştirilemez ancak okunabilir, kişisel bilgilerin ancak PIN doğrulaması ile okunup yazılabilmesi, sağlıkla ilgili bilgilerin de ancak ilgili personelce değiştirilebileceği bir yapı mantıklı görünmektedir.

5.5.2 E-Cüzdan uygulaması

Çoğu iş yerinde çalışanların çeşitli günlük ihtiyaçlarını karşılayabilmeleri için, kontür yüklenebilen cihazlar ya da eski moda yöntemler ile fiş ya da markalar verilir. Böylece çalışanlar yemek, içecek gibi alışverişlerini bu araçlar aracılığı ile yapar. Kart üzerindeki bir E-cüzdan uygulaması ile bu işlemi de basitleştirilebilir, aynı zamanda güvenli bir alışveriş ortamı da sunulmuş olur. Bunun için bir POS cihazı ya da kart okuyuculu otomatlar yeterli olacaktır. Harcama yapılırken de ağ yapısında kurulan karşılıklı doğrulama mantığı kullanılabilir. Yani POS ya da otomatlarda da bir SAM kart aracılığı ile karşılıklı doğrulama sağlanır ise sahte kartların önüne geçilmiş olur. Aynı zamanda yapılan işlemler gerek kullanıcı gerekse de cihazlardaki kartlarda raporlanarak kaçakların önüne geçilebilir. POS yada otomatlara periyodik olarak bir liste yüklenmesi ile kartlar otomatik olarak doldurulabilir yada hatalı bakiyesi olduğu anlaşılan kartların bakiyesi düzeltilebilir, hatta sistemde yasaklı ilan edilen bir kart bir daha kullanılamayacak şekilde

bloke edilebilir. Böylelikle işletme maliyetleri azaltılacağı gibi, yüklenen ve harcanan miktarların kontrolü de sağlanmış olur.

5.5.3 Geçiş güvenlik uygulaması

Bu uygulama özellikle işyerlerinde çalışanın sadece yetkili olduğu yerlere girebilmesinin sağlanması açısından büyük kolaylıklar sağlayacaktır. Gerekli olan donanım artık modern işyerlerinde sıklıkla görülen kart okuyuculu kapılardır. Uygulamanın dosyalarının sadece yetkili kişilerce değiştirilebilmesi gerekir. Geçiş uygulaması da özellikle çalışanların giriş çıkış saatlerinin raporlanmasında büyük kolaylıklar sağlayacaktır. İstenildiği zaman kart sahibinin çalışma saatleri ile ilgili bilgi kolaylıkla karttan okunabilecektir. Bunun için gerekli olan donanım kart okuyuculu bir turnikedir. Bu uygulamanın da okuma izni sürekli açık, yazma izni ise belli koşullara bağlı olmalıdır. Ayrıca bilgiyi tutan dosyanın da döngüsel yağıda tanımlanması gerekliliği açıktır. Akıllı kartların, bu projede kullanılan kart da dahil olmak üzere, pek çoğu bu tür dosyaları desteklemektedir.

6. GERÇEKLEME

Bu bölümde tasarlanan sistemin çalışma prensiplerini, anahtarların kullanım şekillerini ayrıntıları ile ele alacağız. Anlaşılırlığı arttırmak için tasarım bölümündeki parçaların ayrı ayrı ele alınması daha doğru olacaktır.

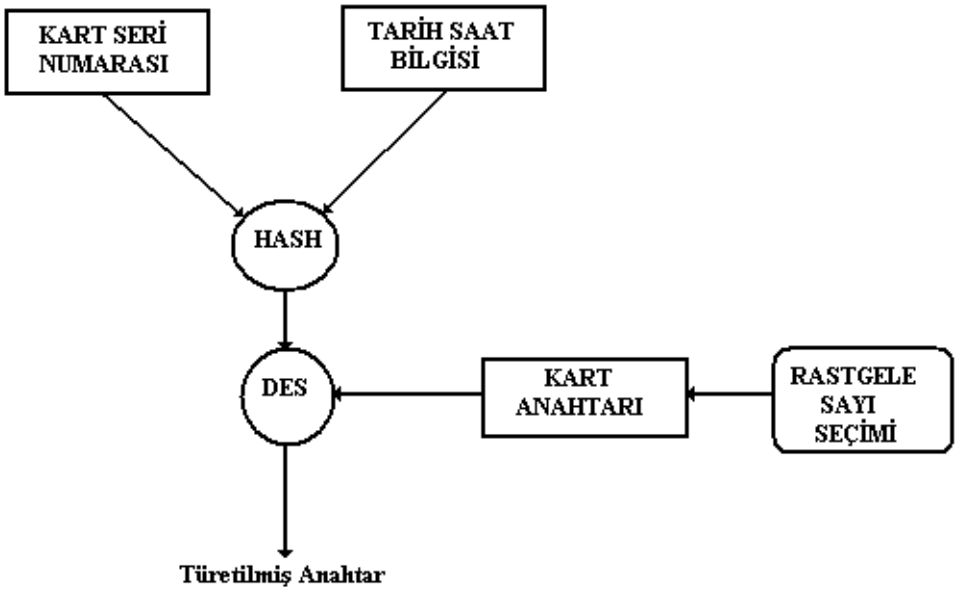
6.1 Türetilmiş Anahtarların Oluşturulması

Önceki bölümlerde, gerek bağlanma gerekse de veri şifreleme esnasında türetilmiş anahtarların kullanılacağı belirtilmişti, bu bölümde türetilmiş anahtar oluşturma mantığı belirtilecektir.

6.1.1 Akıllı kart yardımı ile türetilmiş anahtar oluşturma

Her akıllı kart, üretim esnasında oluşturulan her karta ait özel ve yinelenemez bir seri numarasına sahiptir. Bu bilgiyi tutan dosya da üretim esnasında MF altında oluşturulur. Bu bilgiyi anahtar türetmek için kullanmak türetilen her anahtarın karta özel olmasını sağlayacaktır. Güvenlik önlemini arttırabilmek için kartın üreteceği anahtar her defasında farklı olması tercih edilecektir. Bu nedenle dinamik bir değişkene ihtiyaç bulunmaktadır. Bunu sağlayabilecek en basit örnek sistem tarihi olabilir. Bu iki bilgi bir karıştırma (hash) fonksiyonuna tabi tutularak sürekli değişen bir bilgi elde edilebilir. Kullanılacak hash fonksiyonu tüm uç birimlerce bilinecek şekilde istenildiği şekilde belirlenebilir. Örneğin her iki bilginin son sekiz sekizlik bilgisi dar veya işlemine sokulabilir. Kartın ilgili anahtar kümesinden rasgele bir anahtar seçilir ve oluşturulan veri ve seçilen anahtar numarası karta gönderilerek yeni bir anahtar oluşturması talep edilecektir. Yeni oluşturulan anahtarın kopyasının karşı uç birimde de hesaplanabilmesi için dinamik veriler diğer uç birime gönderilecektir. Uç birimler arasında bağlantı sağlandıktan sonra, aralarında dinamik veri transferi olmayacak, bağlantı esnasında kullanılan dinamik veriler geçerli olacaktır.Bağlantı sonrasındaki haberleşme paketlerinde şifrelenmemiş olarak

sadece seçilen anahtar numarası gönderilecektir. Seçilen anahtar numarasını da iki yönlü bir karıştırma fonksiyonu ile gizlemek mümkündür. Şekil 6.1'de türetilmiş anahtar oluşturma mantığı gösterilmiştir.



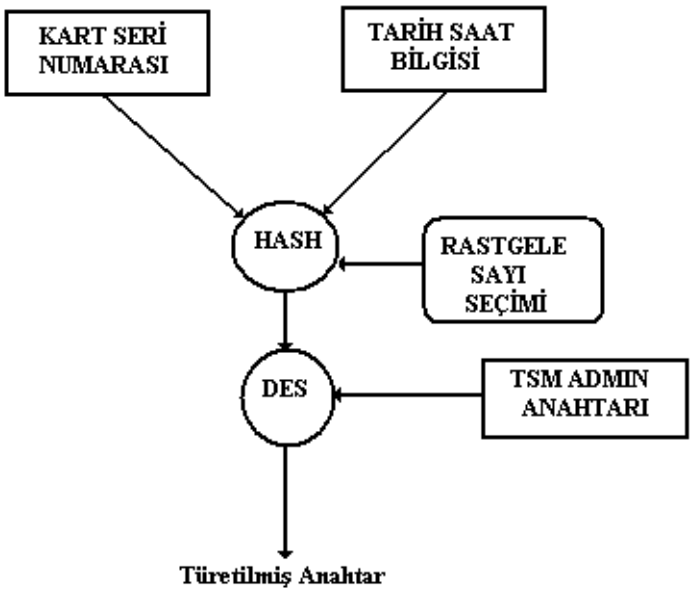
Şekil 6.1 Her iki uç birimin açık olması halinde anahtar oluşturma mantığı.

Uygulamaların, özelleştirilmelerine bağlı olarak her uygulamanın altında bir kart numarası dosyası oluşturulabilir. Böylece her uygulamaya ait kart numarası farklı olabileceği gibi, şifreleme işlemlerinde de bu numara dinamik veri olarak kullanılarak her uygulamanın oturum anahtarlarının daha da farklılaştırılması sağlanabilir. Ancak SERİ NO dosyası kartın üreticisinin yarattığı bir dosyadır ve içindeki bilginin her katta farklı olacağı garanti edilmiştir. Bahsedilen türde bir kullanım için ise kart numaralarının dağıtımı dikkatlice yapılmalıdır.

6.1.2 TSM'in geçici oturum anahtarı üretmesi

Kilitli bir uç birim üzerinde çalışan TSM bir kart aracılığı ile türetilmiş anahtar oluşturamaz. Bunun yerine HSM'den aldığı bilgileri ve barındırdığı yarı erişim anahtarını ve HSM'den gelen dinamik verileri kullanarak bir türetilmiş anahtar oluşturur ve bundan sonraki her iletişimde bu anahtarı kullanır. İstemde bulunan uç birim de HSM'in gönderdiği bilgilere dayanarak aynı anahtarı türetir. Farklı olan durum HSM'in istemde

bulunan uç birime anahtarı kartını kullandırarak ürettirmesidir. İstemci uç birime gidecek dinamik veri buna göre hesaplanır. Her iki uç birim de türetilmiş anahtarı üretmek için bir kendi seçeceği anahtar numarası da kullanır ve bunu karşı terminale göndereceği her paketin başında bildirir. Bu bilgi türetilen anahtarın oluşturulmasında kullanılır böylelikle oturum anahtarının her defasında farklı olması sağlanır. İşleyişe ilişkin şema Şekil 6.2'de gösterilmiştir.



Şekil 6.2 Uç birimlerden birinin kilitli olması haline anahtar oluşturma mantığı.

6.2 Paket Yapısı

Ağ üzerinde dolaşan tüm paketler, bağlantı istekleri de dahil olmak üzere, şifrelenecektir. Şifrelenmiş bir paketin başında, şifreleme için kullanılan anahtarın numarası açık ya da bir karıştırma fonksiyonuna tabi tutulmuş şekilde gönderilir. Şekil 6.3'de bir paketin genel yapısı görülmektedir.Paketin şifrelenmemiş kısmının son verisi paket tipidir. Bunlar, ağ bağlantı isteği, ağ bağlantı cevabı, uç birimler arası bağlantı isteği, cevabı, şifrelenmiş bilgi paketi ve kontrol paketi şeklindedir.



Şekil 6.3 Genel paket yapısı.

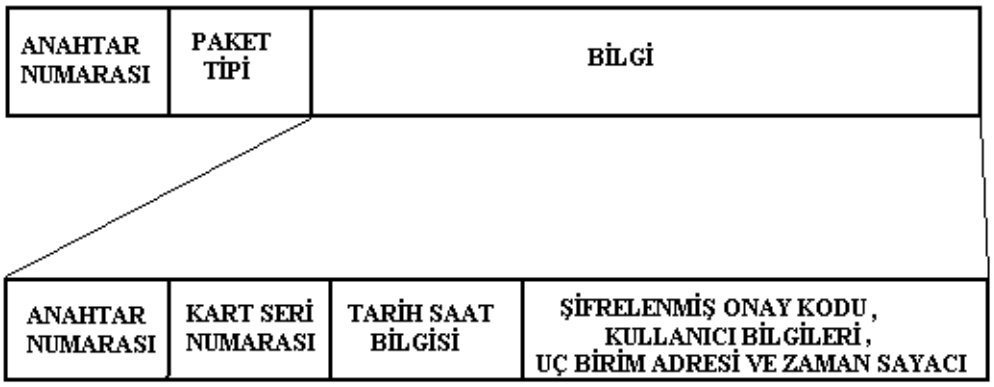
6.2.1 Bağlantı isteği ve cevabı

Bağlantı isteği ve cevabında, kartın seri numarası, tuttuğu zaman bilgisi ve seçilen anahtar numarası açık olarak gönderilir. Takip eden, kullanıcı bilgileri ve üretilen onay kodu şifrelenmiş olarak gönderilir. Ancak bağlantı istek ya da cevap kısmı da bir paket olduğundan paket için uygulanan şifreleme modeli tüm pakete uygulanır. Yani aslında açık olarak belirttiğimiz alanlar, iletişim anahtarları ile, diğer alanlar ise önce bağlanma anahtarları ardından da iletişim anahtarları ile şifrelenmiş olarak gönderilir. Böylelikle çok katmanlı bir güvenlik yapısı kurulmuş bulunur[24]. Şekil 6.4'te bağlantı paketlerinin genel yapısı gösterilmiştir.

Duruma tek istisna, kilitli bir terminale HSM tarafından gönderilen uç birim bağlantı isteğidir. Bu durumda bağlantı isteği kısmında ek olarak istekte bulunan uç birim bilgileri ve oturum anahtarı zaman sayacı da gelmektedir ve şifreleme için TSM yarı erişim anahtarı kullanılır.Benzer şekilde, istemcinin kilitli bir uç birime bağlanma isteğini HSM'e bildirmesi durumunda bağlanılmak istenen terminal bilgisi de HSM'e gönderilir. Duruma ilişkin paket yapısı Şekil 6.5'te gösterilmiştir.



Şekil 6.4 Bağlanma isteği paketinin genel yapısı.



Şekil 6.5 Uç birimlerden biri kilitli olduğunda bağlantı paketi.

6.2.2 Kontrol paketleri

Kontrol paketleri ağdaki durum bilgilerinin bildirilmesi için kullanılır. Bu paketler kara liste yayını, ağ bloke ve açma yayını, her türlü ret cevabı olacaktır.Bu paketlerin yapısı Şekil 6.6'da gösterilmiştir.



Şekil 6.6 Kontrol paketi genel yapısı

6.3 Uç Birimin Açılması, Ağa Bağlanma ve Haberleşme

Bir uç birimin açılması, ağa bağlanması ve haberleşme esnasında TSM ve HSM modülünde koşan uygulamaların temel mantığı basitçe verilebilir.

6.3.1 TSM modülü

TSM modülünün yapması gereken kartı kontrol etmek ve HSM'e bağlantı isteğini iletmektir.Bu bölüme ilişkin basitleştirilmiş kod aşağıdaki gibi düşünülebilir.

```
void TSM_Main(void)
{
    status = GetStatus ();
    if (status == LOGGED_OFF)
   {
          if (PromptUserCardInsertion()) TSM_RegisterUser();
   }

   switch (CheckIncomingRequest ( packet ))
  {

     case CONNECT_REQ : SendConnectResponse(status);break;
     case DATA_PACK      : if (status == LOGGED_ON)TSM_Decrypt(packet);break;
     case CNTRL_PACK    : HandleControlPacket(packet);break;
     case HSM_CON_REQ : HandleOfflineConnection(packet, status);break;
     default                      : break;

   }

}
```

```
int TSM_RegisterUser(void)
{

    if (!SELECT (SERIAL_NO_FILE)) return (-1);
    if (!READ (cardSerialNo)) return (-1);
    if (!ReadAndParseAID (NET_SECURE_LOGON_APP, &aidData)) return (-1);
    if (!SELECT(aidData.pinFile)) return (-1);
    if (!AskUserVerifyPIN ()) return ( -1 ) ;
    if (!SELECT(aidData.parameterFile)) return (-1);
    if (!READ(userData)) return (-1);
    status = TSM_NetLogonUser(aidData, userData, cardSerialNo);
    if (status < 0) return ( -1);
    if (!TSM_KernelUserLogon(userData)) return (0);
    RegisterLogonStatus ( status );
    return (0);
}
```

Kart seri numarası okunduktan sonra, ağ bağlanma uygulaması seçilecek şekilde AID dosyaları taranır ve yaprak AID'nin bilgileri bir yapıya aktarılır. Uygulamaya ait PIN doğrulandıktan sonra uygulama parametre dosyasından kullanıcı bilgileri alınır ve güvenli ağ bağlanma işlemi başlatılır. İşlemin başarılı sonuçlanması durumunda işletim sistemi bağlanması gerçekleştirilir ve güvenli ağ bağlanmasının tipi durum bilgisi olarak kaydedilir (sunucuya bağlantı olup,olmadığı şeklinde). Ağ bağlantısının yapılamaması durumunda kullanıcı sisteme kabul edilmez.

TSM yazılımının ana döngüsü sürekli akıllı kartın takılı olup olmadığını kontrol edecek, çıkarıldığı anda makul bir süre içerisinde aynı kart takılmaz ise sistemi kilitleyecek ve bağlantıları koparacaktır.Örnek kod şu şekildedir:

```
void TSM_CheckUser(void)
{
     if (!SmartCard_IsExist())
    {
         PromtCardRemovalWarning();
          if (WaitForCardReconnect())
               if (CheckCardInserted()) return;

          RegisterLogonStatus (LOGGED_OFF);
          ClearConnecetionTable(ALL);
          ClearTimers (ALL);
    }
}
```

Kullanıcıyı ağa bağlamak için gerekli rutinler aşağıdaki basitleştirilmiş fonksiyonlar ile gerçekleştirilebilir.

```
int TSM_OfflineLogon( AIDStruct aid, char *cardSerialNo )
{

        TSM_DESEncrypt ( cardSerialNo, ADMIN_KEY,  cardKey);

        LockCardUsage();
        if (!SmartCard_VerifyKey(ADMIN_KEY,cardKey)) return (-1);
        ReleaseCardUsage();

        return (0);
}



int TSM_NetLogonUser( AIDStruct aid, char *userData, char *cardSerialNo )
{
      LockCardUsage();
      GetSystemDate(dateTime);
      randomString  = Hash ( cardSerialNo, datetime);
      randomKeyNo = GetRandomKeyNumber();
      if (!SELECT(aid.keySetFile)) err = TRUE;
      if (!err) if (!SmartCard_GenerateDiversifiedKey(randomKeyNo, randomString))
         err = TRUE;
      if (!err) if (!SmartCard_GetResponse(authenticationChallenge))
         err = TRUE;
      ReleaseCardUsage();
       if (err) return (-1);
      packet = PrepareLogonPacket( randomKeyNo, dateTime, cardSerialNo,
                                          authenticationChallenge);
      if (!TSM_SecureSendReceive ( HSMAddress,  packet))
          return(TSM_OfflineLogon(aid, cardSerialNo));

      if(ResolveLogonPacket(packet,randomKeyNo, authenticationChallengeHSM))
     {
          LockCardUsage();
          if (SmartCard_GenerateDiversifiedKey(randomKeyNo, randomString))
               if (SmartCard_GetResponse(authenticationChallenge))
                   if ( authenticationChallenge == authenticationChallengeHSM)
                   {
                        ReleaseCardUsage();
                        return (1);
                    }
     }
     ReleaseCardUsage();
     return (-1);
}
```

Uç birimi diğer bir uç birime bağlayan fonksiyon, karşılıklı doğrulamayı gerçekleştirmeli, doğrulama yapıldığı taktirde iletişimin devamı için kullanılması gerekli bilgileri o proses için saklamalıdır.

```
int TSM_ConnectionRequest ( char *IPaddress )
{
    LockCardUsage();
    if (!SELECT (SERIAL_NO_FILE)) err = TRUE;
    if (!err) if (!READ (cardSerialNo)) err = TRUE;
    if (!err) if (!ReadAndParseAID (NET_SECURE_LOGON_APP, &aidData))
                 err = TRUE;
    if (err)
    {
       ReleaseCardUsage();
       return (-1);
    }
    GetSystemDate(dateTime);
    randomString  = Hash ( cardSerialNo, dateTime);
    randomKeyNo = GetRandomKeyNumber();
    if (!SELECT(aid.keySetFile)) err = TRUE;
    if (!err) if (!SmartCard_GenerateDiversifiedKey(randomKeyNo, randomString))
                err = TRUE;
    if (!err)  if (!SmartCard_GetResponse(temporaryKey))    err = TRUE;
    ReleaseCardUsage();
    if (err ) return (-1);

    packet = PrepareLogonPacket( randomKeyNo, dateTime, cardSerialNo );
    if (!TSM_SecureSendReceive ( termialAddress,  packet))   return(-1);
    ret=ResolveConnectionResponse(packet,randomKeyNo,serialNoOtherTerm,
                                      dateTimeOtherTerm, auhenticationChallengeOtherTerm))
    if (ret == CONNECTION_ACCEPTED)
  {
    LockCardUsage();
    randomString  = Hash (serialNoOtherTerm, dateTimeOtherTerm);
    if (!SmartCard_GenerateDiversifiedKey(randomKeyNo, randomString))
         err = TRUE;
    if (!err)  if (!SmartCard_GetResponse(authenticationChallenge))
                 err = TRUE;
    if (!err)  if ( authenticationChallenge != authenticationChallengeOtherTerm)
                 err = TRUE;
    if(!err) RegisterConn(terminalAddress,  dateTimeOtherTerm, serialNoOtherTerm,  dateTime,
                                serialNo);
    ReleaseCardUsage();
    if (err)  return (-1);
    return (1);
 }
 else if (ret ==TERMINAL_LOCKED)
        return ( TSM_OfflineConnectionRequest (terminalAddress));

  return (-1);
}
```

```
int TSM_SecureSendReceive (char *IPAddress, char *data)
{
   GetConnProperties(terminalAddress, dateTimeOtherTerm, serialOtherTerm,  dateTime,
                              serialNo);

    if (!ReadAndParseAID (NET_SECURE_TRANS_APP, &aidData))
          err = TRUE;
    if (err)
    {
       ReleaseCardUsage();
       return (-1);
    }
    randomString  = Hash (serialNo, dateTime);
    randomKeyNo = GetRandomKeyNumber();
    if (!SELECT(aid.keySetFile)) err = TRUE;
    if (!err) if (!SmartCard_GenerateDiversifiedKey(randomKeyNo, randomString))
          err = TRUE;

    if (!err)  if (!SmartCard_GetResponse(temporaryKey))    err = TRUE;

    ReleaseCardUsage();
    if (err ) return (-1);

    TSM_Encrypt(data, temporaryKey, encryptedPacket);
     packet = PrepareTransPacket (randomKeyNo, serialNo, encryptedPacket);
    TSM_IPSend(IPAddress, packet);

     if (!TSM_IPReceive(IPAddress, packet)) return (-1);
     ResolveTransPacket(packet, keyNo, encryptedPacket);

     LockCardUsage();
     randomString  = Hash (serialOtherTerm, dateTimeOtherTerm);
     if (!SELECT(aid.keySetFile)) err = TRUE;
     if (!err) if (!SmartCard_GenerateDiversifiedKey(keyNo, randomString))
          err = TRUE;

     if (!err)  if (!SmartCard_GetResponse(temporaryKey))    err = TRUE;

      ReleaseCardUsage();
      if (err ) return (-1);

      TSM_Decrypt(encryptedPacket, temporaryKey, data);
      if (ControlPacket(data, serialNoOtherTerm)   return(1);
      return(-1);
  }
```

6.3.2 HSM modülü

HSM modülü temel olarak tüm TSM fonksiyonlarına sahip olmalıdır. TSM'den farklı olarak yapabildiği işlemler kara liste ya da ağ bloke yayınlarını yapmak ve uç birimlerden birinin kilit olması durumunda iki uç birim arasındaki bağlantıyı sağlamaktır. Önemli olan bu son fonksiyonun basitleştirilmiş kodu :

```
void HSM_InitiateOfflineConnection( char *requestIP, char *targetIP, char *requestSerialNo, char
                                        *requestDateTime)
{

        keyNo = GetRandomKeyNumber();
        packet=PrepareOfflinePacket ( requestIP, requestSerialNo,requestDateTime,keyNo,
                                        KEY_LIFE_TIME);

        if (!HSM_OfflineRequest (targetIP, packet))  return;

        ResolveOfflineResponse(packet, targetSerialNo, targetDateTime);

        LockCardUsage();
        Generate_KeyChallenge( requestSerialNo, requestDateTime, keyNo, keyChallenge);
        ReleaseCardUsage();

         packet  = PrepareOfflineResponse (  keyNo, keyChallenge);
         HSM_SecureSend (  requestIP, packet);
}
```

7. SONUÇLAR

Ağ ortamlarında güvenlik genellikle yazılım tabanlı çözümler ile çeşitli seviyelerde sağlanabilir. Konu üzerinde gerek teorik gerekse de gerçek hayata geçirilmiş pek çok tasarım bulunmaktadır ve bunların bazılarına bu tez içerisinde değinilmiş, fayda görülen özellikleri kullanılmıştır. Yine de konu üzerinde tercih edilen yöntemler, ağa bağlanmayı zorlaştırmaya yönelik olmaktan çok, ağdaki verileri gizlemeye yöneliktir.

Tezde başlangıç noktası olarak, bir ağı korumanın en güvenilir yolunun yetkisiz kişilerin uç birimlere ve dolayısı ile ağa bağlanmalarını engellemek ile başlayacağı savı seçilmiş, incelenen sistemlerin güçlü yanları mümkün olduğunca birleştirilmeye çalışılarak sonuç sistem elde edilmeye çalışılmıştır.Aynı zamanda sıkı bir kullanıcı denetim mantığı ile kullanıcıların uç birimlerden uzaklaşmaları ile bağlantının koparılması sağlanmıştır. RSA yönteminin güçlü tarafı olan anahtarların gizli kalabilmesi, akıllı kartlar kullanılarak nispeten sağlanmış, aynı zamanda karşılıklı doğrulama mantığı ile sağlanan bağlantının güvenilirliği pekiştirilmiştir. Akıllı kartların güvenilirliği ispatlanmış cihazlar olması da sistem için bir artı hanedir.

Ağ güvenliğinin sağlanması yanında, kişisel bilgisayarlar ve verileri de korunmaya çalışılmış, her türlü olasılığa karşı bilgisayar üzerindeki verilerin, kullanıcısının dahi bilemediği anahtarlar ile şifrelenmesi sağlanmıştır, bunun için yine akıllı kartların özelliklerinden faydalanılmıştır. Kullanılan yöntem şifrelenmiş dosya paylaşımını da destekleyecek şekilde seçilmiştir.

Sistem tasarlanırken düşük maliyetli ve gerçekleştirilebilir kriterlerine uygun davranılmasında azami gayret gösterilmiş, mühendisçe bir yaklaşım ile en düşük maliyetli ancak zayıf ya da en yüksek maliyetli ancak gereğinden fazla güçlü bir sistem yerine azami maliyet/performans oranını sağlayan bir sistem ortaya çıkarılmıştır. Kerberos türevi sistemlerde kullanılan anahtar dağıtım sunucusu, asıl sunucu üzerinde

koşan bir yazılım, güvenli cihaz ve akıllı kart gibi nispeten düşük maliyetli ancak yeterince güvenilir bir sistem ile değiştirilmiş böylece performans artışı da sağlanmıştır.

Tezin hedef aldığı insan hata ve ihmallerinden kaynaklanan açıkları asgariye indirmek amacına, kullanıcıların sürekli denetlenmesi ile yaklaşılmıştır.

8. TARTIŞMA

Bu aşamaya kadar tezde bahsi geçen sistem ile ilgili tüm açıklamalar sunulmuştur.Güvenlik için kurulan sistem, ek fonksiyonları ile bir çeşit ofis otomasyon sistemi haline getirilmiştir. Geliştirilen sistem gerek ek özellikleri bakımından gerekse de güvenlik özellikleri bakımından daha ileri noktalara götürülebilir.

Yapılan araştırmayı bir aşama ileriye götürüp, büyük ölçekteki ağlara, özellikle internet ortamına uygun hale getirmek olasıdır. Ancak bu tür ortamlarda farklı ağ topolojileri ve protokolleri de kullanılabileceği, yoğun ağ trafiğinin büyük bir kısmının veri gizlemeye ihtiyaç duymaması oluşturulması gereken programları çok karmaşık hale getirebilecektir. Oluşturulan akıllı kart yapısı, büyük ölçekteki ağlara da kolaylıkla uyarlanacak şekilde, uygulama eklenebilecek şekilde tasarlandığından dolayı akıllı kart tarafında büyük bir değişikliğe gerek kalmayacaktır. İç içe girmiş şekildeki ağ ortamlarında ise uç birimlerde koşan yazılımların kullanıcı kontrolü yanısıra ek olarak ağ seviye kontrolü de yapması gerekliliği açıktır. Zira bir uç terminalin hangi ağda bulunduğu ve hangi ağlara erişim yetkisinin bulunduğu ayırt edilebilmelidir.

Sistemde geliştirilebilecek noktalardan biri yazılımların işletim sistemi çekirdeğine entegre edilmesidir. Bunu yapmak için işletim sisteminin kaynak kodlarına gerek olmayabilir. Modern çoğu işletim sistemi, geliştiricilere işletim sistemine ek özellikler kazandırabilmesi için bir çok API sunar. Uç birimlerde koşan yazılımlar bu API'ler kullanılarak işletim sistemine entegre edildiğinde, dosya paylaşımları ile ilgili özellikler, uç birim uygulaması yerine işletim sistemi özellikleri kullanılarak değiştirilebilir. Bu sayede uç birim yazılımlarının yükü daha hafifleyecektir.

Sistemde paket şifrelemesi ve bağlantı yordamları uygulama düzeyinde yapılmaktadır. Temel görev olarak bu bir ağ protokolünün bir parçası olarak görülürse, bu tür bir işleyişin ağ donanımı ile beraber çalışması, hem güvenilirlik hem de performansı arttırıcı

bir özellik olacaktır. Sistemin ideal halinde, paylaşım hakları işletim sistemi özellikleri ile kontrol edilebilmeli, iletişim ise ağ donanımını kontrol eden yazılım ya da sürücü tarafından yapılmalıdır.

Tasarlanan sistemde tüm anahtarlar gizlenirken, bu anahtarlarların bir şekilde elde edilmesi de mümkün olduğunca zorlaştırılmıştır. Ancak kilitli uç birimlere erişimde ve sunucunun olmaması halinde kullanılan "yarı erişim" anahtarı, uç birim yazılımının içine gömülü halde bulunmaktadır. Bu anahtar,işinde ustalaşmış bir kişi tarafından çalıştırılabilir kod makine dilinde incelenerek elde edilebilir. Her ne kadar çalıştırılabilir dosyayı kopyalamak için bile uç birimi açabilecek bir akıllı karta ihtiyaç olsa da, bir şekilde sistem içindeki biri çalıştırılabilir kodu dışarıya transfer edebilir. Bu nedenle bu anahtarlar ile yapılan işlemlerde kullanıcılara sınırlamalar getirilmektedir. Bu noktada, kilitli terminaller ile çalışılmasında daha güvenilir bir yapı kurulabilir.

Sistemde gelişmeye açık diğer bir nokta anahtarların güncellenmesidir. Mevcut sistemde seçilen akıllı kartlar sadece anahtar yüklemeye izin vermektedir. Bu nedenle anahtar küme dosyası akıllı kartın izin verdiği ölçütlerde ne kadar büyük seçilirse seçilsin kullanılabilecek anahtarların bir sınırı bulunmaktadır. Ayrıca yeni anahtar yükleme eski anahtarları silmediği için eski anahtarların kullanımı TSM ve HSM modüllerinde bir liste tutulmadığı taktirde engellenemeyecektir. Var olan anahtarlarını değiştirebilen akıllı kartların daha popüler hale gelmesi ile HSM modülü tarafından bir anahtar değiştirme paketi vasıtası ile uç birim kartlarının anahtarları değiştirilebilirse anahtarların gizliliği pekiştirilmiş olur.Anahtarlar değiştirilirken, yeni anahtar bilgisi HSM modülü tarafından üretilebilir bu sayede anahtarların hiç kimse tarafından bilinmezliği sağlanmış olur ancak acil durumlarda anahtar bilgilerine ihtiyaç duyulduğu taktirde anahtarların bilinememesi problem doğurabileceğinden HSM kartını koruyan ve çalıştıran donanım üzerindeki yazılama yeni anahtarların sadece yetkili bir kişinin, güvenli bir ortamda girilmesi sağlanabilir.

Sistemin şu an performansı önemli ölçüde düşürebilecek en kritik noktası her uç birimde sadece bir kart kullanılmasıdır, sonuçta akıllı kart paylaşılan bir kaynaktır ve tektir. Karta bir anda TSM'in bir prosesi ulaşmalıdır. Bu nedenle de akıllı kartı kullanan tüm prosesler semafor mantığıyla çalışmaktadır.Ancak bir prosesin kart ile işi bittikten sonra diğer bir

proses karta erişebilir. Doğrulama ve anahtar türetme komutları arasında karta erişilmesi, oluşturulacak doğrulama verilerinin geçersiz olmasına neden olacaktır.

Sonuç olarak, tezin kaleme alındığı tarihte piyasaya çıkan bir işletim sistemi için 200 milyon USD gibi bir harcamanın sadece güvenliğe yönelik olarak yapılması konunun ciddiyetinin bir göstergesidir.

KAYNAKLAR

[1] Ward, R.; Skeffington,P. ;1990. Networn Management Security, IEEE (web)

[2] Schier, Katrin; 2000. Multifunctonal Smartcards for Electronic Commerce- Application of the Role And Task Based Security Model, IEEE Spectrum

[3] Stockel, A.; 1995. Securing Data And Financial Transactions Institute of Electrical and Electronics Engineers 29th Annual 1995 International Carnahan Conference on, 18-20 Oct

[4] Ploog, Hagen; Timmermann, Dirk; 1996. FPGA Based Architecture Evaluation of Cryptographic Coprocessors For Smartcards, IEEE (web)

[5] Scheuermann, D.; 2002. The Smartcard As a Mobile Security Device Electronics & Communication Engineering Journal, Volume: 14 Issue: 5, Oct 2002

[6] Thomasson, J.-P.; Baldi, L.; 1997. Smartcards: Portable Security Innovative Systems in Silicon, 1997. Proceedings., Second Annual IEEE International Conference on, 8-10 Oct 1997

[7] Vaidya, A.W.; 1995, Keeping Card Data Secure At Low Cost Security and Detection, 1995., European Convention, 16-18 May 1995

[8] ISO, 7816, Part I-IV

[9] Abrial, A.; Bouvier, J.; Senn, P., Renaudin; M., Vivet P.; 2001, A New Contactless SmartCard IC Using On-Chip Antenna and Asynchronous Microcontroller, IEEE (web)

[10] Hallsall, Fred; 1995. Data Communications, Computer Networks and Open Systems, Addison-Wesley, ISBN 0-201-42293-X

[11] Itoi, Naomaru; Honeyman, Peter; 1999. Practical Security Systems With SmartCards, IEEE(web)

[12] Soriano, M.; Forne, J.; Melus, J.L.; Recacha F.; 1993. Implementation of a Security System in a Local Area Network Environment, IEEE(web)

[13] Tseng, J.R.; Yang, Wei-Pang; 1990. A New Access Control Scheme With High Data Security, IEEE (web)

[14] Meadows, Catherine; 1997. Computer Security :The Good, The Bad and The Ugly, IEEE (web)

[15] O'Halloran, Collin; 1999. Trusted System Construction, IEEE (web)

[16] Lavery, Janet; BoldyReff, Cornelia; 2001. Issues In Securing Web Accesible Information Systems, IEEE (web)

[17] Schlumbereger Systems, 1998. Schlumberger PayFlex Mk5 Reference Manual. www.schlumberger.com

[18] Hadjina, N.; Thompson, P.; 2000. Data Security on Ethernet LANs Electrotechnical Conference, 2000. MELECON 2000.

[19] Chang, C.-C.; Hwang, R.-J.; Buehrer, D.; 1993. Using Smart cards To Authenticate Passwords, Security Technology, 1993. Security Technology, Proceedings. Institute of Electrical and Electronics Engineers 1993 International Carnahan Conference, 13-15 Oct 1993

[20] Microsoft Developers Network, 2002. Microsoft Corporation

[21] Wong, Ping Wah; Tretter, Daniel; Kitte, Thomas; Lin, Qian; Nguyen, Hugh; 1998.A Web Based Secure System For The Distributed Printing Of Documents And Images, IEEE (web)

[22] Chung-Huang Yang; Shy-Ming Ju; Rao, T.R.N.; 1998. A Smartcard-Based Framework for Secure Document Exchange Security Technology, 32nd Annual 1998 International Carnahan Conference , 12-14 Oct 1998

[23] Ham, Lein; Chien, Yaw-Ruey; Kiesler, Thomas; 1990.An Extended Cryptographic Key Genereation Scheme For Multilevel Security, IEEE (web)

[24] DOS Bilgisayar A.Ş. www.dos.com.tr

ÖZGEÇMİŞ

Hakan Uluöz, 16.09.1977 tarihinde Ankara'da doğmuştur. Temel eğitimini Diyarbakır ve Trabzon illerinde tamamlamıştır. 1995 yılında Trabzon Anadolu Lisesi'nden mezun olmasına müteakip aynı yıl İstanbul Teknik Üniversitesi Kontrol-Bilgisayar Mühendisliği bölümünde yüksek öğrenim hayatına başlamıştır. 1999 yılında bölümden mezun olup aynı yıl yine İstanbul Teknik Üniversitesi'nde, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği dalında yüksek lisans öğrenimine başlamıştır.

1999 yılında atıldığı iş hayatında, kredi kartı ve akıllı kartlı ödeme sistemleri, kablosuz finansal işlem sistemleri üzerine yurt içi ve yurt dışında çalışmalar yapmış ve konular üzerinde uzmanlaşmıştır. Halen DOS Bilgisayar A.Ş.'de proje yöneticisi olarak çalışmaktadır.

İyi seviyede İngilizce, başlangıç seviyesinde Almanca bilmektedir. İlgi alanları, elektronik sistemler, robotlar, yapay zeka ve hayvanlardır.

İletişim bilgileri:

E-mail h_uluoz@yahoo.com

Tel +90 532 658 32 33