İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ŞEBEKE GÜVENLİĞİ ve AKILLI KARTLAR

YÜKSEK LİSANS TEZİ
Müh. Tarık TAKTAKÇI
504981068

Tezin Enstitüye Verildiği Tarih : 7 Kasım 2003
Tezin Savunulduğu Tarih : 14 Ocak 2004

Tez Danışmanı :	**Prof.Dr. Günsel DURUSOY (İ.T.Ü.)**
Diğer Jüri Üyeleri	**Prof.Dr. Bilge GÜNSEL (İ.T.Ü.)**
	Yrd.Doç.Dr. Demir ÖNER (İ.Ü.)

OCAK 2004

ÖNSÖZ

Bana kendisi ile çalışma imkanı veren, araştırmalarımda yol göstererek değerli katkılarını esirgemeyen sayın hocam Prof. Dr. Günsel DURUSOY'a, akıllı kart okuyucuları ve örnek akıllı kartlar vererek uygulama geliştirme ortamı sağlayan OYTEK Firması Kart Teknolojileri Grup Müdürü sayın Tayfun ÖZÇAY'a ve Kart Teknolojileri Yönetmeni sayın Onur Ayata'ya, çalışmalarım boyunca büyük destek, moral veren aileme teşekkür etmeyi bir borç bilirim.

Kasım 2003 Tarık TAKTAKÇI

İÇİNDEKİLER Sayfa No

KISALTMALAR

3DES : Triple Data Encryption Standard
AAM : Application Abstract Machine
AC : Access Condition
ACK : Acknowledge
ADC : Application Delete Certificate
ALC : Application Load Certificate
APDU : Application Protocol Data Unit
API : Application Protocol Interface
ATR : Answer To Reset
BGT : Block Guard Time
bps : Bits Per Second
BWT : Block Waiting Time
CAP : Converted Applet
CERT/CC : Computer Emergency Response Teams Coordination Center
Cks : Checksum
CLA : Class
CRC : Cyclic Redundancy Check
CRYCKS : Cryptographic Checksum
CS : Checksum
CWT : Character Waiting Time
DES : Data Encryption Standard
DF : Dedicated File
EF : Elementary File
etu : Elementary Time Unit
FDB : File Description Byte
FID : File Identifier
FN : File Number
FP : File Pedigree
G/Ç : Giriş / Çıkış
ICMP : Internet Control Message Protocol
IDEA : International Data Encryption Algorithm
IEC : International Elecrotechnical Commission
INS : Instruction
ISO : International Standards Organization
JCRE : Java Card Runtime Environment
JCVM : Java Card Virtual Machine
KMA : Key Management Agent
Kv : Key Version
LRC : Longitudinal Redundancy Check
lsb : Least Significant Bit
MEL : MULTOS Executable Language
MF : Master File

MPCOS : Multi-application Payment Chip Operating System
MPN : Maximum Presentation Number
msb : Most Significant Bit
NIST : National Institute of Standards and Technology
PB : Procedure Byte
PGP : Pretty Good Privacy
PTS : Protocol Type Selection
RC5 : Rivest Cypher Version 5
RFU : Reserved For Future Use
RL : Record Length
RSA : Rivest Shamir Adleman
SCL : Serial Clock
SCN : Secret Code Number
SCR : Secret Code Ratification
SDA : Serial Data
SK : Session Key
SMTP : Simple Mail Transfer Protocol
SN : Serial Number
SW : Session Word
SYN : Synchronization Bit
TTL : Transistor – Transistor Logic
UCR : Unlock Code Reference
URL : Uniform Resource Locator
VPN : Virtual Private Network
XOR : Exclusive OR

ŞEKİL LİSTESİ

ŞEBEKE GÜVENLİĞİ VE AKILLI KARTLAR

ÖZET

Bu çalışmanın amacı, bilgisayar ağlarının güvenliğini arttırmada akıllı kartların katkılarını incelemektir. Bu konuda genel bir giriş birinci bölümde verilmiştir.

İkinci bölümde şebeke güvenliği konusunda temel bilgiler verilmiştir. Bu bölümün birinci kısmında, kurulum amacı internet güvenliğini arttırmaya yönelik araştırmalar yapmak olan CERT/CC (Computer Emergency Response Teams Coordination Center) organizasyonu tarafından yapılmış bir çalışma olan "Güvenlik Uygulama Döngüsü" tanıtılmış ve bu döngünün her bir adımında yapılması gereken işlemler belirtilmiştir. Bölümün ikinci kısmında ise bilgisayar sistemlerine yapılan başlıca 5 saldırı metodu incelenmiş, bunlara karşı alınabilecek önlemler üzerinde durulmuştur.

Üçüncü bölümde temel kriptoloji bilgisi verilmiştir. Bu bölümde simetrik ve asimetrik şifreleme algoritmaları karşılaştırmalı olarak anlatılmış, çırpma fonksiyonları ve sayısal imzalar hakkında bilgi verilmiştir.

Dördüncü bölümde akıllı kartlar konusunda detaylı bir inceleme yapılmıştır. Akıllı kartların sınıflandırılması, elektriksel özellikleri, veri iletişim protokolleri, işletim sistemleri ve uygulama geliştirme ortamları bu bölümde anlatılan konulardır.

Beşinci bölümde, akıllı kartlar kullanılarak yapılan bir güvenli şebeke haberleşmesi uygulaması gerçekleştirilmiştir. Öncelikle uygulamada kullanılan akıllı kartların işletim sistemi MPCOS (Multi-application Payment Chip Operating System) incelenmiştir. Daha sonra uygulamanın temel özellikleri olan kimlik doğrulama ve şifreli iletişim fonksiyonları anlatılmıştır.

Altıncı bölüm, tezin sonuç kısmıdır. Bu bölümde akıllı kart kullanımının avantajları ve dezavantajlarından bahsedilmiştir. Akıllı kartların günlük hayatımıza bu denli hızla girmesinin nedenlerinden anlatılmıştır. Akıllı kart uygulama geliştirme ortamlarının akıllı kart kullanımına sağladığı faydalardan bahsedilmiş, işletim sistemlerinin akıllı kart kavramını desteklemesinin önemi tartışılmıştır.

NETWORK SECURITY AND SMART CARDS

SUMMARY

The aim of this study is to investigate the contribution of smart cards to computer network security. Part 1 is an introduction to the study.

In Part 2, a basic explanation about network security was given. In the beginning of this section the study of "Security Practices Structure" made by CERT/CC (Computer Emergency Response Teams Coordination Center) was introduced. CERT/CC is an organization which was established to research various methods of internet attacks and develop solutions for them. All the defensive tactics that can be taken in any phase were detailed. Then the 5 most common internet attacks were examined and the precautions against these attacks were explained.

In Part 3 a general summary of cyrptography was given. Symetrical and asymetrical encryption algorithms, hash functions and digital signature concept were explained.

In Part 4 a detailed investigation about smart cards was reported. Smart card classification, electrical properties, communication protocols, operating systems and development environments were also explained.

In Part 5 an application using smart cards was accomplished. To start with, the operating system of sample cards, MPCOS (Multi-application Payment Chip Operating System), was examined. Following this, the main functions of the application, authentication and encryption, were explained.

Part 6 is the result section. In this part, the advantages and disadvantages of smart cards were discussed. The reasons of smart cards popularity were explained. The benefits of smart card application developing environments were introduced. The importance of popular operating systems, which support the smart card concept, was discussed.

1. GİRİŞ

Bilişim sistemlerine olan bağımlılığımız arttıkça sistemlerde meydana gelebilecek arıza ve saldırılara karşı duyarlılığımız da artmaktadır. Bilgisayar sistemleri ve ağlarına yönelik saldırılar önemli miktarda para, zaman, prestij ve değerli bilgi kaybına neden olabilmektedir. Sistemlere yapılan saldırıları en az zararla atlatabilmek için güvenli ve sağlam yapılar kurulmalıdır.

Güvenli şebeke haberleşmesinin temelinde kullanıcı kimlik tanımlayıcıları, parolalar ve şifreleme algoritmaları vardır. Taraflar kimlik tanımlayıcılarını kullanarak birbirlerini tanır, parolalar ile yetkileri dahilinde işlemler yapar, şifreleme yöntemleri kullanarak yalnızca haberleşen tarafların anlayacağı biçimde veri iletiminde bulunur. Sistemlerde kullanılan kimlik tanımlayıcıları, parola ve şifrelerin sağlamlığı, taraflar arasındaki değiştokuş yöntemleri, sistem güvenliğinin önemli bir kısmını oluşturmaktadır.

Yapılan çalışmada ele alınan güvenlik sistemi, bir bilgisayar ağında haberleşen tarafların birbirlerinin kimliklerinden emin olmalarını sağlamakta ve yetkisiz olarak araya giren üçüncü bir tarafın haberleşmeyi dinleyebilmesini kriptografik algoritmalar kullanarak önlemektedir.

Güvenlik sistemlerinde akıllı kartların kullanılmasının amacı, kullanıcıya ait şifre ve kimlik bilgilerinin daha güvenli bir ortamda saklanması ve gerektiğinde bu bilgilerin hızlı ve hatasız olarak sisteme girilebilmesidir. Akıllı kartı kullanmadan önce kullanıcının kart şifresini girmesini zorunlu kılmak, kartın yetkisiz kişilerin eline geçmesi ihtimaline karşı alınabilecek güçlü bir önlemdir.

Ne tür ataklarla karşı karşıya olunduğu bilinmeden, bir bilgisayar şebekesini korumak mümkün değildir. Güvenli bir yapı oluşturmak için öncelikle yapılması gereken, sistemlere yapılan saldırıların nasıl başarılı olduklarını anlamak, saldırganların sistemleri ele geçirmek için kullandıkları metotları öğrenmektir.

Çalışmanın ikinci bölümünde şebeke güvenliği konusunda temel bilgiler verilmiştir. Kurulum amacı internet güvenliğini arttırmaya yönelik araştırmalar yapmak olan CERT/CC (Computer Emergency Response Teams Coordination Center) organizasyonunun yapmış olduğu "Güvenlik Uygulama Döngüsü" çalışması tanıtılmış ve bu döngünün her bir adımında yapılması gereken işlemler belirtilmiştir.

Gene bu bölümde bilgisayar sistemlerine yapılan en genel 5 saldırı metodu incelenmiş, bunlara karşı alınabilecek önlemler üzerinde durulmuştur. Burada anlatılanlar sıkça karşılaşılan saldırı metotlarıdır ve günümüz işletim sistemleri bu saldırıların büyük çoğunluğu için çeşitli önlemler almışlardır. Bununla beraber bazı türde saldırılara karşı doğrudan önlem almak imkansızdır. Sistemleri bu tür saldırıları en az zararla atlatacak şekilde yapılandırmak gerekmektedir. Bu bölümün amacı, sistem zayıflıkları, bu zayıflıkların ne gibi tehlikeler oluşturabileceği ve bu tehlikelerin nasıl önlenebileceği ya da zararın nasıl en düşük seviyeye düşürülebileceği konusunda temel bir bakış açısı sunmaktır.

Üçüncü bölümde kriptoloji konusunda genel bilgi verilmiş ve kriptolojinin bir alt kolu olan kriptografinin 4 temel amacından bahsedilmiştir. Daha sonra simetrik ve asimetrik anahtar algoritmaları incelenmiş, bunların birbirlerine göre güçlü ve zayıf tarafları anlatılmış ve bu algoritmaları kullanan çeşitli şifreleme yöntemleri örnek verilmiştir. Simetrik ve asimetrik anahtar algoritmalarının güçlü yönlerinden oluşan karışık algoritmalar kullanılarak, az işlem gücüne ihtiyaç duyan, hızlı, yüksek güvenilirlikli yapılar kolaylıkla gerçekleştirilebilmektedir.

Gene bu bölümde, bilginin doğru kişiden geldiğini ispatlayan, belgelere resmiyet kazandıran sayısal imza kavramı anlatılmıştır. Uzun belgelerin tamamını imzalamak yerine, belgeyi geri dönüşümsüz bir çırpma fonksiyonundan geçirmek, belgeye has sabit uzunlukta bir veri elde etmek ve bu bilgiyi imzalayıp iletmek zamanın ve işlem gücünün daha etkili kullanılmasını sağlamaktadır.

Dördüncü bölümde akıllı kartlar konusunda ayrıntılı bir çalışma yapılmıştır. Akıllı kartlar kırmık tiplerine göre bellek kartları ve mikroişlemcili kartlar, iletişim arayüzlerine göre de kontaklı, kontaksız ve kombi kartlar olarak sınıflandırılırlar.

Kontaklı kartlarda altısı aktif olarak kullanılan toplam 8 kontak yüzeyi vardır. Kullanılan kontaklar besleme gerilimi, saat girişi, programlama gerilimi, ilklendirme işareti girişi, giriş/çıkış ve topraklama arayüzleridir.

Akıllı kartlarda veri iletişim protokolleri kırmık tiplerine bağlı olarak farklılık göstermektedir. Bellek kartları kırmığa bağlı senkron iletişime sahiptirler ve oturum boyunca kontrol tamamen terminaldedir. Kart, terminalin gönderdiği komutların gereğini yapar. Mikroişlemcili kartlarda asenkron veri iletişimi kullanılmaktadır ve oturum boyunca kontrol, kart ve terminal arasında ortaktır. Kart, terminalin gönderdiği komutu alır, kendi işletim sistemi ve ilgili uygulama programının oluşturduğu sonucu terminale iletir.

Asenkron iletişim protokolleri "T=n" şeklinde ifade edilir. Burada n protokol kodudur ve 1 ile 16 arasında olabilir. Dolayısıyla 16 farklı akıllı kart asenkron iletişim protokolü vardır. Akıllı kart bu protokollerden en az birini desteklemek zorundadır. Terminal karta ilklendirme işareti gönderdiğinde kart, desteklediği protokolleri ve iletişim parametrelerini belirten bir ilklendirme cevabı (ATR – Answer To Reset) iletir. Terminal bu bilginin hemen ardından protokol tipi seçimi (PTS – Protocol Type Selection) paketini göndererek iletişimde kullanmak istediği protokol ve parametreleri bildirir.

Akıllı kartların uygulama katmanı komut yapıları (APDU – Application Protocol Data Unit) alt katmandaki iletişim parametrelerinden bağımsızdır. Uygulama katmanı verileri 5 baytlık zorunlu bir başlık ve komuta bağlı uzunlukta olabilen bir gövde kısmından oluşmaktadır.

Mikroişlemcili kartların bellek kullanımını kontrol eden, giriş/çıkış işlemlerini yöneten, uygulamaları çalıştıran, kendilerine has dosya yapıları olan sınırlı kapasiteli işletim sistemleri vardır. Bilgi ve uygulama içeren dosyalar üzerinde işlemler yapmak, yine işletim sistemi tarafından sağlanan dosya yönetim komutlarıyla mümkün hale getirilmiştir.

Bazı işletim sistemleri, kart üretimi aşaması sonrasında da çeşitli uygulamaların karta yüklenebilmesine imkan vermektedir. Günümüzde ismi en fazla duyulan uygulama geliştirme ortamları olan Java Card ve MULTOS işletim sistemleri, dördüncü bölümün son kısmında anlatılmıştır.

Beşinci bölümde akıllı kartlar kullanılarak güvenli haberleşme yapan bir uygulama geliştirilmiştir. Uygulamada Gemplus firması tarafından üretilen kartlar kullanılmıştır. Bu kartlarda MPCOS (Multi-application Payment Chip Operating System) işletim sistemi bulunmaktadır.

Bu bölümde MPCOS işletim sisteminin dosya yapıları incelenmiş, uygulamada kullanılan anahtar dosyaları ve gizli kod dosyaları anlatılmıştır. MPCOS işletim sisteminde her dosya, gizli kodlar ve 3DES (Triple Data Encryption Standard) şifreleme algoritmalarıyla korunmaktadır. Bir dosyaya erişirken dosya tanımlayıcısında belirtilmiş olan erişim koşulları sağlanmalıdır.

Yine bu bölümde uygulamanın geliştirilme aşamaları anlatılmıştır. Akıllı kart güvenlik uygulaması temel olarak iki adımdan oluşmaktadır. Birinci adımda, uygulamada kullanılacak olan kartın, ihtiyaç duyulan dosyaları içerecek şekilde konfigüre edilmesi işlemleri yapılmıştır. Bu adımda kartlara, tarafların birbirlerini doğrulamaları esnasında kullanacakları bir anahtar dosyası, gönderilen verileri DES (Data Encryption Standard) şifreleme algoritmasından geçirmeleri istendiğinde kullanacakları DES anahtarının bulunduğu bir veri dosyası ve bu dosyayı yetkisiz kişilerin kullanmasını önlemek için bir gizli kod dosyası eklenmektedir.

Uygulamanın ikinci adımında ise, birinci adımda oluşturulan kart kullanılarak tarafların istedikleri aralıklarla birbirlerini doğrulayabildikleri ve yine istedikleri anlarda şifreli haberleşmeye geçebildikleri kullanıcı arayüz bölümü yazılmıştır.

İki adımda oluşturulan bu güvenlik modülü, şebeke haberleşmesi ve dosya transferi yapan bir kod ile birleştirilmiş, güvenli şebeke haberleşmesi ve dosya transferi yapan bir uygulama elde edilmiştir.

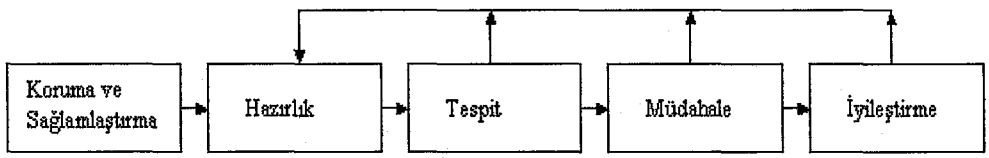
2. ŞEBEKE GÜVENLİĞİ

Teknolojinin gelişimi bilişim sistemlerine olan bağımlılığımızı arttırmakta ve bizleri sistemlerde meydana gelebilecek arıza ve saldırılara karşı daha duyarlı hale getirmektedir. Sistemlere yapılan saldırılar başarılı olmaları halinde şirketleri önemli miktarda zarara uğratabilmektedir.

Sürekli yeni saldırı metotlarının geliştirilmesi ve bunların saldırganlar arasında hızlıca yayılması, kullanıcıların yeni ve gösterişli özellikleri güvenlik becerilerine tercih etmeleri, bilişim güvenliği konusundaki eğitilmiş uzman eksikliği, bilişim suçlarının her yıl neredeyse ikiye katlanacak biçimde artmasına neden olmaktadır.

2.1 Güvenlik Uygulama Döngüsü

Uluslar Arası Bilgisayar Acil Durum Müdahale Ekipleri Koordinasyon Merkezi (Computer Emergency Response Teams Coordination Center – CERT/CC), bilgisayar ağları ve sistemlerini korumak için gerçekleştirilmesi gereken işlemleri beş ana başlık altında toplamıştır. CERT/CC, Şekil 2.1'de gösterilen bu gruplamayı yaparken geçmiş yıllarda rapor edilen saldırıları ve zayıflıkları temel almıştır.



Şekil 2.1 Güvenlik uygulama döngüsü

2.1.1 Koruma ve sağlamlaştırma

Bu ilk adımda sistemlerin ve ağın güvenliğini arttırmaya yönelik faaliyetler gerçekleştirilir. Yaygın biçimde bilinen saldırılara karşı önlemler alınır. Diğer

adımlar bu adımın sonucunda ulaşılan düzeye göre geliştirileceğinden bu adımın etkin biçimde planlanması ve gerçekleştirilmesi çok önemlidir.

Koruma ve sağlamlaştırma adımında tüm sistemlerin ve aktif şebeke cihazlarının yazılım güncellemeleri gerçekleştirilmeli, ilgili tüm yamalar uygulanmalıdır. Bilgisayar sistemlerinin üzerinde kullanılması zorunlu olanlar dışındaki yazılımlar kaldırılmalıdır. Aynı şekilde ağ hizmet sunucuları üzerinde işleyen sunucu yazılımları incelenmeli ve bunlardan gerekli olmayanlar kaldırılmalıdır. Çalışan her sunucu yazılımı, ağ üzerinden gelebilecek saldırılar için potansiyel yeni bir yetkisiz giriş noktasıdır.

Aktif şebeke cihazlarının ve bilgisayar sistemlerinin ön tanımlı parolaları değiştirilmelidir. Ön tanımlı bu bilgiler saldırganlar için çok kolay bir giriş noktası olabilmektedir.

Kullanıcılara ve sunucu yazılımlarına mümkün olan en az yetki atanmalı, bu yolla kullanıcıların ya da yazılımların art niyetli davranışlar göstermesi riski en aza indirilmelidir.

Parola temelli kullanıcı doğrulama mekanizmaları bir çok sistem için en zayıf noktalardan birisidir. Parolalar yerine kullanılabilecek farklı kullanıcı doğrulama mekanizmaları uygulama güvenliğini önemli ölçüde arttıracaktır. Kimlik tanıtımında akıllı kartların kullanılması bu konuda oldukça güzel ve güncel bir örnektir.

Yedekleme ve geri yükleme konusunda uygun bir çözüm oluşturulmalı ve belirli aralıklarla geri yükleme denemeleri yapılmalıdır. Yedeğin bir kopyasının kurum dışında saklanması, doğal afetlere karşı verilerin korunmasını sağlayacaktır.

Hizmet sunucu yazılımlarında, yazılım üreticileri ve güvenlik uzmanları tarafından tavsiye edilen güvenlik ayar şablonlarından faydalanılmalıdır.

Büyük şebekeleri, birbirlerinden bağımsız küçük parçalara bölerek yapılacak olan denetim daha başarılı bir biçimde gerçekleştirilecektir. Bu yöntemle alt şebekelerden birisine yapılan saldırıdan tüm yapının etkilenmesi önlenmiş olacaktır.

Oluşturulan sistem, zayıflık tarama testlerinden geçirilerek zayıf yönleri keşfedilmeli ve gerekli önlemler alınmalıdır. Zayıflık tarama yazılımları, bilinen sistem

zayıflıkları veritabanı barındıran ve bu zayıflıkların test edilen sistemde var olup olmadığını denetlemek için kullanılan yazılımlardır. Bu yazılımlar kullanılarak gereksiz yere sistemde bulunan açık portlar ve servisler kolaylıkla gözlemlenebilmektedir.

2.1.2 Hazırlık

Koruma ve sağlamlaştırma adımında alınan önlemlere rağmen bilinmeyen bir takım zayıflıklar nedeniyle güvenlik ihlalleri gerçekleştirilebilmektedir. Bu nedenle bilinmeyen saldırılara karşı sistemi koruyacak çalışmalar yapılmalıdır.

Hazırlık aşamasında, bilinmeyen saldırıların tespit edilmesi ve bu saldırılara zamanında müdahale edilebilmesi için gerekli olan hazırlıklar yapılır. Koruma ve sağlamlaştırma safhasında bilinen saldırılar için alınmış olan önlemler ile bu adımda gerçekleştirilen faaliyetler farklıdır. Saldırıları ve şüpheli durumları tespit edebilmek için yeterli verinin güvenilir bir biçimde toplanmasına imkan veren düzenekler hazırlık aşamasında kurulmalıdır.

Korunmak istenilen sistemin karakteristik özellikleri belirlenmelidir. Şebeke trafiğinin, sistemler üzerinde çalışan yazılımların, kullanıcı davranışlarının, dosyaların ve dizinlerin karakteristik özellikleri belirlendiğinde bu karakteristiğin dışında kalan durumların tespit edilmesi ve ele alınması mümkün hale gelir.

Şebeke ve sistemler üzerinde toplanan kayıtlar, şüpheli durumların tespit edilebilmesi için önemli bir kaynak oluşturmaktadır. İşletim sistemleri ya da uygulama yazılımları tarafından üretilen bu kayıtlar merkezi ve güvenli bir yerde toplanmalıdır. Kayıtların saldırganların ulaşamayacağı bir yerde tutulması önemlidir.

Yazılımlar için uygulanan kayıt politikası aktif şebeke cihazlarına da uygulanmalıdır. Günümüzde aktif şebeke cihazlarının çoğu, harici bir sunucu üzerinde kayıt tutma özelliğine sahiptir.

Sunucu ve ağ temelli saldırı tespit sistemleri kurularak şebeke üzerinden gelen saldırıların tespit edilmesi yine bu adımda gerçekleştirilmektedir.

Bilgisayar sistemleri üzerinde kurulabilecek olan dosya bütünlük denetleyicileri ile dosya ve dizinlerde olağan dışı değişikliklerin olup olmadığı tespit edilebilir. Bu

denetleyiciler dosya ve dizinlerin çırpma değerlerini oluşturur ve bunları güvenli bir ortamda saklar. Belirli aralıklarla bu değerler tekrar oluşturulur, eski değerlerle karşılaştırılır ve değişiklik olup olmadığı tespit edilir.

Saldırganları yanıltmak için, kolay hedef olarak görünen fakat önemli bir işlevi olmayan ve önemli bilgi bulundurmayan tuzak sunucular kurulabilir. Bu sunuculara yapılan saldırılar saldırı veritabanına kaydedilmelidir.

2.1.3 Tespit

Hazırlık adımında gerçekleştirilen hazırlıklar sonucunda şüpheli bir durum tespit edildiğinde bu adım uygulamaya alınır.

Güvenlik ihlallerini hatasız bir şekilde tespit edebilmek için, kullanılan veri kaynaklarının hatasız olduğundan emin olunmalıdır. Başarılı bir saldırı sonrası saldırgan tutulan kayıtları silmeyi ya da değiştirmeyi deneyebileceği gibi, kayıt tutan yazılımları da değiştirmeye çalışabilir. Kayıt tutan yazılımlar belirli aralıklarla bütünlük denetleyicileri ile denetlenmelidir.

Sistem kayıtlarının silinmesi durumunda bir saldırganın varlığı kolayca anlaşılabilir. Bu nedenle bir çok saldırgan tüm sistem kayıtlarını silmek yerine yalnızca kendisine ait olan kayıtları silmeye çalışmaktadır. Kayıtların kronolojik sırasında bozukluklar olması ya da uzun bir süre hiç kayıt tutulmamış olması bir saldırganın varlığına işaret ediyor olabilir.

Tespit adımında şüpheli durumlar değerlendirilir, saldırı olduğu belirlenen eylemler hakkında detaylı inceleme ve çözümleme işlemleri yapılmak üzere bir sonraki adıma geçilir.

2.1.4 Müdahale

Bu adımda, tespit edilen saldırıların detaylı incelemeleri gerçekleştirilir ve kurum politikasına uygun biçimde ele alınmaları sağlanır. Tam olarak ne olduğunun anlaşılması için detaylı analizlerin yapılması, gerekli ise saldırıya ilişkin bilgilerin kurum içi ve kurum dışı gruplar ile paylaşılması, delillerin toplanması, hasarın

artmasının önlenmesi, en kısa sürede saldırı öncesi duruma dönülmesi ve saldırının izlerinin temizlenmesi bu adımda gerçekleştirilen faaliyetlerdir.

Saldırı ile ilgili üretilen kayıtlar incelenerek gerçekleşen olayın tam olarak ne olduğu konusunda bir sonuca varılabilir. Bilgisayar disklerinin, çalışmakta olan süreçlerin, şebeke trafik kayıtlarının incelenmesi, saldırı ile ilgili veri kümesinin genişletilmesine yardımcı olacaktır.

Saldırının şebeke ve sistemler üzerindeki etkisi tespit edildikten sonra atak izleri temizlenmeli ve saldırgan tarafından yapılmış olan değişiklikler geri alınmalıdır. En kısa sürede saldırı öncesi duruma dönülmelidir. Güncel yedekler ve bütünlük denetleme yazılımları bu aşamada ihtiyaç duyulan araçlardır.

Saldırının başarılı olmasına sebep olan zayıflık tespit edilmeksizin atak öncesi duruma dönülmesi halinde saldırının aynı şekilde tekrar gerçekleşmesi riski olacaktır. Bu tür durumlarda kurum tercihleri önceden belirlenmiş, acil durum planları önceden yapılmış olmalıdır.

2.1.5 İyileştirme

Bu adımda, diğer adımlarda edinilen bilgi ve deneyimlerin ışığında şebeke ve sistem güvenliğinin arttırılmasına, tespit mekanizmalarının iyileştirilmesine ve müdahale süreçlerinin etkinliğinin arttırılmasına çalışılmalıdır.

Koruma ve sağlamlaştırma adımında kurulan yazılımlar ve uygulanan yamalar zamanla güncelliğini kaybedecektir. Bu adımda, sistemdeki yazılımlar düzenli olarak güncellenmeli, en son yamalar uygulanmalı, problemli güncellemeler geri alınmalıdır. Güncellemelerin yakından takibi için yazılım üreticilerinin sağlamış olduğu güncelleme takip yazılımları ve duyuru listeleri kullanılmalıdır.

Yine bu adımda, veri toplama ve incelemede kullanılan düzenek ayarları ve yazılım bileşenleri güncellenmeli, veri kaynakları arttırılmalıdır.

Koruma ve sağlamlaştırma adımının sonunda çalıştırılan zayıflık tarama yazılımları, iyileştirme adımı kapsamında da düzenli olarak çalıştırılmalıdır.

Kurumsal güvenlik politikası ve buna bağlı olarak belirlenmiş olan prosedürler de belli aralıklarla gözden geçirilmeli ve yenilenmelidir.

2.2 Bilgisayar Sistemlerine Yapılan Saldırı Metotları

İnternet, bilgi paylaşımı açısından eşsiz bir kaynak olmasının yanında bir çok kötü niyetli kullanıcının bulunduğu güvensiz bir ortamdır. Bu çalışmada internet üzerinde en sık karşılaşılan saldırı metotları incelenmiştir:

- Taklit etme atakları (Spoofing): Bir başkasının kimliğine bürünme işlemidir.
- Oturum ele geçirme atakları (Session Hijacking): Yetkili bir kullanıcının sunucu ile kurduğu oturumu ele geçirme işlemidir.
- Hizmetin aksatılması atakları (Denial of Service): Bir sunucu sistemi aşırı derece meşgul ederek diğer istemcilere hizmet vermesini engelleme işlemidir.
- Bellek taşması atakları (Buffer Overflow): Üzerinde program çalıştıran sunuculara anormal uzunlukta parametre girilerek sunucu belleğini bozmaya yönelik yapılan saldırılardır.
- Parola atakları: Sistemlerde kullanılan zayıf parolaları ele geçirme amacıyla yapılan ataklardır.

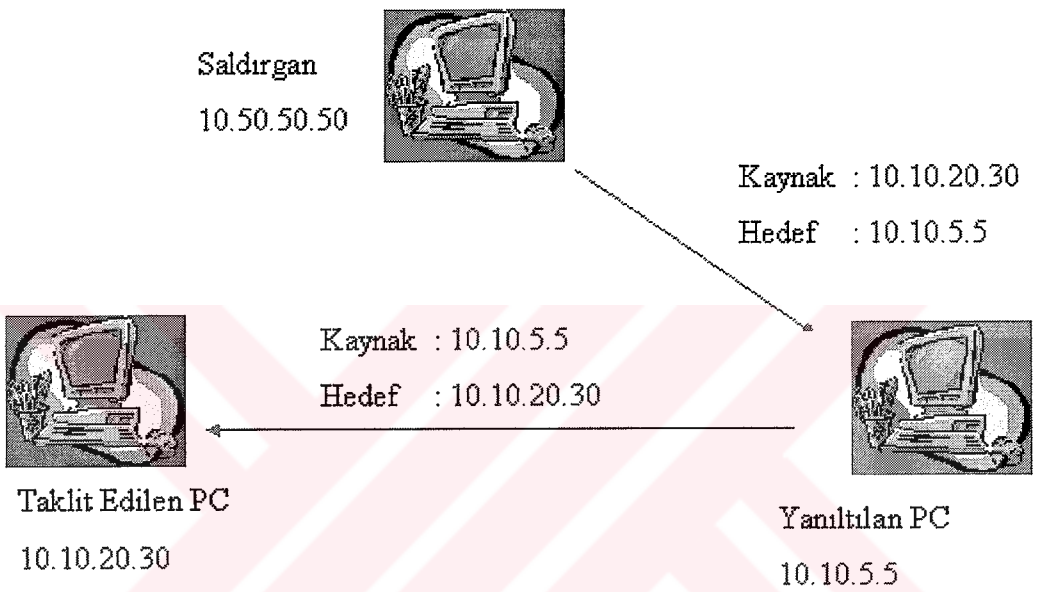
2.2.1 Taklit etme atakları (spoofing)

Taklit etme atağı, bir başkasının kimliğine bürünme işlemidir. Üç çeşit bilgisayar tabanlı taklit etme atağı vardır:

- IP Adresi Taklit Etme (IP Spoofing)
- Elektronik Posta Adresi Taklit Etme (E-mail Spoofing)
- Web Sayfası Taklit Etme (Web Spoofing)

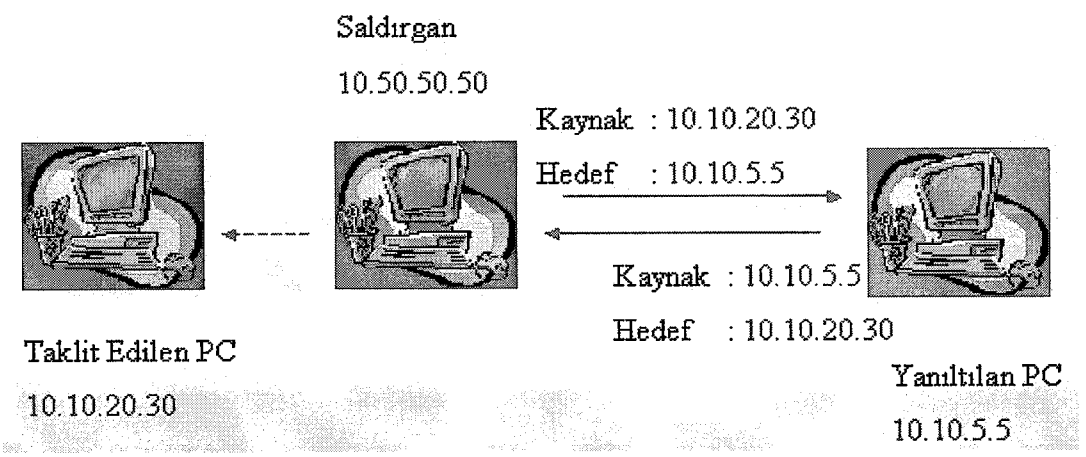
IP adresi taklit etme atağı :

Taklit etme atakları arasında en çok bilinen atak biçimidir. Saldırgan, kurbana gönderdiği paketlerdeki kaynak IP adresini güvenilen bir bilgisayarın IP adresiyle değiştirir. Dolayısıyla kurban bu paketi gönderen bilgisayara güvenir ve cevabını taklit edilen bilgisayara iletir. Yani saldırgan kurbanın cevabını elde edemez. Şekil 2.2'de gösterilen bu tipte ataklara tek yönlü ataklar denir. Bu atak hizmetin aksatılması amaçlı olarak kullanılabilmektedir.



Şekil 2.2 IP adresi taklit etme atağı

Eger saldırgan, Şekil 2.3'te gösterildiği gibi taklit ettiği bilgisayar ile kurban arasına girebilirse, kurbanın gönderdiği cevap paketlerini elde edebilir.



Şekil 2.3 IP adresi taklit etme atağında saldırganın araya girmesi durumu

Araya girme işlemi iki şekilde olabilir:

- Birinci yöntemde saldırgan gerçekten iki bilgisayar arasına girer. Fakat internette fiziksel olarak iki bilgisayar arasına girmek çok zordur çünkü IP adresleri özel kurumlar tarafından dağıtılmaktadır. Üstelik internet dinamik bir ortamdır. Yani istenilen IP adresi elde edilse bile, kurbanın göndereceği bir sonraki paket farklı bir yoldan hedefe varabilir.

- İkinci yöntem olarak, TCP/IP standardı dahilinde olan "kaynak yönlendirme" özelliği kullanılabilir. Saldırgan TCP/IP paketine sahte kaynak adresi yanısıra paketin alacağı yol bilgisini de ekler. Paket sırasıyla belirtilen bu IP adreslerinden geçer ve kurbana varır. Kurban da TCP/IP protokolü gereği olarak aynı yol üzerinden cevap gönderir. Bu yöntemle saldırgan sırayla geçilmesi gereken en çok 8 IP adresi verebilir. Bu adreslerden bir tanesini kendi adresi olarak veren saldırgan iki bilgisayar arasına kolaylıkla girebilir.

Temel taklit etme ataklarından şebekeyi korumak için yönlendiricilere iki çeşit filtre eklenmektedir :

- Giriş filtresi : İnternetten gelen TCP/IP paketleri içinde kaynak adresi olarak, bulunulan şebekeye ait bir adres gözüküyorsa bunlar büyük ihtimalle IP adresi taklit edilmiş paketleridir ve yönlendiriciler bu paketleri yok etmelidir.

- Çıkış filtresi : Şebekedeki bir bilgisayar internete, bulunulan şebekeye ait bir adrese TCP/IP paketleri gönderiyorsa bu paketler yönlendiriciler tarafından yok edilmelidir.

Bu iki çeşit filtre, taklit edilen bilgisayar ve kurban bilgisayar aynı şebekede ise işe yarayacaktır. Yönlendiricilerde kaynak yönlendirme özelliği iptal edilerek bu yönde gelen ataklardan tamamen korunmak mümkündür.

Elektronik posta adresi taklit etme atağı :

Bir başkası adına elektronik posta gönderme işlemidir. 3 şekilde yapılabilir:

- Benzer elektronik posta adresi kullanmak

- Elektronik posta istemci programı modifiye etmek
- SMTP (Simple Mail Transfer Protocol) portuna doğrudan telnet bağlantısı yapmak

Birinci yöntemde saldırgan, güvenilen bir kişi adına elektronik posta adresi alır ve bu adresten bir elektronik posta gönderir. Mesajı alan kişiler mesaj kaynağı konusunda yeteri dikkati göstermezlerse değerli birtakım bilgileri bu mesaja cevap olarak gönderebilirler.

Bu zayıflığa çözüm olarak organizasyonlar kendi elektronik posta sunucularını kurmalı, tüm önemli haberleşmeler bu sunucu üzerinden yapılmalı, sunucu yerel ağdan ve internetten ulaşılabilir olmalıdır.

Bir diğer çözüm olarak açık anahtar teknolojisinin getirdiği dijital imzalar kullanılabilmektedir. Gönderen taraf gizli anahtarı ile mesajını imzalar, alıcı taraf mesajı gönderenin açık anahtarıyla imzayı çözer ve gönderenin kimliğinden emin olur.

Elektronik posta sunucuları, mesaj okumak için bir istek geldiğinde kendilerine bağlanan kullanıcıların kimlik bilgilerini doğrular, daha sonra mesajların okunmasına izin verir. Fakat mesaj gönderirken kimlik doğrulama işlemi yapılmamaktadır. Bu yüzden bir elektronik posta istemci programı modifiye edilerek istenilen bir elektronik posta adresi adına şifreye gerek duymaksızın mesaj atılabilmektedir.

Bu tür ataklardan korunmak için çeşitli politikalar uygulanabilmektedir: Mesajı gönderen tarafın IP adresi kontrol edilebilir, sürekli tutulan kayıtlarda bu tür atak yaptığı farkedilen kişilere yaptırımlar uygulanabilir, gönderen tarafın mesajı imzalaması istenebilir.

Elektronik posta istemci programlarının yaptığı işlem, sunucunun SMTP portuna telnet bağlantısı yapmak, ilgili komutları göndererek elektronik posta göndermek ve almaktır. Bir elektronik posta istemci programı kullanmadan doğrudan sunucunun SMTP portuna telnet ile bağlanılarak da elektronik posta gönderme işlemi yapılabilir.

Bu ataklar tamamen önlenemezler. Gerekli politikalar uygulanarak, uğranılacak zarar en aza indirilebilir.

Web sayfası taklit etme atağı :

Web sayfası taklit etme atağı 3 şekilde yapılabilir:

- Başkası adına web adresi alınabilir
- Benzer bir isme bağlantı verilebilir
- URL (Uniform Resource Locator) yönlendirme metodu kullanılabilir

Önce başvuruda bulunan kişi istediği web sayfası adresini kiralayabilmektedir. Bu adres bir alışveriş merkezinin ya da bankanın ismi olabilir. Böyle bir durumda bu adrese bağlanan kişiler doğru yerde olduklarını sanarak kredi kartı numaralarını ya da şifre bilgilerini girebilirler. Bu durumda bu bilgileri çalınmış olacaktır.

Saldırgan mutlaka bu ismi ele geçirmek zorunda değildir. Örneğin bir bankanın web sayfası "www.banka.com" olsun. Saldırgan "wwwbanka.com" aktif bağlantısını taşıyan bir elektronik postayı kurbanlara gönderip, yeterince dikkat göstermeyen kişilerin kendi sahte sitesine bağlanmalarını bekleyebilir.

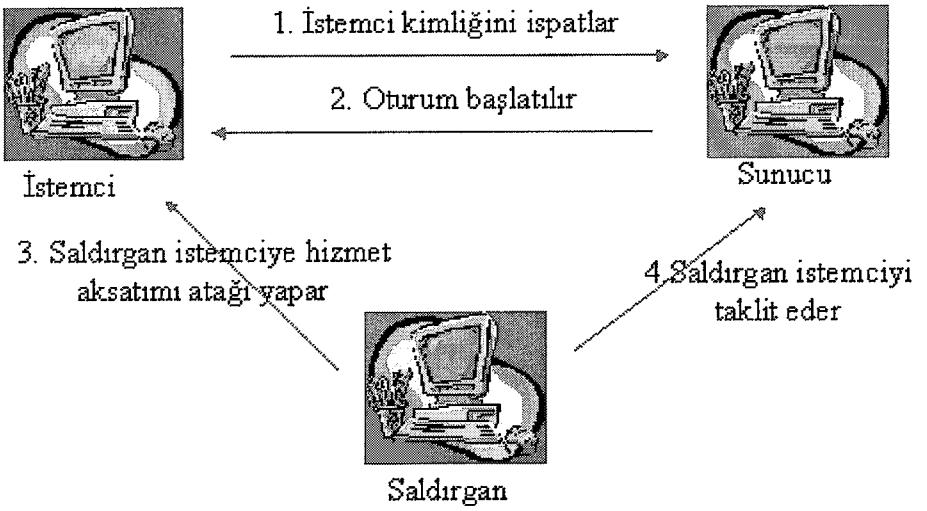
URL Yönlendirme metodunda kurban saldırganın bilgisayarına bağlanınca saldırgan kurbanın asıl bağlanmak istediği sunucuya bağlanır ve gelen bilgiyi kurbana yönlendirir. Böylece kurban istediği sayfalara gidecektir fakat saldırgan her zaman aradaki haberleşmeyi dinliyor olacaktır.

Web sayfası taklit etme ataklarına karşı korunmak için en iyi yöntem sunucu taraflı sertifikalar kullanmaktır. Özel bilgilerin verileceği sayfalar güvenli sayfalar olmalıdır. Güvenli sayfalar bir sertifika otoritesi tarafından verilmiş olan sertifikalara sahiptirler. Böyle bir sertifikaya sahip olmayan web sayfalarında kesinlikle özel bilgiler gönderilmemelidir.

2.2.2 Oturum ele geçirme atakları (session hijacking)

Oturum ele geçirme atağı aşamaları Şekil 2.4'te gösterilmiştir. İstemci sunucuya kendi doğruluğunu ispatlar ve sunucu ile istemci arasında bir oturum başlatılır. Bu andan itibaren saldırgan harekete geçer, istemciyi hizmetin aksatılması ataklarıyla etkisiz hale getirir ve bu istemciyi taklit ederek oturumu ele geçirir. Bu sayede

saldırgan doğrulama işlemini atlayarak sisteme girmiş olur. Doğrulama koşulunu yalnızca oturum başlangıcında arayan sunucular bu tarz ataklara karşı duyarlıdır.

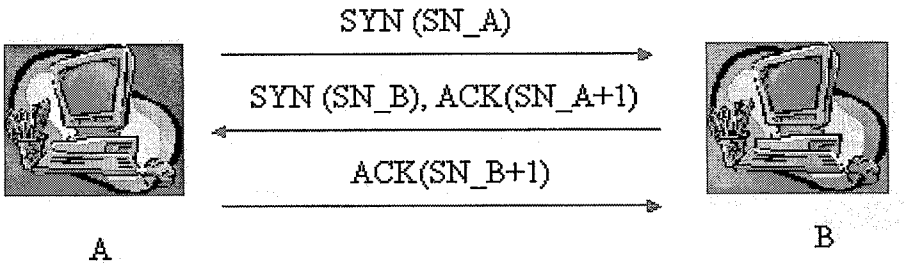


Şekil 2.4 Oturum ele geçirme atağı aşamaları

Oturum ele geçirme atağında, taklit etme ataklarından farklı olarak, her üç taraf da (saldırgan, yanıltılan ve taklit edilen taraflar) aktif olarak rol oynamaktadır. Oturumu ele geçiren saldırgan, sunucu üzerinde o oturum dahilinde taklit ettiği bilgisayarın tüm haklarına sahip olur.

TCP/IP protokol özeti

TCP/IP bağlantı yönelimli bir protokoldür, gönderen tarafın paketin alındığından emin olması sağlanır. Bunun için öncelikle bir bağlantı kurulur. Bağlantının kurulması için Şekil 2.5'te gösterilen 3 yönlü el sıkışma işlemi gerçekleştirilir.



Şekil 2.5 TCP/IP 3 yönlü el sıkışma protokolü aşamaları

3 yönlü el sıkışma protokolünün aşamaları şunlardır:

- A bilgisayarı, senkronizasyon biti (SYN – Synchronization Bit) 1 yapılmış bir paketi B'ye gönderir. Bu paket bir bağlantı isteğidir ve aynı zamanda A için rasgele seçilmiş bir paket dizisi başlangıç numarası (SN-A – Serial Number of A) içermektedir.

- B bu paketi alınca, SYN biti 1 yapılmış olan ve B rasgele seçilmiş bir paket dizi başlangıç numarası (SN-B – Serial Number of B) içeren yanıt paketini A'ya gönderir. A'nın bir önceki adımda gönderdiği pakete bir cevap olarak gönderildiği için bu paketin bilgi (ACK - Acknowledge) biti de 1 yapılmıştır. Bu paket ayrıca A'nın paket seri numarasını bir arttırarak oluşturulmuş bilgiyi içermektedir. (SN-A + 1)

- A bu paketi alınca bunu B'nin seri numarasını bir arttırdığı bir bilgi paketiyle (SN-B + 1) cevaplar. Bu andan itibaren A ve B arasında bağlantı yönelimli bir oturum başlatılmıştır.

Paket seri numaraları alıcı tarafta paketlerin doğru bir şekilde sıralanmasını sağlamaktadır. Yine bu numaralar kullanılarak, gönderen tarafa, paketlerin doğru biçimde alıcıya ulaştığı bilgisi verilir. Alındığı bildirilmeyen paketler bir süre sonra tekrar gönderilmektedir.

Oturum ele geçirme atağının aşamaları şunlardır :

- Hedef bulmak
- Hedefi bir süre izlemek
- Aktif bir oturum bulmak
- Paket seri numaralarını tahmin etmek
- Taraflardan birini saf dışı bırakmak
- Saf dışı bırakılan tarafı taklit ederek oturumu ele geçirmek

Oturum ele geçirmeye kalkışmadan önce hedef bir süre izlenmelidir. Bu sürede TCP/IP paketlerinin seri numaraları takip edilir. Bu işlem, daha sonraki adımlarda paket seri numarası tahmin etmekte işe yarar.

İki bilgisayarın TCP/IP protokolünü kullanarak haberleşebilmesi için IP adresleri, port numaraları ve paket seri numaraları gerekmektedir. IP adresleri ve port numaralarını bulmak kolaydır. Çünkü IP paketlerinde bu bilgiler vardır ve oturum sonlanana kadar değişmez. Fakat paket seri numaraları sürekli olarak değişmektedir. Eğer saldırgan, karşı tarafın beklediği seri numarasını doğru tahmin edebilirse, bu numarayla ve taklit edilen bilgisayarın bilgileriyle bir paket gönderirse, yanıltılan bilgisayar bu pakete cevap verecektir. Bu andan itibaren saldırgan oturuma sızmış olur.

Bu aşamada saldırgan taklit ettiği bilgisayarı saf dışı bırakmalıdır. Bunun için en iyi yöntem hizmetin aksatılması atakları yapmaktır. Saldırgan, safdışı bıraktığı bilgisayarın bilgilerini kullanarak yanılttığı bilgisayar ile haberleşmesini sürdürür. Oturum artık el değiştirmiştir.

Oturum ele geçirme işlemi karmaşık gözükse de bu işi yapan çeşitli programlar vardır. Bu programları kullanarak fazla bilgi sahibi olmayan bir kullanıcı bile bir oturumu kolaylıkla ele geçirebilmektedir.

Oturum ele geçirme atağına karşı korunmada şu yollar izlenebilir:

- Şifreleme kullanılması durumunda, saldırgan oturumu ele geçirse bile yanıltılan bilgisayardan bilgi elde edemez ya da ona hatalı bilgi gönderemez.
- Güvenli bir protokol kullanılabilir. Sunucu ile istemci arasındaki iletişimi güvenli hale getirecek sanal özel bağlantı (VPN – Virtual Private Network) protokolleri bulunmaktadır.
- Gelen bağlantılar sınırlandırılabilir.
- Güvenli bir doğrulama protokolü kullanılabilir. Normal şartlarda doğrulama protokolü kullanmanın hiçbir etkisi yoktur çünkü saldırgan zaten kurulmuş olan bir oturumu ele geçirecektir. Fakat kısa aralıklarla doğrulama işlemi tekrarlanırsa, oturum ele geçirilse bile bu durum fazla uzun sürmez.

2.2.3 Hizmetin aksatılması atakları (denial of service)

Hizmetin aksatılması atağı ile saldırgan, kaynakları aşırı derecede meşgul ederek ya da anlamsız protokol paketleri göndererek bir sistemi tamamen kullanılamaz ya da son derece yavaş hale getirebilir. Bu tür ataklardan korunmak son derece zordur. Bu atakların etkisini azaltacak yöntemler vardır ama eğer saldırganın yeterli zamanı ve kaynağı varsa muhtemelen başarılı olur.

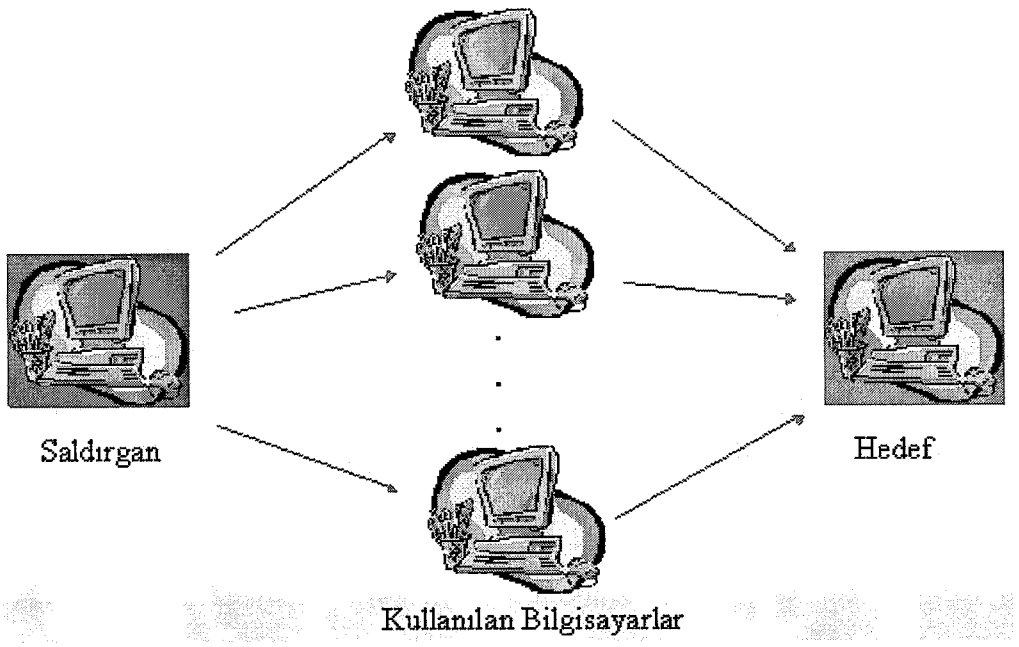
Hizmetin aksatılması atakları temel olarak iki şekilde yapılabilir :

- Saldırılan bilgisayara tahmin etmediği veri paketleri gönderilerek sistemin kilitlenmesi ya da yeniden başlatılması sağlanabilir.

- Saldırılan bilgisayara cevap veremeyeceği kadar yoğun paket gönderilip bir süre hizmet dışı kalması sağlanabilir. Saldırgan paket göndermeye son verdiği anda hedef bilgisayar normal işleyişine geri döner.

Dağıtılmış hizmetin aksatılması atağı (distributed DOS)

Normal bir hizmetin aksatılması atağında bir saldırgan bir hedefe saldırır. Dağıtılmış bir atakta ise saldırgan bir kaç makinayı ele geçirir ya da onlarla anlaşır ve hep beraber Şekil 2.6'da görüldüğü gibi bir hedefe saldırı gerçekleştirirler.



Şekil 2.6 Dağıtılmış hizmetin aksatılması atağı

Hizmetin aksatılması atak örnekleri

Hizmetin aksatılması atakları, en sık karşılaşılan ve en kolay gerçekleştirilen saldırılardır. Çeşitli yöntemlerle hizmetin aksatılması yönünde saldırı gerçekleştiren bir çok hazır program bulunmaktadır.

Ping of death :

Ping, iki bilgisayar arasındaki ağ bağlantısını test etmeye yarayan basit bir ICMP (Internet Control Message Protocol) paketidir. Bir bilgisayar diğerine ping paketi gönderir ve diğer bilgisayar bu ping paketine cevap verir. Böylece bu iki bilgisayar arasında bağlantının sağlamlığı basit bir şekilde test edilir.

Ping paketlerinin boyutu en fazla 65536 byte olabilir. Bu atakta saldırgan bu boyuttan daha büyük ping paketi gönderir. Aşırı uzun paketler karşısında ne yapacağını bilemeyen işletim sistemleri kilitlenmektedir.

Bu atağı yapmak son derece basittir. Ping komutu bulunan tüm bilgisayarlarda, paket boyutu parametresi verilerek şu şekilde uygulanabilmektedir :

```
ping –l 65540 192.168.1.10
```

Korunma amaçlı olarak işletim sisteminin en son yamalarının uygulanması gerekmektedir. Ayrıca yönlendiriciler ve ateş duvarları aşırı uzunluktaki IP paketlerini geçirmeyecek şekilde konfigüre edilmelidir.

SSPing :

Büyük boyutlu IP paketleri parçalara ayrılıp iletilir ve alıcı tarafta tekrar birleştirilirler. Alıcı tarafta bu birleştirme işleminin yapılabilmesi için bellekte bir yığın bölgesi kullanılmaktadır.

Bu atakta saldırgan uzun bir ICMP paketini ufak ufak birçok pakete bölerek gönderir. Alıcı tarafta bu parçalar tüm paket tamamlanana kadar yığın bölgesinde saklandığından, bir süre sonra bellek taşması olur. Dolayısıyla makina kilitlenir.

Bu atağa karşı korunma amaçlı olarak işletim sisteminin en son yamaları uygulanmalıdır. İşletim sistemleri bu atağa karşı TCP/IP bellek yığın bölgesini güncellemişlerdir.

Land :

Bir IP paketi başlık kısmında kaynak adresi ve portu, hedef adresi ve portu bilgileri bulunmaktadır. Haberleşme başlatmak isteyen taraf senkronizasyon bitini 1 yaparak bir paket gönderir. Bu paketi alan taraf da paketteki kaynak adresi ve port numarasını kullanarak cevap verir.

Bu atakta saldırgan, kaynak adresi ve portu ile hedef adresi ve port bilgileri aynı olan bir senkronizasyon paketi gönderir. Alıcı taraf bu paketi alır ve değerlendirir. Yanıtlayacağı bilgisayar yine kendisi olacağından bu durum karşısında bazı sistemler nasıl davranacağını bilemeyip kilitlenmektedir.

Bu atağa karşı korunma amaçlı olarak işletim sisteminin en son yamaları uygulanmalıdır. Ayrıca yönlendiriciler, kaynak adresi kendi yerel şebekesine bağlı olan paketleri geçirmeyecek şekilde ayarlanmalıdır.

Smurf :

Kendilerine gönderilen ICMP paketlerini alan bilgisayarlar bu paketlere cevap verirler. IP adresi, şebeke adresi ve bilgisayar adresi olmak üzere iki bölümden oluşmaktadır. Bilgisayar adresi kısmının tüm bitleri 1 yapılarak elde edilen adres, ilgili şebekenin yayım (broadcast) adresidir.

Bu atakta saldırgan kaynak adresi olarak hizmet aksatımı yapmak istediği bilgisayar adresini, hedef adresi olarak da bir şebeke yayım adresini verir. Bu paketi alan şebekedeki tüm bilgisayarlar kaynak adresine cevap verir. Böylece saldırıya maruz kalan sistem aşırı yoğunluktan dolayı kilitlenir.

Bu atağa karşı taklit edilen bilgisayar hiçbir şey yapamaz. Fakat kullanılan şebeke için birtakım önlemler alabilir. Şebeke yönlendiricileri, şebeke yayım adresi içeren IP paketlerini önlemelidirler. Ayrıca şebekedeki bilgisayarlar, yayım adresi ile elde ettikleri ICMP paketlerine cevap vermeyecek şekilde ayarlanmalıdırlar.

Hizmetin aksatılması ataklarına karşı alınabilecek önlemler

Hizmetin aksatılması ataklarını tamamen yok etmenin bir yolu yoktur. Aşağıdaki önlemler alınarak bu tipte atakların etkileri en aza indirilebilir:

- Sağlam tasarım: Büyük bir organizasyonun internete birçok bağlantısı olmalı ve bu bağlantılar coğrafik olarak farklı bölgelerde bulunmalıdır. Ayrıca internetten hizmet veren sunucular yedeklenmelidir.

- Bant genişliği sınırlamaları: Portlara belli oranlarda bant genişliği tahsis edilmelidir. Böylece bir porta yapılan hizmet aksatımı atağı, başka portlardan hizmet veren uygulamaları meşgul etmeyecektir.

- Sistem yamaları: Yeni bir saldırı çeşidi çıkınca işletim sistemi üreticileri eğer kendi sistemleri bu saldırıya karşı korumasız ise hemen yama programları geliştirir. Bu yama programları takip edilmeli ve sistem güncel tutulmalıdır.

- Minimum sayıda servis çalıştırma: Bir bilgisayarda ne kadar çok servis çalışıyorsa ona atak yapılabilecek zayıf noktası o ölçüde fazla olur. Ayrıca az sayıda servisi sürekli izlemek ve yönetmek çok daha kolaydır.

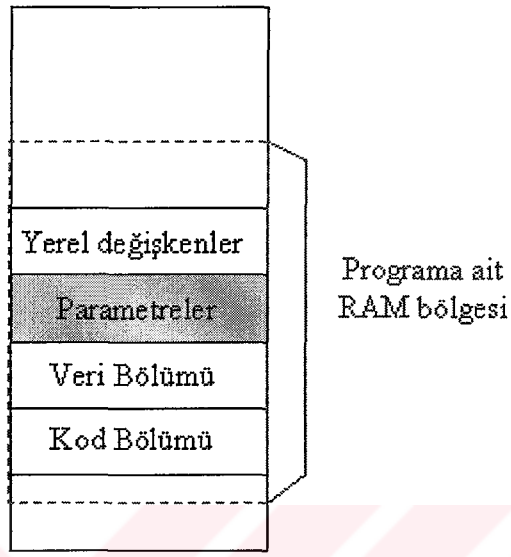
- Yalnızca gerekli trafiğe izin verilmesi: Ateş duvarlarında yalnızca gerekli trafiğe izin vermek, diğer paketleri sonlandırmak, sistemi olası tehlikelerden korur.

- Filtrelenmiş IP adresleri: Bir atak algılandığında atağı yapan IP adresleri kaydedilmeli ve yönlendiriciye tanıtılarak bu adreslerden gelen paketler için filtreler uygulanmalıdır. Her ne kadar sisteme giremese bile bu paketler bant genişliğini tüketmeye devam eder. Bu yüzden bu adresler en kısa zamanda internet servis sağlayıcısına bildirilmeli ve henüz kurumun yönlendiricisine gelmeden yok edilmelidir.

2.2.4 Bellek taşması atakları (buffer overflow)

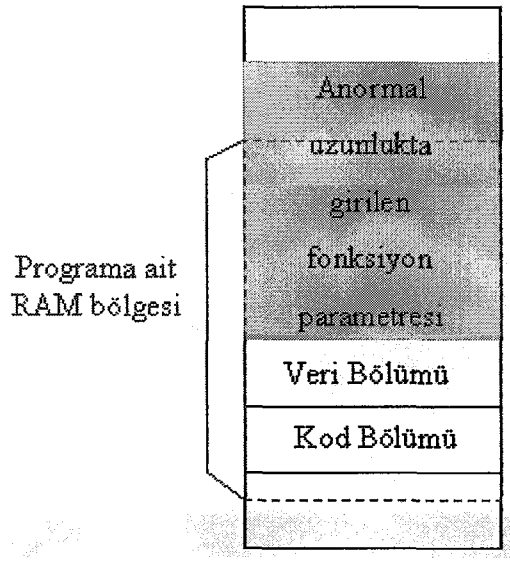
Bilgisayarlarda RAM bölgesi, çalıştırılan programların kod ve değişkenlerinin saklandığı geçici hafızadır. Bir program belleğe Şekil 2.7'de gösterildiği gibi yüklenir. Kod bölgesinde, programın makina komutları bulunur. Veri bölümünde,

programın çalışma süresi boyunca yeri ayrılmış olan global değişkenler ve diziler saklanır. Yığın bölgesinde ise yerel değişkenler, fonksiyon sonlanınca kodun döneceği bellek bölgesi ve fonksiyona geçirilen parametre değerleri bulunur. Çağırılan fonksiyon gerekli parametreleri yığın bölgesinden çeker ve kullanır.



Şekil 2.7 Normal bellek dizilimi

Program yığın bölgesine kopyalanan verinin boyutunu kontrol etmezse, aşırı uzun bilgi girişi sonrası sınırlı olan parametre bölümü yetmez ve Şekil 2.8'de gösterildiği gibi, bellekteki diğer programların verileri üzerine kaydedilme olabilir. Bellekteki programların işleyişini bozan bu hataya bellek taşması denilmektedir.



Şekil 2.8 Bellek taşması durumu

Bellek taşması atakları iki amaçla yapılmaktadır:

- Hizmetin aksatılması amaçlı : Saldırgan manasız ve çok uzun bir parametre girerek yığın bölgesini taşırır, bu taşma bellekteki programları bozar ve makina kilitlenir. Parametre kontrolü yapılmayan programlar için bu son derece kolay gerçekleştirilebilir bir atak şeklidir.

- Kod çalıştırma amaçlı : Saldırgan çalıştırılabilir makina kodu ve bu kodun başlangıcını gösteren geri dönüş bellek bölgesi adresini de içeren çok uzun bir parametre girer. Fakat bu kodu çalıştırabilmek için bellek durumunun doğru bir şekilde tahmin edilmesi gerekmektedir. Bu nedenle bu atak tipi son derece zor gerçekleştirilir. Bu tip ataklar belirli bir uygulama üzerinde yoğun bir şekilde çalışılarak, ilgili programa hangi uzunluktan başlayarak nasıl kodlar girilmesi gerektiği bulunarak yapılır. "NetMeeting Bellek Taşması", "Outlook Bellek Taşması" gibi belirli uygulamaya yönelik çeşitleri bulunmaktadır. Fazla teknik bilgisi olmayan saldırganlar bu tip saldırılarda bulunmak için kendilerine yardımcı olacak programları internette kolayca bulabilmektedir.

Bellek taşması ataklarına önlem olarak şunlar yapılabilir:

- Kullanılmayan portlar ve servisler kapatılmalıdır. Çünkü çalışan ne kadar çok servis varsa, bellek taşmasına karşı açığı olan servis bulunması ihtimali o denli fazladır.

- Kullanılan programların son versiyonları yüklenmeli ve tüm yamaları uygulanmalıdır.

- Programlar kurulmadan önce bellek taşması ataklarına göre sıkı bir şekilde test edilmelidir.

- Mümkünse programın kodları incelenmeli ve anormal parametre girişlerine karşı gerekli önlemlerin alındığından emin olunmalıdır.

2.2.5 Parola atakları

Günümüzde parola atakları oldukça sık karşılaşılan, parola sahiplerinin dikkatsiz davranmaları sonucunda kişilere ya da organizasyonlara büyük zararlar verebilen saldırılardır.

Parolanın gelişimi :

- Kullanıcı adı : Şirketler 1980'li yıllarda yeni yeni bilgisayar aldıklarında kullanıcıların kendi bilgilerini saklama ihtiyacı doğmuştur. Kullanıcı, bu tip bir sisteme kendisini tanımlayan isimle bağlanmakta, bilgilerini bu basit yöntemle korumaktadır.

- Kullanıcı adı ve parola : Zamanla bazı kullanıcıların başkalarının kullanıcı isimlerini kullanarak sisteme girdikleri ve kişisel bilgileri elde ettikleri gözlemlenmiş ve bunu engellemek amacıyla kullanıcı isimlerinin yanısıra parolalar kullanılmaya başlanmıştır.

- Sistemin atadığı şifreler : Parolalar ilk kullanılmaya başlandığı dönemlerde oldukça basit olarak verilmekteydi. Kullanıcının bir yakınının ismi, yaptığı spor, doğum tarihi parola olarak seçilebilmekteydi. Bu zayıflığın önüne geçmek için parolalar sistem tarafından, tahmin edilmesi güç bir kombinasyonla seçilerek verilmeye başlanmıştır.

- Parola politikaları : Sistem tarafından verilen parolalar, hatırlanması zor olduğundan bir yerlere kaydedilmekte ya da zaman geçtikçe unutulmaktadır. Günümüzde aşağıdaki gibi bazı politikalar belirlemek kaydıyla parolaların yine kullanıcılar tarafından belirlenmesine izin verilmektedir.

    - Parolaların minimum uzunluğu olması, en az bir rakam, bir harf ve bir özel karakter içerip anlamlı bir kelime içermemesi koşulu koyulabilir.

    - Harici bir program kullanılarak parolalar elde edilmeye çalışılıp, elde edilme durumlarında kullanıcının yeni ve tahmin edilmesi zor bir parola seçmesi istenebilir.

- Parolaların sık sık yenilenmesi istenebilir ve birkaç sefer önceki parolanın tekrar verilmesi engellenebilir. Minimum parola süresi verilerek, kullanıcıların ardarda parola değiştirip tekrar eski parolasını elde etmesi engellenebilir.

- Ardarda birkaç hatalı parola girişi sonrası kullanıcı hesabı bir süreliğine dondurulabilir.

Parolanın geleceği :

- Güvenlik donanımı kullanımı : Harici bir donanım kullanılarak şifrenin kimse tarafından bilinmemesi, yalnızca bu donanıma sahip olan kimselerin sisteme girebilmesini sağlayan metoddur. Sunucu sistem ve bu donanım ortak bir şifre içermektedir.

    - Sunucu ve istemcideki donanım tek kullanımlık parolalar üreterek çok kısa aralıklarla parolaları güncelleyebilir.

    - Sunucu istemciye rasgele bilgiler gönderir (challenge), istemci tarafindaki donanım da bu bilgi ve kendi içindeki gizli paroladan oturum parolası elde edebilir. Böylelikle asıl parola doğrudan şifreleme işlemlerinde kullanılmaz ve sürekli olarak gizli kalır.

- Biyometrik yöntemler : İnsanlara ait çeşitli ayırt edici özellikler kullanılarak geliştirilen güvenlik yöntemleridir. Biyometrik yöntemler olarak en genel kullanılan özellikler şunlardır:

    - Parmak izi tarama

    - El tarama

    - Retina tarama

    - Yüz tarama

    - Ses tarama

Parola kırma :

İşletim sistemleri kullanıcıların parolalarını tek yönlü bir çırpma işleminden geçirmekte ve sonucu parola dosyalarında saklamaktadır. Kullanıcı, tanımlayıcı ismini ve parolasını yazıp sisteme girmek istediğinde bu parola yine aynı işlemden geçirilir ve işlemin sonucu dosyadaki değerle karşılaştırılır. Değerler uyuşursa kullanıcının sisteme girmesine izin verilir. Bir şekilde bu dosyayı ele geçiren saldırgan çeşitli yöntemler deneyip kullanıcı parolalarına ulaşmaya çalışabilir. Geri dönüşümsüz bir fonksiyonla elde edilmiş olan bu değerden kullanıcı parolasını elde etme işlemine "parola kırma" denir.

En basit şekliyle parola kırma şu şekilde gerçekleştirilir:

- Geçerli bir kullanıcı adı bulunur.
- Olası parolalar listesi hazırlanır.
- Her bir aday parola yüksek olasılıktan düşük olasılığa göre sıralanır ve en yüksek olasılıklı olan denenir.
- Sistem girişe izin verirse parola elde edilmiş olur.
- Sistem girişe izin vermezse hesabın kilitlenme ihtimali de dikkate alınarak sırayla denemelere devam edilir.

Parola ele geçirme işlemi bir çok denemeye dayalı bir işlemdir. Bu nedenle bir program yardımıyla otomatik denemeler gerçekleştirilebilir. Genel algoritma şöyledir:

- Geçerli kullanıcı adları bulunur.
- Parola şifreleme algoritması bulunur.
- Şifrelenmiş parolalar elde edilir. Olası parolalar listesi hazırlanır.
- Tüm ihtimaller sırayla denenir.
- Her deneme sonrası uyuşan bir şifrelenmiş parola var mı diye bakılır.

Şifreleme algoritmasının güvenliği, tamamen algoritmada kullanılan anahtara bağlıdır. Çünkü algoritmanın gizliliği garanti edilemez. Ayrıca hemen hemen tüm işletim sistemlerinin kullandığı parola şifreleme algoritmaları bilinmektedir.

İşletim sistemleri şifrelenmiş parolalar listesini belirli klasörler altında ve bilinen dosya isimleriyle saklamaktadır. Dolayısıyla sisteme girebilen bir kimse kolaylıkla şifre dosyasını elde edebilmektedir. Bu denemeler ilgili işletim sistemi tarafından yapılmadığı için hesabın kilitlenmesi ihtimali bulunmamaktadır.

Olası parola listelerinin hazırlanması aşamasında şu yöntemler kullanılabilir:

- Sözlük atakları : Saldırgan bir sözlükteki tüm kelimeleri kullanarak kısa sürede birçok zayıf parolayı elde edebilir. Ayrıca kullanıcıların kişisel özellikleri ve ilgi alanları da dikkate alınarak sözlüğe eklemeler yapılabilir.

- Tüm ihtimaller atakları : Teorik olarak kırılamayan şifre yoktur. Sadece sağlam bir şifre verilerek çok uzun süre kırılamaması sağlanabilir. Bu atak türünde saldırgan parola vermede kullanılan karakterlerin oluşturduğu tüm kombinasyonları zamanı ve kaynakları elverdiği şekilde dener.

- Hibrit ataklar : Yukarıdaki iki atak türünün birleştirilmesiyle oluşturulan bir yöntemdir. Saldırgan sözlükteki kelimelerin önüne ve/veya sonuna ihtimaller ekleyerek kaynaklarını ve zamanını daha verimli kullanır ve parolayı ele geçirme ihtimalini arttırır.

Parola ataklarından korunmanın en etkili yolu, tahmin edilmesi zor olan ve içinde en az birer alfanümerik, rakam ve işaretin bulunduğu uzun parolaların kullanılması ve belirli aralıklarla bu parolaların güncellenmesidir.

3. KRİPTOLOJİ

Yunanca Cryptos Logos (gizli kelime) anlamına gelen kriptoloji, matematiğin şifre bilimi (kriptografi) ve şifre analizi (kriptanaliz) bölümlerini kapsayan dalıdır. Şifre biliminin amacı iletilen ya da saklanan verinin güvenliğini sağlamaktır. Şifre analizinin amacı ise var olan şifreyi çözmektir.

Şifreleme, elimizdeki veriyi, ele geçirecek olan kişide gerekli anahtar olmadan çözülmesi zor hale getirmektir. Bu zorluğun derecesi şifreleme sürecinde uyguladığımız politikaya bağlıdır. Şifre çözme ise şifrelenmiş veriyi çözüp orijinal haline getirmektir. Anahtar ise bir metni şifrelemekte ya da açmakta kullanılan veri parçasına verilen isimdir.

Kriptografi, fiziksel dünyada sahip olduğumuz güvenli ortamı elektronik dünyaya taşımamıza izin verir. Güvenli internet siteleri için gerekli olan güvenli elektronik iletişime imkan sağlar. Bu sayede insanlar internet üzerinden herhangi bir kaygı duymadan bankacılık işlemleri gerçekleştirebilmekte, kredi kartlarıyla alışveriş yapabilmekte, gizli bilgi alışverişinde bulunabilmektedir.

Elektronik posta günlük kişisel ve iş yazışmalarında vazgeçilmez bir yere sahiptir. Ancak mesajlar, herhangi bir önlem alınmadığında başkaları tarafından ele geçirilebilmekte ya da değiştirilebilmektedir. Şifreleme, elektronik postaların istenmeyen kişilerce okunabilmesini ya da içeriğinin değiştirilebilmesini engeller.

Bazı durumlarda kriptografi elektronik ortamda, fiziksel dünyadaki işlemlerde sahip olunandan daha fazla güven sağlamaktadır. Fiziksel imzalar taklit edilebilmekte ya da imzalanan bir belge değiştirilebilmektedir fakat elektronik imzanın gizli anahtarı bilinmediği sürece taklit edilmesi ya da belgenin içeriğinin değiştirilmesi imkansızdır.

Kriptografi sadece internet üzerinde değil, telefonlarda, televizyonlarda ve günlük hayatın bir çok alanında kullanılmaktadır.

Kriptografinin 4 temel amacı vardır:

- Gizlilik (confidentiality) : Orijinal veriyi gerekli anahtar olmadan açılamayacak bir şekle sokup başkaları tarafından okunmadığından emin olunmasını sağlar.
- Veri bütünlüğü (data integrity) : Verinin iletimi sırasında değişikliğe uğramadığından emin olunmasını sağlar.
- Kimlik doğrulama (authentication) : Bilginin doğru kaynaktan alındığını ispatlar.
- İnkar edememe (non-repudation) : Veriyi gönderen kişinin bunu inkar edememesini sağlar.

Kriptografi algoritmaları kabaca iki grupta toplanabilir: Kapalı algoritmalar ve anahtar tabanlı algoritmalar. Kapalı algoritma yönteminde güvenlik, şifreleme algoritmasının gizli olmasıyla sağlanır. Sadece haberleşecek olan taraflarca bilinen bu fonksiyonun güvenliğinin sınanması, kalite kontrol ve standardizasyon işlemleri imkansız hale gelir. Ayrıca algoritmanın açığa çıkması durumunda sistem yeni bir algoritmaya göre baştan tasarlanmalıdır. Kapalı algoritmalar büyük işletmeler için uygun bir çözüm değildir.

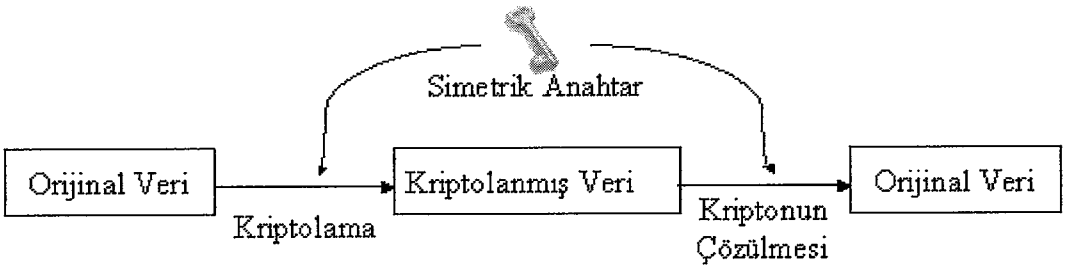
Anahtar tabanlı algoritmalarda ise algoritma açıktır fakat algoritmada kullanılan anahtar gizlidir. Fonksiyon, gizli bir anahtar ile kullanıldığında ihtiyaç duyulan güvenliği sağlar. Üstelik açığa çıkması durumunda sadece anahtarı değiştirmek sistemi tekrar güvenli hale getirir ve bu son derece pratiktir.

Anahtar tabanlı şifreleme sistemlerinde veriyi şifrelemek ve şifrelenmiş veriyi çözmek için iki ayrı anahtar kullanılmaktadır. Anahtar tabanlı şifreleme algoritmaları bu iki anahtarın seçilme şekillerine göre sınıflandırılmıştır.

3.1 Simetrik Anahtar Algoritmaları

Bu tür algoritmalarda Şekil 3.1'de gösterildiği gibi, veriyi şifrelemek için ve şifreli veriyi çözmek için kullanılan anahtarlar kolaylıkla birbirinden elde edilebilir. Hatta

kimi zaman bu iki anahtar aynıdır. Anahtarın gizli tutulmasından dolayı bu tip algoritmalara gizli anahtar algoritmaları da denilmektedir.



Şekil 3.1 Simetrik anahtar şifreleme / şifre çözme yapısı

Bu algoritmalar basit, kolay uygulanabilir, hızlı ve verimlidir. Fakat tarafların kullanılacak anahtar üzerinde anlaşmaları en zayıf noktalarıdır. Bu algoritmalar paylaşımın olmadığı durumlarda uygundur. Örneğin kişisel bilgisayarlarda dosyaların veya sabit diskin şifrelenmesi için rahatlıkla kullanılabilirler.

En genel kullanılan bazı simetrik anahtar algoritmaları şunlardır:

DES (Data encryption standard) : IBM tarafından geliştirilen DES 1977 yılında ABD Ulusal Teknoloji ve Standartlar Enstitüsü (NIST – National Institute of Standards and Technology) tarafından resmi bir standart olarak kabul edilmiştir. 64 bit blok genişliğine sahip bir blok şifreleyicidir ve 56 bit anahtar kullanmaktadır. Modern bilgisayarlar ve özel amaçlı donanımlar ile yapılan ayrıntılı anahtar taramalarına karşı yeterince güvenli değildir.

Üçlü DES (3DES), DES'in birbiriyle ilişkisiz anahtarlarla üç kere tekrarlanması temeline dayanır ve şifreleme-deşifre-şifreleme sırasıyla uygulanır. Anahtar uzunluğu DES'in 2 ya da 3 katı olduğu için DES'e göre çok daha fazla güvenlidir.

IDEA (International data encryption algorithm) : Bu yöntem DES'e kıyasla iki kat daha hızlı ve oldukça yüksek güvenliklidir. Şimdiye kadar üretilen en hızlı ve en güvenilir algoritmalardandır. 128 bit anahtar kullanmaktadır. Kaynak kodu ücretsiz olmasına rağmen ticari kullanımlar için lisans gerekmektedir.

RC5 (Rivest cypher version 5) : RSA (Rivest Shamir Adleman) grubu tarafından geliştirilmiş olan bu yöntemin anahtar uzunluğu 56, 64 veya 128 bit olabilir. Kaynak kodu ücretsiz olmasına rağmen ticari amaçlı kullanımlar için lisans gerekmektedir.

Blowfish : Bruce Schneier tarafından yazılmış olan bu algoritma yüksek güvenilirliğe ve orta düzeyde hıza sahiptir. Anahtar uzunluğu 32 ile 448 bit arasında değişmektedir ve ücretsiz olarak kullanılabilmektedir.

Tek kullanımlı şerit (One-time pad) : Kesin güvenilirliği kanıtlanmış basit bir şifreleme yöntemidir. Bu algoritmada gönderilecek veri uzunluğu kadar, rasgele seçilmiş harflerden oluşan bir anahtar kullanılmaktadır. Şifreleme işleminde, mesajdaki her harf ve o harfe şeritte karşı gelen harf sayıya çevrilip toplanır. Harfler sayıya çevrilirken alfabedeki sıra numaraları kullanılabilir. Toplam, alfabedeki harf sayısını aşarsa, alfabedeki harf sayısı toplamdan çıkarılır. Elde edilen sayı tekrar harfe dönüştürülür ve iletilir.

Şifreyi çözmek için aynı anahtar şerit ile işlemler tersinden yapılır. Bu sefer şeritteki harfler şifreli mesajdaki harflerden çıkartılır, toplam sıfırın altına düşerse alfabedeki harf sayısı eklenir. Elde edilen sayı tekrar harfe dönüştürülür ve düz metin elde edilmiş olur.

Aşağıda tek kullanımlı şerit kullanımına bir örnek verilmiştir:

Şifresiz metin : BUMESAJSIFRELENMISTIR

Şerit : DOKPEZRLSSZCHEBZMQCHJ

Şifreli metin : EIWTWZADAXQGSIOLUIVPA

3.2 Asimetrik Anahtar Algoritmaları

Simetrik anahtar algoritmalarında karşılaşılan anahtar yönetimi problemini çözme amacıyla Whitfield Diffie ve Martin Hellman, 1976 yılında asimetrik anahtarlı yapıları geliştirmiştir. Bu sistemde her kişi biri açık, diğeri gizli olmak üzere bir çift anahtar edinir. Açık anahtar herkese açıktır, gizli anahtar ise sadece sahibinin erişebileceği şekilde saklanmalıdır. Yani tarafların bildiği ortak bir gizli anahtar yoktur. Bu sistemde kaygı duyulacak tek nokta açık anahtar ve sahibinin güvenilir ve doğru bir şekilde eşleştirilmesidir.

Asimetrik anahtar algoritmalarının tümü, çok büyük sayılarla yapılan bazı işlemlerin bir yönde çok kolay, aksi yönde ise çok zor olması özelliğini kullanmaktadır. Asimetrik kriptografi terimi bu dengesizliği ifade eder.

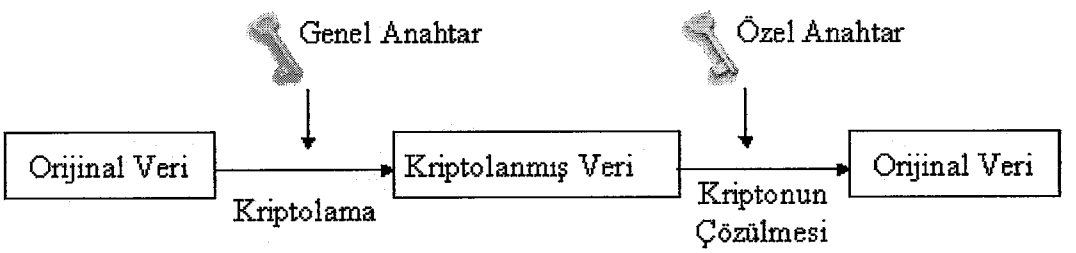
Asimetrik anahtarlı sistemlerde gizli anahtar ile açık anahtar arasında matematiksel bir bağlantı vardır. Bu sistemlere açık anahtarı kullanarak saldırmak her zaman mümkündür. Bu yüzden açık anahtardan gizli anahtarı elde etme işlemi mümkün olduğunca zorlaştırılmalıdır. Bu anahtarları oluşturmak için çözülememiş matematik problemleri kullanılmaktadır.

İki büyük asal sayı çarpımından oluşan bir sayının çarpanlarını bulma problemine "çarpanlarına ayırma problemi" denir. RSA ve Rabin Williams algoritmaları bu problem üzerine kuruludur.

$y = g^x \pmod p$ olacak şekilde p, g ve y'nin bilinmesi durumunda x'in elde edilmesi problemine "ayrık logaritma problemi" denir. Diffie – Hellman anahtar paylaşımı algoritması, ElGamal şifreleme ve imza algoritmaları ve DSA sayısal imza algoritması bu probleme dayalıdır.

"Eliptik eğri ayrık logaritma problemi", $y^2 = x^3 + ax + b \pmod p$ denkleminin çözümü üzerine kurulmuştur. Schnorr imzalama algoritmaları ve Nyberg – Ruppel imza algoritmaları bu problem üzerine geliştirilmiştir.

Asimetrik şifreleme sistemlerinde gönderici taraf alıcının açık anahtarını elde eder, Şekil 3.2'de görüldüğü gibi, mesajı bu anahtarla şifreler ve iletir. Alıcı taraf gelen mesajı kendi gizli anahtarıyla çözer ve mesajı okur. Bu gizli anahtarı bilmeyen herhangi bir kimse şifrelenmiş mesajı elde etse bile bu mesajı çözemez.



Şekil 3.2 Asimetrik anahtar şifreleme / şifre çözme yapısı

3.3 Simetrik ve Asimetrik Algoritmaların Karşılaştırılması

Asimetrik kriptografide gizli anahtarın herhangi bir şekilde taşınması gerekmediği için daha fazla güvenlik sağlanmaktadır. Simetrik algoritmalarda şifreleme ve şifre çözme işlemleri aynı anahtar ile yapıldığı için, gizli anahtarın iletişim kanalları üzerinden iletilmesi söz konusudur. Bu da gizli anahtarın istenmeyen kişiler tarafından elde edilme olasılığını doğurmaktadır.

Gizli anahtarlı sistemlerde kimlik denetiminde gizli bir bilginin paylaşılması ve bazı durumlarda üçüncü bir kişiye güven duyulması gerekliliği vardır. Açık anahtarlı sistemlerde ise herkes kendi anahtarından sorumludur. Kimse, gönderdiği sayısal imzalı mesajın gizli anahtarını bilen bir başkası tarafından oluşturulduğunu iddia edemez.

Açık anahtarlı sistemler simetrik anahtarlı sistemlere göre daha fazla işlem gücü ister. Bu yüzden kullanım alanları kısıtlıdır ya da başka algoritmalarla birlikte kullanılırlar.

Açık anahtarlı kriptografide onay kurumuna yapılan başarılı bir saldırı sonucu herhangi bir kullanıcının açık anahtarı yerine istenilen açık anahtar koyularak bu kullanıcıya gönderilen mesajlar elde edilebilir, mesajın içeriği değiştirilerek kullanıcının gerçek açık anahtarıyla şifrelenip gönderilebilir.

Bazı durumlarda açık anahtarlı yapılar gereksizdir. Örneğin gönderici ve alıcı yüz yüze görüşerek anahtar üzerinde anlaşabilir ya da bütün anahtarları bilen bir otorite bulunduğu durumlarda açık anahtarlı yapılardan vazgeçilebilir. Ancak kullanıcı sayısı arttığında problem olacaktır.

Tek kullanıcının bulunduğu bir ortamda açık anahtarlı yapılar çok anlamlı değildir. Örneğin kişisel dosyalarını şifreleyip saklamak isteyen bir kullanıcının kendi seçeceği gizli anahtar ile simetrik bir şifreleme algoritması kullanması daha uygundur.

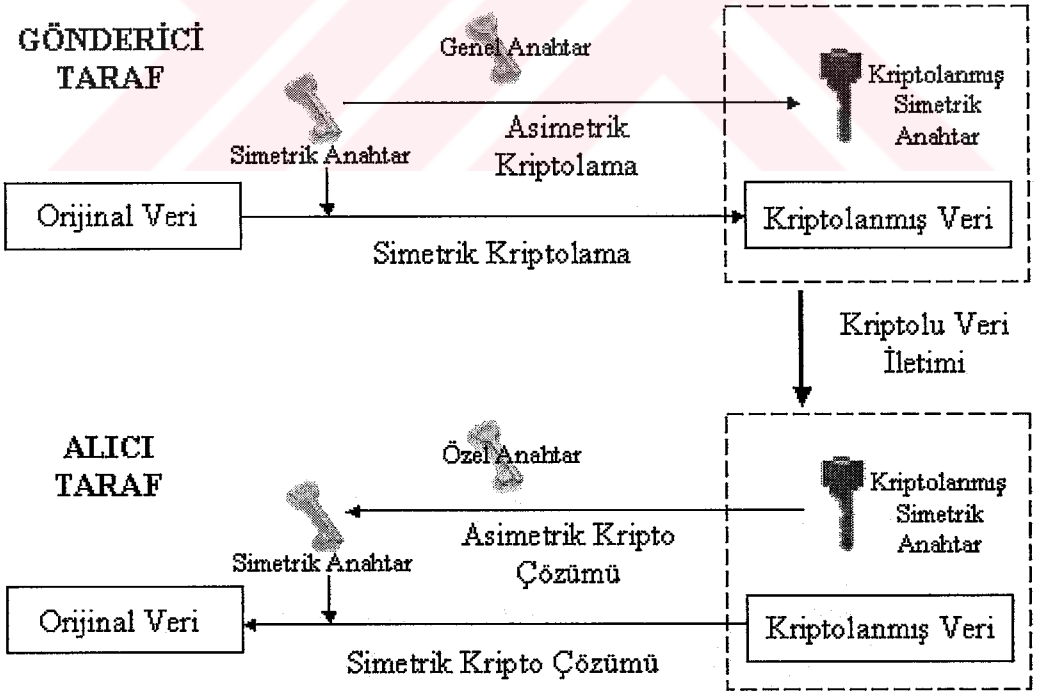
Açık anahtarlı yapılar gizli anahtarlı yapıların yerine geçmeye aday değil, daha çok onları daha güvenli hale getirecek tamamlayıcı unsurlardır. Örneğin simetrik gizli

anahtarları açık ağlar üzerinden taşımak için asimetrik anahtar algoritmaları kullanılabilmektedir.

3.4 Karışık Algoritmalar

Açık anahtar tabanlı algoritmalar iyi güvenlik seviyeleri sağlamasına rağmen oldukça fazla işlem gücü harcar. Örneğin DES, RSA algoritmasına göre yazılım açısından yaklaşık olarak 100 kat, donanım açısından ise yaklaşık olarak 1000 kat daha hızlıdır. Asimetrik algoritmaların güvenlik seviyesi ve simetrik algoritmaların hız avantajını aynı anda kullanabilmek için karışık algoritmalar kullanılmaya başlanmıştır.

Orijinal veri Şekil 3.3'te gösterildiği gibi simetrik anahtar kullanılarak şifrelenir. Kullanılan simetrik anahtar, alıcının açık anahtarı kullanılarak şifrelenir ve mesajın sonuna eklenerek iletilir. Alıcı taraf mesaja ekli olan şifreli anahtarı kendi gizli anahtarını kullanarak çözer ve metni şifrelemede kullanılan simetrik anahtarı elde eder. Daha sonra bu anahtarı kullanarak şifreli metni çözer ve mesajı elde eder.



Şekil 3.3 Karışık algoritma kullanan şifreleme / şifre çözme yapısı

PGP (Pretty Good Privacy), karışık algoritma yöntemini kullanan ve özellikle elektronik posta iletiminde yaygın olarak kullanılan bir algoritmadır.

3.5 Çırpma Fonksiyonları

Çırpma fonksiyonları, değişken uzunlukta bir mesajı almakta ve sabit uzunlukta bir çıktı üretmektedir. Belli bir mesaja aynı çırpma fonksiyonu uygulanması durumunda aynı çıktı elde edilmektedir. İyi bir çırpma fonsiyonuna uygulanan mesajda yapılan tek bitlik bir değişiklik, çıkış değeri bitlerinin yaklaşık olarak yarısını değiştirir.

Çırpma fonksiyonundan geçirilen bir mesajın geri dönüşümü yoktur. Yani çırpma fonksiyonları açık olmalarına rağmen çırpma değerinden mesajın kendisini elde etmek mümkün değildir.

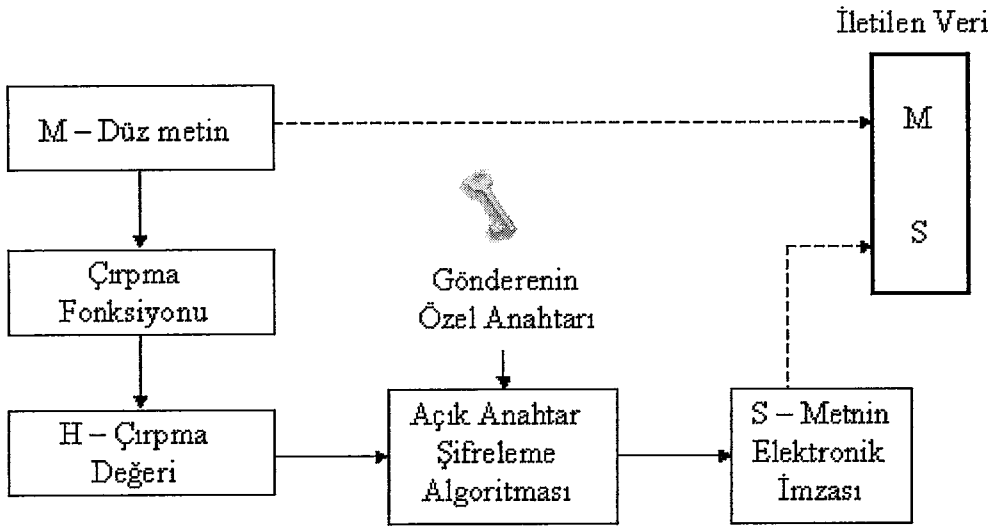
Çırpma fonksiyonları asıl olarak sayısal imzalar ile birlikte kullanılır. Bir mesajın tamamının imzalanması yerine mesajın çırpma değeri çıkartılır ve bu değer imzalanır. Alıcı taraf da bu değeri tekrar hesaplar ve imzayı kontrol eder.

Ayrıca çırpma fonksiyonları taraflardan birisinin gizli bir bilgiye sahip olduğunu ispatlaması için de kullanılabilmektedir. Örneğin gizli bir anahtar çırpma fonksiyonundan geçirilir ve bulunan değer iletilirse, bu mesajı alan taraf mesaj sahibinin ilgili anahtara gerçekten de sahip olduğunu anlar. Bu sayede ilgili gizli bilgi tehlikeye atılmamış olur.

Tek bir bitteki değişikliğin çıkışı büyük ölçüde değiştirmesi özelliği sayesinde çırpma fonksiyonları anahtar türetme ve rasgele sayı üretme işlemlerinde de kullanılmaktadır.

3.6 Sayısal İmzalar

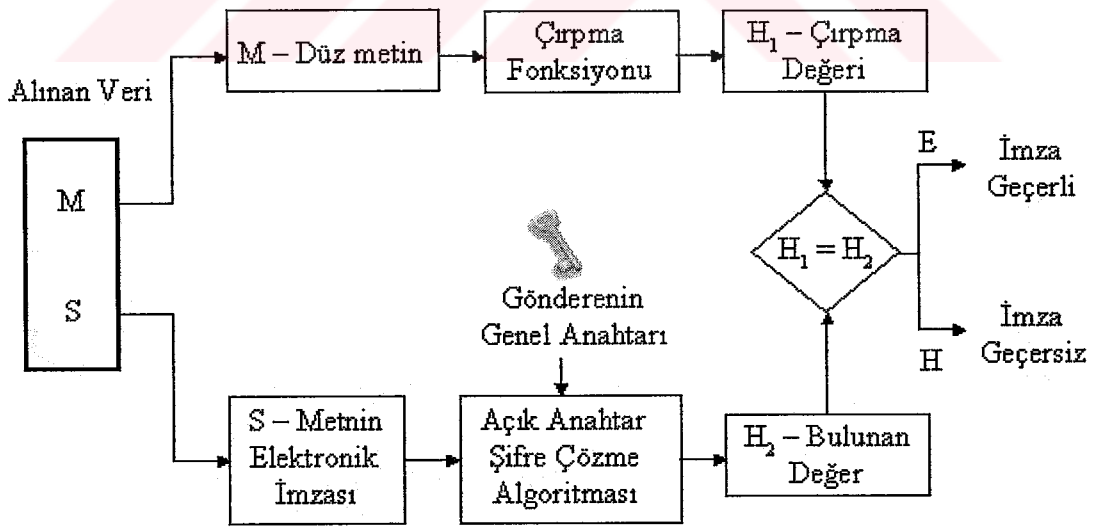
Sayısal imza, imzalanacak metin ve imzalayacak kişinin gizli anahtarı kullanılarak elde edilen bir dizi karakterden oluşmaktadır. Elle atılan imzanın elektronik ortamdaki karşılığıdır. Sayısal imza doğru şekilde kullanıldığında mesajın bütünlüğünün korunması, kaynağın doğruluğunun ispatlanması ve reddedilemez olmasını sağlamaktadır.



Şekil 3.4 Sayısal imzalama

Bir mesaj Şekil 3.4'te görüldüğü gibi, şu şekilde imzalanmaktadır:

- M mesajı çırpma fonksiyonundan geçirilerek H çırpma değeri elde edilir.
- Elde edilen H çırpma değeri gönderen tarafın gizli anahtarıyla şifrelenerek S imzası elde edilir.
- Elde edilen imza mesaja eklenerek {M, S} şeklinde iletilir.



Şekil 3.5 Sayısal imzanın kontrol edilmesi

Bir imza Şekil 3.5'te görüldüğü gibi, şu şekilde kontrol edilmektedir:

- Alınan {M, S} mesajında M ve S kısımları birbirinden ayrılır.

- M mesajı çırpma fonksiyonundan geçirilerek H_1 çırpma değeri bulunur.
- Gönderen tarafın açık anahtarıyla S imzası çözülür ve H_2 değeri elde edilir.
- H_1 ve H_2 değerleri birbirine eşitse mesaj doğrulanmış ve imza güvenilir demektir. Eğer değerler birbirlerinden farklı ise, mesaj ya iletim sırasında değiştirilmiştir ya da mesajı gönderen kişi, olduğunu iddia ettiği kişi değildir.

4. AKILLI KARTLAR

Akıllı kartlar, içine bir entegre devre gömülerek işlem yapma yeteneği kazandırılmış plastik kartlardır. Görünüş olarak, üzerlerindeki metal kontaklar dikkate alınmazsa manyetik kartlara çok benzerler fakat gerçekleştirdikleri uygulamalar açısından tamamen farklıdırlar. Özellikle üzerlerinde büyük miktarda bilgi saklayabilmeleri, veri güvenliği sağlayabilmeleri ve uzun ömürlü olmaları bakımından üstünlük sağlarlar.

Üzerinde gömülü mikroişlemci bulunan kart fikri ilk olarak 1967 yılında iki Alman mühendisi tarafından ortaya atılmıştır. Fakat ticari anlamda akıllı kartlar ilk defa 1974 yılında Fransız Roland Moreno'nun patent almasıyla ortaya çıkmıştır.

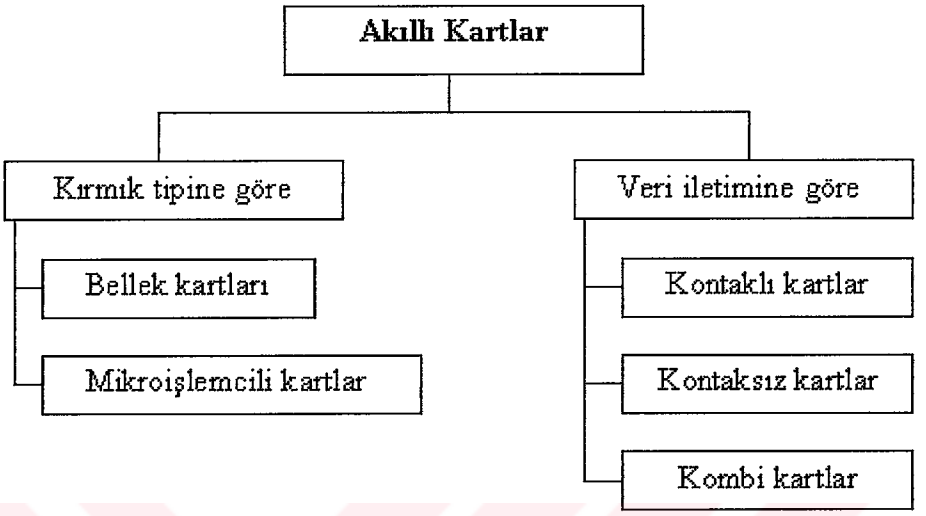
Üzerlerinde sadece bellek ünitesi bulunan akıllı kartlar ilk olarak 1984 yılında Fransız PTT'si tarafından jetonlu telefonlarda yapılan sahtekarlıkları önlemek amacıyla kullanılmıştır. Önceden tanımlanmış temel işlemleri yürütebilen bu bellek kartları zamanla, daha fazla fonksiyonelliğe sahip ve karttaki veriler üzerinde işlem yapmaya olanak tanıyan mikroişlemcili kartlar ile yer değiştirmiştir.

Bir kırmığın plastik kart yerine daha sert bir modül içinde üretilmesi çok daha kolaydır. Buna rağmen kırmığın plastik kartlar içinde üretilmesinin sebebi bu kartların sahip olduğu uluslararası standartlardır. Akıllı kartlar temel olarak ISO (International Standards Organization) – 7816 standardına uyar.

Standartlar geniş alana yayılmış ve yüzeysel olduğundan bazı özel uygulamalarda eksik kalabilmektedir. Bu noktalarda bazı organizasyonların oluşturdukları tanım özelliklerinden faydalanılmaktadır. Finansal uygulamalarda faaliyet gösteren Europay, MasterCard ve Visa firmalarının belirledikleri EMV tanım özellikleri ve haberleşme uygulamalarında belirlenen GSM tanım özellikleri en çok bilinenleridir.

4.1 Akıllı Kartların Sınıflandırılması

Akıllı kartlar Şekil 4.1'de görüldüğü gibi, temel olarak üzerlerinde kullanılan kırmık tiplerine ve veri iletim şekillerine göre sınıflandırılır.

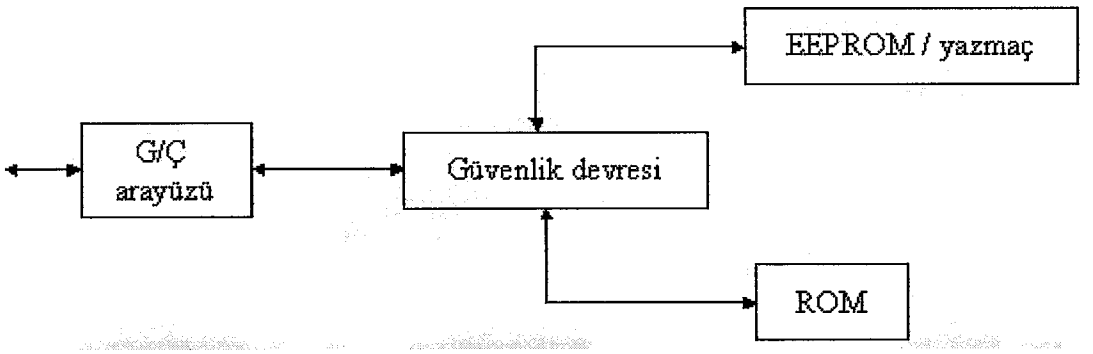


Şekil 4.1 Akıllı kartların sınıflandırılması

4.1.1 Kırmık tiplerine göre akıllı kartlar

Bellek kartları :

Terminal bellek kartına bir komut gönderir ve kart bu komutun gereğini yapar. Yani terminal ile iletişim sırasında kontrol tamamen terminaldedir. Bu nedenle bellek kartları "senkron kartlar" olarak da bilinir. Bellek kartın mimarisi Şekil 4.2'de gösterildiği gibidir.



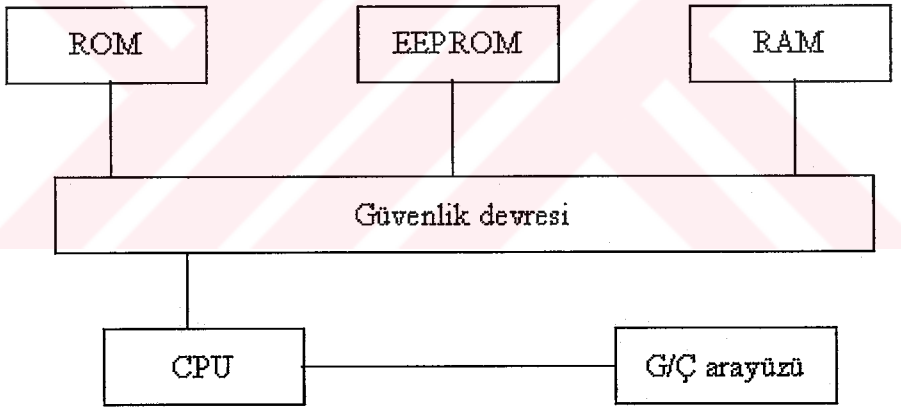
Şekil 4.2 Bellek kartları mimarisi

EEPROM bellek kartları bilgi saklama ortamlarındaki mantıkla çalışmaktadır. Terminal karta bilgi okuma, yazma, silme gibi temel komutları göndererek kart üzerindeki bilgiyi kullanır.

Yazmaçlı bellek kartlarında ise durum biraz farklıdır. Bu kartlarda Abacus tarzında, üretim esnasında doldurulan ve kullanımda sadece azalmasına izin verilen bir sayaç bulunmaktadır. Sayaç sonlanınca kartın ömrü de tamamlanmış olur. Telefonlarda kullanılanlar genellikle bu tip kartlardır.

Mikroişlemcili kartlar :

Gerçek manada akıllı olan kartlar bunlardır. Şekil 4.3'te gösterildiği gibi, mikroişlemci, RAM, ROM, G/Ç (Giriş / Çıkış) donanımı ve kalıcı bellek (EEPROM ya da EPROM) içermektedirler. Terminal tarafından girilen komut sonrası bir takım işlemler yürütür ve sonucunu terminale geri döndürürler. Bu yüzden "asenkron kartlar" olarak da bilinirler.



Şekil 4.3 Mikroişlemcili kart mimarisi

Mikroişlemcili kartın merkezinde adından da anlaşılacağı gibi, diğer 4 birim tarafından çevrelenen bir mikroişlemci vardır. Mikroişlemci temel aritmetik işlemleri gerçekleştirir, basit kesme sistemini yönetir ve tüm birimleri kontrol eder. Kesme sistemi kartın işleyişini durdurup terminalle haberleşmesini sağlamaktadır.

ROM'da kartın üretimi esnasında yüklenen ve daha sonra değiştirilemeyecek olan işletim sistemi bulunmaktadır.

EEPROM kırmığın kalıcı belleğidir. İşletim sistemi kontrolünde veri ve program kodu yazılabilmekte ve okunabilmektedir. Bazı uygulamalarda ROM işletim sisteminin temel komutlarını içerirken programlar EEPROM'da saklanmaktadır.

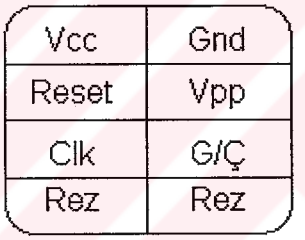
RAM bölgesi işlemcinin çalışma bölgesidir, kartın beslemesi kesildiğinde tüm içeriği silinmektedir.

G/Ç arayüzü terminal ile komut ve veri alışverişinin yapıldığı bölümdür.

4.1.2 Veri iletimine göre akıllı kartlar

Kontaklı kartlar :

Şekil 4.4'te gösterilen kontaklara sahip bu kartlar en fazla kullanılan akıllı kart çeşididir. Terminal ile etkileşim kartın yüzeyindeki kontaklar aracılığıyla yapılır. ISO-7816 standartları kontaklı akıllı kartlar içindir.



Şekil 4.4 Akıllı kart kontakları

ISO-7816 standardı 7 bölümden oluşmaktadır:

Bölüm 1'de kalınlık, boyut gibi kartın fiziksel özellikleri, çalışma ortamı ve bunların nasıl test edileceği belirtilmiştir.

Bölüm 2'de kart üzerindeki kontağın boyutu ve yeri belirtilmiştir. Bazı kontaklı akıllı kart uygulamaları kart üzerinde manyetik bir alan da içerebildiğinden, kontağın yeri belirlenirken manyetik kart standartlarındaki manyetik bölüm ile çakışmamasına dikkat edilmiştir. Bu yüzden kontak kartın tam olarak ortasında bulunmamaktadır.

Bölüm 3, terminal ile akıllı kart arasındaki iletişim protokollerini belirtir. En sık kullanılan iletişim protokolleri, T=0 asenkron yarı çift yönlü karakter iletimi ve T=1 asenkron yarı çift yönlü blok iletimidir.

Bölüm 4, endüstride kullanılan temel komut setini belirtir. Bu komutlar kart üreticilerinin karta yükleyeceği çekirdek komut setini oluşturmaktadır.

Bölüm 5, uygulama tanımlayıcıları için sayı sistemi ve kaydetme prosedürünü belirtmektedir.

Bölüm 6, kart üzerinde tanımlanan veri elemanlarını belirtir. Bunlar isim, adres, şifre, son kullanma tarihi gibi bilgilerdir.

Bölüm 7, kart iletişimindeki güvenlik prosedürü üzerinde durmaktadır ve halen geliştirme aşamasındadır.

Kontaksız kartlar :

Kontaklı kartların oldukça güvenli bir ortam sağlamalarına rağmen bazı durumlarda kısa sürede haberleşmenin tamamlanmasına ihtiyaç duyulur. Örneğin otoyollarda sürücülerin kontaklı akıllı kartlarını bir kart okuyucuya sokmaları oldukça fazla zaman kaybına yol açacaktır. Bu gibi durumlarda radyo frekansını kullanarak belirli bir mesafede kullanıcı bilgisini iletme mantığıyla çalışan kontaksız akıllı kartlar oldukça etkin bir uygulama ortamı sağlar.

Kart ve terminal arasındaki iletişim elektromanyetik sinyaller kullanılarak yapılır. Kartın çalışması için gereken güç kart içerisine yerleştirilen bir pil ile sağlanabileceği gibi terminal tarafından mikrodalga frekanslarıyla da karta iletilebilir.

Kontaksız akıllı kartlar yoğun kart ulaşımı ve veri iletimi için oldukça uygun bir ortam sağlamasına rağmen henüz bir standartları yoktur. Şu anda pazarda bir çok farklı kontaksız kart teknolojisi bulunmaktadır. Bunların her birinin kendine özgü uygulama alanı ve avantajı bulunmaktadır.

Kombi kartlar :

Kontaklı ve kontaksız akıllı kartların avantajları ve dezavantajları vardır. Kontaklı kartlar daha güvenlidir ve mevcut bir alt yapıları vardır. Kontaksız kartlar ise daha elverişli ve verimli bir işlem ortamı sunar. Bu iki kartın da avantajlarından yararlanmak için her iki özelliğe sahip kombi kartlar geliştirilmiştir.

Kombi kart terimi bazen yanlış kullanılmaktadır. İki tür kartı birleştirme iki şekilde olabilir. Birincisinde kontaklı devre ve kontaksız kırmık tamamen ayrı modüllerdir ve aralarında elektriksel bir bağlantı yoktur. Bu tür kartlar "hibrit kartlar"dır. Kombi kartlarda ise kontaklı ve kontaksız kartlar birbirlerine fiziksel olarak bağlıdır ve bunlar haberleşebilmektedir.

4.2 Akıllı Kartların Elektriksel Özellikleri

Akıllı kartların fonksiyonu tanımlanmış 6 kontak bağlantısı vardır:

- V_{cc} – Besleme gerilimi
- Gnd – Topraklama
- Clk – Saat girişi
- V_{pp} – Programlama gerilimi
- Rst – İlklendirme girişi
- G/Ç – Giriş / çıkış arayüzü

V_{cc} – Besleme gerilimi :

Terminal bu kontağı kullanarak kartın çalışması için gereken besleme gerilimini sağlar. Uygulanan gerilim, geleneksel TTL (Transistor – Transistor Logic) devrelerinde kullanılan gerilim olan 5 volttur ve en fazla %10'luk bir sapmaya izin verilmektedir.

Piyasada GSM telefonlarının ağırlığını azaltma yönünde bir eğilim vardır. Bu baskı, cep telefonlarında 3 voltluk batarya kullanılmasına neden olmuş ve bu telefonlarda kullanılan akıllı kartların 3 volt V_{cc} gerilimiyle çalışabilir olarak tasarlanmasını gerektirmiştir.

Clk – Saat girişi :

Tümleşik devrenin çalışabilmesi için devreyi tetikleyen bir saat sinyali olmalıdır. Kimi tümleşik devreler kendi saat devresini içerebilmesine rağmen genellikle harici bir saat girişi uygulanmaktadır. Akıllı kart kırmığı saat sinyalini terminalden alır. ISO standartlarında, ucuz ve kolayca uygulanabilen 3.5712 MHz ve 4.9152 MHz harici saat frekansları belirtilmiştir. Bu iki frekans da 9600 bps (bits per second) seri iletişimi destekler. İlklendirme sonrası bu frekanslardan biri seçilmelidir. Daha sonra PTS paketi ile bu frekans değiştirilebilmektedir.

V_{pp} – Programlama gerilimi :

EPROM gibi kalıcı belleklerin programlanabilmesi için akıllı karta yüksek gerilim uygulanmalıdır. V_{pp} girişi bu gerilimin verilme noktası olarak ayrılmıştır. Fakat akıllı kartlarda zamanla EEPROM belleklerin kullanılması sonucunda bu kontak kullanılmaz olmuştur.

Rst – İlklendirme girişi :

İlklendirme sinyali akıllı kartın ROM'undaki programı başlatmak amacıyla terminal tarafından uygulanır. Kart terminale yerleştirilince kontakları terminalin kontaklarıyla mekanik olarak temas eder. Daha sonra beş kontak belli bir sırayla aktif hale getirilir. Kart ilklendirmeye cevap olarak ATR bilgisini gönderir.

Kontakların aktif hale getirilme sırası şöyledir:

- Rst düşük seviyeye çekilir,
- Vcc gerilimi verilir,
- G/Ç arayüzü alıcı moda alınır,
- Vpp gerilimi bekleme moduna alınır,
- Saat girişi uygulanır,
- Rst yüksek seviyeye çekilir.

G/Ç – Giriş / çıkış arayüzü:

Terminal ve kart arasındaki haberleşme bu arayüzden yapılır. Haberleşme için bir tek hat olduğundan belli bir anda ya terminalden karta ya da karttan terminale bir iletim yapılabilmektedir. Yani yarı çift yönlü bir iletişim söz konusudur.

4.3 Akıllı Kartlarda Veri İletimi

Terminal ve kart arasında veri iletimi tek bir hattan yapılır. Bu yüzden haberleşme yarı çift yönlü olmaktadır. Akıllı kartın 8 kontağından 2 tanesi ileride kullanılmak üzere rezerve edilmiştir. Bunlardan birini ikinci bir G/Ç arayüzü olarak atamak tam çift yönlü haberleşmeye imkan verecektir.

Kart ile haberleşme daima terminal tarafından başlatılır. Kart yalnızca terminalin gönderdiği komutlara cevap verir, bir istek olmadan veri göndermez.

4.3.1 Fiziksel veri iletim katmanı

Akıllı kartların fiziksel veri iletim katmanı, uluslararası akıllı kart standardı ISO/IEC 7816-3'te belirtilmiştir. Terminal ve kart arasındaki tüm veri iletimi sayısal olarak, yani mantıksal '0' ve '1' değerleri kullanılarak yapılır. Bu işaretleri iletmede kullanılan geleneksel voltaj seviyeleri 0 ve +5 volttur. Bazı yeni kırmıklar 0 ve +3 volt ile de çalışabilmektedir.

Hangi voltaj seviyesinin hangi mantıksal değeri göstereceği kartın ATR sinyalinde gönderilen ilk bayt ile belirtilir. "Düz uzlaşma" durumunda mantıksal '1' değeri +3/+5 volt ile gösterilirken "ters uzlaşma" durumunda 0 volt mantıksal '1' değeridir.

Terminal ve kart arasındaki iletişim seri bit iletimi şeklinde olur. Düz uzlaşma durumunda başlangıç bitinden sonra gönderilen ilk bit en düşük anlamlı bittir (lsb – least significant bit). Ters uzlaşmada ise başlangıç bitinden hemen sonra en yüksek anlamlı bit (msb – most significant bit) gönderilir.

Kart ve terminal arasındaki iletişim asenkron karakter veya blok iletimi şeklindedir. Bu yüzden her karakter ya da blok için senkronizasyonu sağlamak amacıyla ayrıca senkronizasyon bitleri gönderilmelidir.

Her bir karakter 8 bitle ifade edilmektedir. Başlangıç bitini takiben sırasıyla karakter bitleri, parite biti ve sonlandırma bitleri gönderilir. T=0 asenkron seri karakter iletiminde sonlandırma amaçlı 2 bit gönderilirken T=1 asenkron seri blok iletiminde 1 bit gönderilir. Yani her karakter 11 ya da 12 bitle iletilir.

Devrenin çalışma hızı uygulanan saat frekansına bağlıdır. İletişim hızı da bu frekansı belli bir bölücü değere bölerek elde edilir. Akıllı kartlarda televizyon setlerinde kullanılan ucuz standart kristaller kullanılır. Bu kristallerin 3.5712 MHz ve 4.9152 MHz olmak üzere iki temel frekans değeri vardır. Ayrıca akıllı kartlarda kullanılan iki temel frekans bölücü değer 372 ve 512'dir. Akıllı kartların kullandığı en genel veri iletim hızı da buradan gelmektedir.

3.5712 MHz / 372 = 9600 bit/s

4.9152 MHz / 512 = 9600 bit/s

Frekans bölücü değeri düşürerek iletim hızını arttırmak tabii ki mümkündür fakat bu durumda akıllı kartın daha az bir saat darbesinde işlemlerini tamamlaması gerekir. Bu da akıllı kartın program kodunu önemli bir ölçüde arttırır. Akıllı kartlardaki hafızanın kısıtlı olmasından dolayı frekans bölücü değer çok küçük tutulamamaktadır.

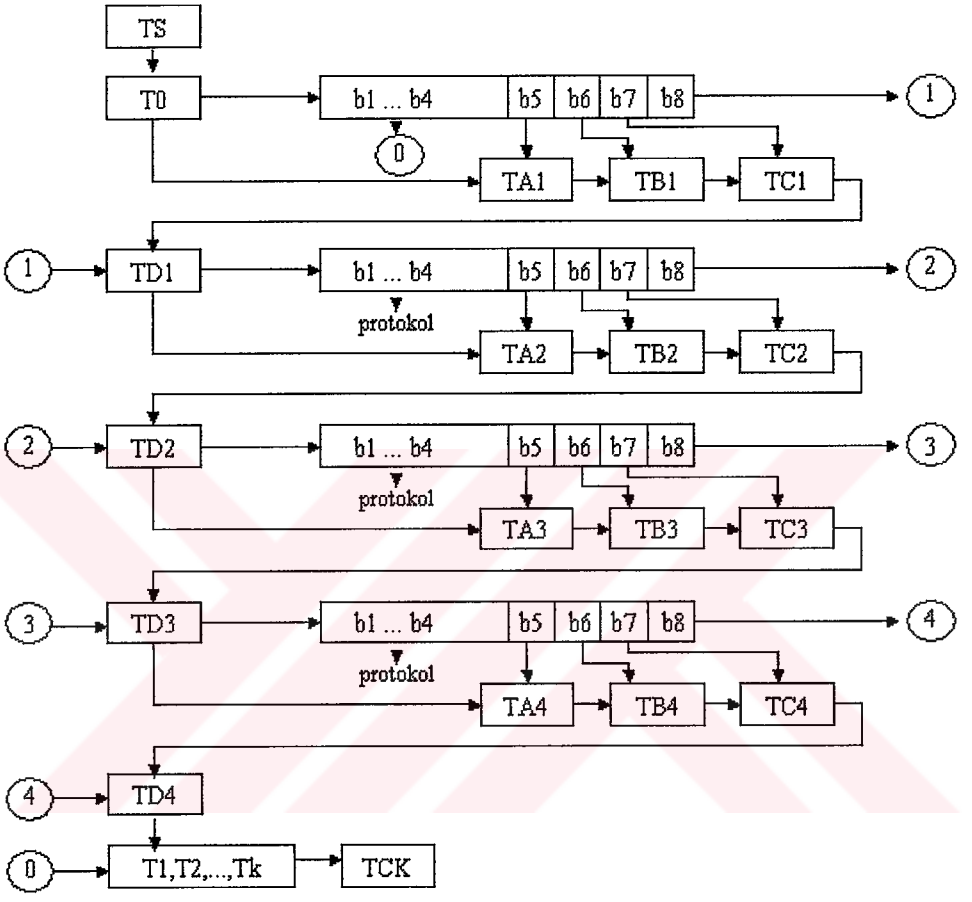
İletim hızı sabit olmadığı için bit iletim süresi kesin bir zaman ile ifade edilemez. Bunun yerine etu (elementary time unit) ifadesi kullanılır.

4.3.2 İlklendirme cevabı (ATR – Answer to reset)

Sırasıyla besleme gerilimi, saat ve ilklendirme sinyalleri verildikten sonra akıllı kart G/Ç arayüzünden ilklendirme cevabını gönderir. Kart ve haberleşme protokolü hakkında bilgi taşıyan ATR en fazla 33 bayt olabilir ve 372 frekans bölücü değeriyle iletilir.

ATR sonrası kullanılacak olan frekans bölücü değer farklı olabilir fakat ATR'de kullanılacak değer mutlaka 372 olmalıdır. Bu sayede iletim parametreleri ne olursa olsun tüm kartların ATR dizisi standart bir şekilde okunabilmektedir.

ATR, ilklendirme sinyali uygulandıktan sonra 400 ile 40000 saat vurumu arasında Şekil 4.5'te gösterildiği gibi gönderilmelidir. Yani 3.5712 MHz saat frekansında 112 µs ile 11.2 ms arasında, 4.9152 MHz saat frekansında 81.38 µs ile 8.14 ms arasında ilklendirme cevabı gönderilmelidir. Aksi halde terminal, kartta hata olduğunu varsayar ve istenilen cevap alınana dek birkaç defa ilklendirme süreci tekrarlanır.



Şekil 4.5 ATR yapısı

Ardarda gönderilen iki ATR karakteri arasında en fazla 9600 etu gecikme olabilir. 3.5712 MHz saat frekansı için standartlar iki ATR karakteri arasında en fazla 1 saniye beklemeye izin vermektedir.

İlklendirme cevap karakterleri 5 grupta incelenir:

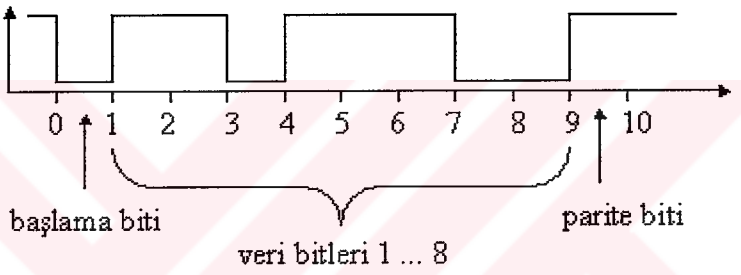
- TS — Başlangıç karakteri
- T0 — Format karakteri
- TA1,TB1,TC1,TD1,... — Arayüz karakterleri

- T1,T2, …, TK Hatırlatma karakterleri
- TCK Kontrol karakteri

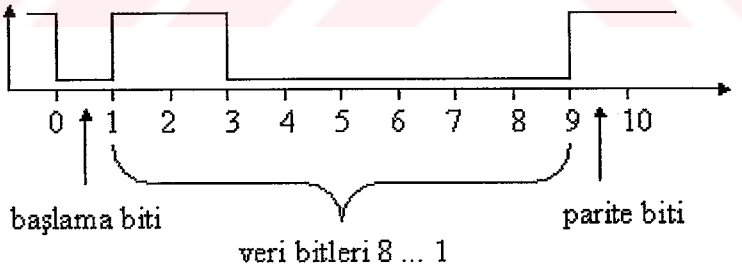
ATR karakterleri

Başlangıç karakteri:

ATR ve sonraki karakterlerin iletiminde kullanılacak olan uzlaşma durumunu belirtir. Mantıksal '1' değerinin yüksek voltaj değeriyle gösterildiği ve düşük anlamlı bitlerin önce gönderildiği düz uzlaşma durumu Şekil 4.6'da gösterildiği gibi '3Bh' değeri ile belirtilirken ters uzlaşma durumu Şekil 4.7'de gösterildiği gibi '3Fh' değeriyle belirtilmektedir.



Şekil 4.6 Düz uzlaşma durumu



Şekil 4.7 Ters uzlaşma durumu

Format karakteri:

Format karakterinin ilk 4 biti arayüz karakterlerinden sonra iletilecek olan hatırlatma karakterlerinin sayısını belirtirken son 4 biti, TA1, TB1, TC1, TD1 arayüz karakterlerinin hangilerinin iletileceğini belirtir.

Arayüz karakterleri:

Arayüz karakterleri kullanılan protokolün tüm parametrelerini belirtir. TA_i, TB_i, TC_i, TD_i, baytlarından oluşur. ATR'de bu karakterlerin gönderilmesi zorunlu değildir. Örneğin protokoldeki temel haberleşme parametreleri kullanılacaksa bu karakterlerin hiçbirisi gönderilmeyebilir.

TA_1'in ilk 4 biti FI ve son 4 biti DI değerlerini belirtir. İlgili tablolara bakılarak bu değerlerden sırasıyla frekans bölücü değer (F) ve bit hızı ayarlama faktörü (D) elde edilir. Bu değerlerden de 1 etu'nun ne kadar bir süreye eşit olduğu aşağıdaki formülle bulunur:

$$1 \text{ etu} = F / (D \times f_s) \text{ saniyedir}$$

Başlangıç için D=1 ve F=372 olduğundan,

$$1 \text{ etu} = 372 / f_s \text{ saniyedir}$$

TA_2 karakteri protokol tipi seçimi (PTS) durumunu gösterir. Kart ile terminal arasındaki haberleşmenin uzlaşılabilir mod ve özel mod arasında anahtarlanabilir olup olmamasını belirtir.

TB_1'in 7. ve 6. bitleri I1, son 5 biti de PI1 değerlerini gösterir. İlgili tablolara bakılarak bu değerlerden EPROM programlama gerilimi ve akımı elde edilir. TB_2 karakteri bu değerler hakkında daha ayrıntılı bilgi vermektedir. Günümüzde akıllı kartlar EPROM yerine EEPROM kullandıkları için bu karakterler genellikle gönderilmemektedir.

TC_1 karakteri, iki ardışıl karakter arasında fazladan bekleme süresi olan N'yi belirlemek için gönderilir. N=FFh ise fazladan bekleme süresi mümkün olan en az bekleme süresi olacaktır. Bu değer T=0 için 2 etu, T=1 için 1 etu'dur. N'nin diğer değerleri için iki karakter arasında N etu kadar fazladan bekleme zamanı bulunacaktır.

TC_2 karakteri iki ardışıl karakter arasında en fazla yapılabilecek bekleme süresinin hesaplanmasında kullanılan WI değerini taşır ve aşağıdaki şekilde hesaplanır.

$$\text{maksimum bekleme süresi} = 960 \times D \times WI \text{ saniye}$$

TC_2 karakteri gönderilmezse WI=10 olarak alınır.

TD_i arayüz karakterinin son 4 biti, T0 format karakterinde olduğu gibi, TA_{i+1}, TB_{i+1}, TC_{i+1} ve TD_{i+1} karakterlerinin hangilerinin iletileceğini gösterirken, ilk 4 biti kartın desteklediği iletişim protokollerinde birisini belirtir. Akıllı kartların kullandıkları iletişim protokolleri şunlardır:

- T = 0 Asenkron yarı çift yönlü bayt aktarımı
- T = 1 Asenkron yarı çift yönlü blok aktarımı
- T = 2/3 Tam çift yönlü işlemler için ayrılmıştır
- T = 4 Etkili yarı çift yönlü bayt aktarımı için ayrılmıştır
- T = 5 ... 13 Gelecek kullanımlar için ayrılmıştır
- T = 14 ISO'ya uymayan protokolleri belirtir
- T = 15 Gelecek kullanımlar için ayrılmıştır

Hatırlatma karakterleri:

İşletim sistemine bağlı olarak, kartın yaşam süresi, üzerinde çalışan programın versiyonu gibi çeşitli bilgiler hatırlatma karakterleriyle sunulabilir. Yorumlaması kolay olması için genellikle ASCII karakteri olarak kodlanırlar. Uzunlukları 4 bitle ifade edildiği için en fazla 15 karakter olabilirler. Hızın önemli olduğu uygulamalarda çok kısa gönderilir ya da hiç gönderilmezler.

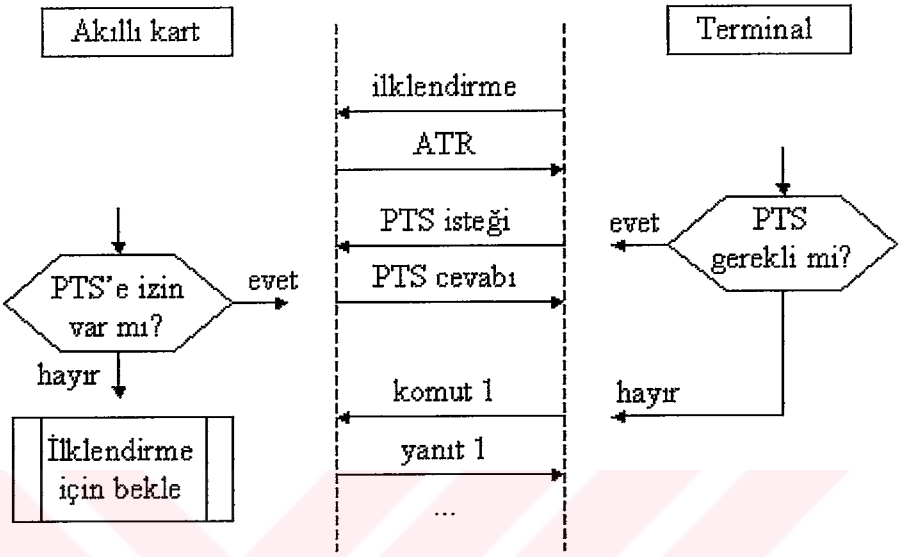
4.3.3 Protokol tipi seçimi (PTS – Protocol type selection)

Akıllı kartlar ilklendirme cevabında arayüz karakterlerini kullanarak çeşitli veri iletişim parametrelerini gönderir. Terminal bu parametrelerden birini ya da bir kaçını değiştirmek isterse haberleşme başlamadan önce bir protokol tipi seçim paketi göndermelidir. Terminal bu paket sayesinde akıllı kartın izin verdiği ölçüde haberleşme parametreleri üzerinde değişiklik yapabilecektir.

Akıllı kartlar parametre seçimine izin verme açısından iki farklı modda çalışmaktadır. "Uzlaşılabilir mod"da terminal ATR'nin hemen sonrasında PTS paketi

göndererek protokol seçiminde bulunabilir. "Değişmez mod"da ise kart protokol tipi seçimine izin vermez. Değişmez modda çalışan kartlar, terminalin göndereceği PTS paketlerini dikkate almaz.

PTS paketinin gönderilmesi durumu Şekil 4.8'de gösterildiği gibidir.



Şekil 4.8 Protokol tipi seçimi

PTS isteği oturum boyunca en fazla bir defa ve mutlaka ATR alınır alınmaz gönderilmelidir. Kart istenilen protokol modifikasyonlarına izin verirse, aldığı PTS paketini aynı şekilde terminale gönderecektir. Aksi takdirde kart hiçbir yanıt vermeyecektir ve terminal kartı bu durumdan çıkarmak için ilklendirme sürecini işletecektir.

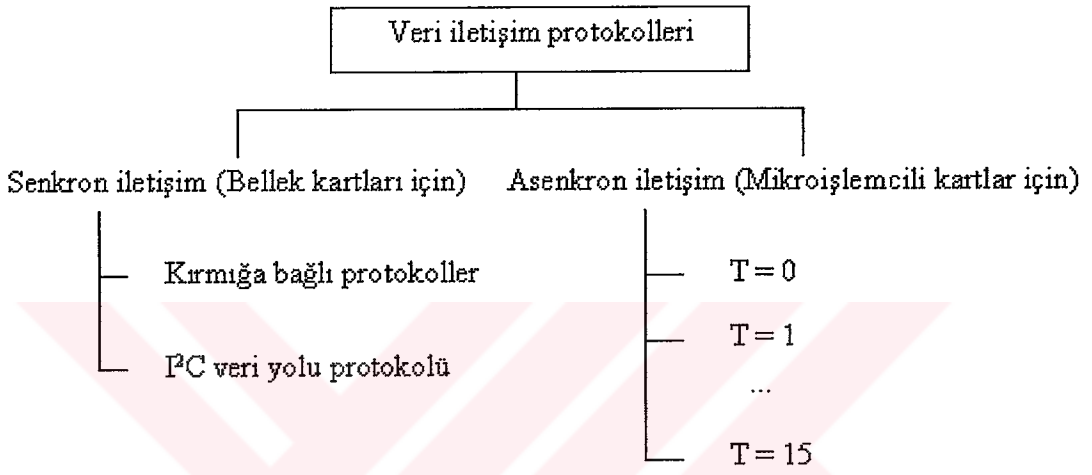
PTS dizisinin ilk karakteri PTSS başlangıç karakteridir ve değeri 'FFh'tir.

Daha sonra, istenilen iletişim protokolünü belirten PTS0 karakteri gönderilir. Bu karakter ayrıca PTS1, PTS2 ve PTS3 karakterlerinden hangilerinin gönderileceğini belirtir.

PTS1 karakteri, frekans bölücü değer ve bit hızı ayarlama faktörlerinin elde edildiği FI ve DI değerlerini içerir. PTS2 karakteri, bir karakter iletimi sonrasındaki bekleme süresini gösteren N değerini içerir. PTS3 karakteri daha sonra kullanılmak üzere ayrılmıştır.

4.3.4 Veri iletişim protokolleri

Mikroişlemcili kartlar ve bellek kartları veri iletişimi bakımından farklılık gösterir. Bellek kartları, tamamen terminal kontrolünde oldukça basit senkron haberleşme protokolleriyle haberleşirken, mikroişlemcili kartlar nispeten daha karmaşık olan ve kontrolün hem terminal hem de akıllı kartta olduğu asenkron veri iletişim protokollerini kullanır. Akıllı kart veri iletişim protokolleri Şekil 4.9'da gösterilmiştir.



Şekil 4.9 Veri iletişim protokolleri

Senkron veri iletişimi

Bellek kartlarının kullandığı, donanımla gerçekleştirilen oldukça basit ve hızlı bir iletişim biçimidir. Terminaldeki uygulama kırmığın belleğine doğrudan ulaşabilmelidir. Protokoller, kırmıkta bulunan verinin fiziksel olarak adreslenmesine, okunmasına ve yazılmasına imkan sağlamaktadır.

Protokollerin hata kontrolü olmadığından hata algılama uygulama seviyesinde yapılır ve hatalı olduğu farkedilen işlem tekrarlanır. Başlangıç biti, parite biti, sonlandırma biti gibi yardımcı bilgi iletimi gerektirmediği için iletilen bitler tamamen veri bitleridir.

Okuma, yazma, adres göstericisini arttırma gibi temel fonksiyonlar, veri, saat ve kontrol bağlantılarına çeşitli sinyal girişleri uygulanarak gerçekleştirilir.

Senkron iletişim protokolleri bir biçimli değildir ve kimi zaman kırmığa bağlı özellikler gösterir. Bu da senkron bellek kartlarını kullanan terminallerin olası her türlü marka ve modelin senkron protokol özelliklerini içermesini gerektirmektedir. Bu sorunu gidermek için senkron haberleşmede standartlaşma yoluna gidilmiş ve I^2C veri yolu haberleşme protokolü senkron kartlara uygulanmıştır.

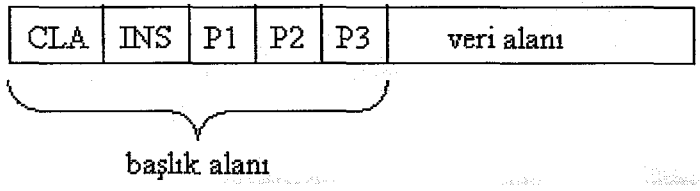
I^2C veri yolu protokolü terminal ve kart arasında SCL (Serial Clock) ve SDA (Serial Data) bağlantıları gerektirir. Terminal SCL bağlantısından iletişim için gereken saat sinyalini sağlar ve SDA bağlantısına da Vcc gerilimini uygular. Veri iletimi yapmak isteyen taraf SDA hattını 0 volta çeker, 1 saat periyodu kadar bekler, verisini iletir ve hattın tekrar Vcc gerilimine gelmesine izin verir.

Akıllı kart I^2C veri yolu iletişiminde en anlamlı bit önce olmak üzere her seferinde 1 bayt iletilmektedir. İletilen her bayt sonrası I^2C veri yolu protokolü gereği diğer taraf bir bilgilendirme bitiyle karşılık verir. Senkron iletişimde temel olarak bir adresten bilgi okuma ve bir adrese bilgi yazma işlemleri vardır. Her bir işlem I^2C veri yolu üzerinden birkaç bayt iletilerek yapılır.

T=0 veri iletişim protokolü

Standartlaşmış ilk akıllı kart veri iletişim protokolüdür. Akıllı kart teknolojisinin başlangıç yıllarında oluşturulmuş olduğundan en az bellek kullanımı ve basit gerçekleştirme mantığıyla geliştirilmiştir. Günümüzde hala en sık kullanılan asenkron veri iletişim protokolü çeşididir.

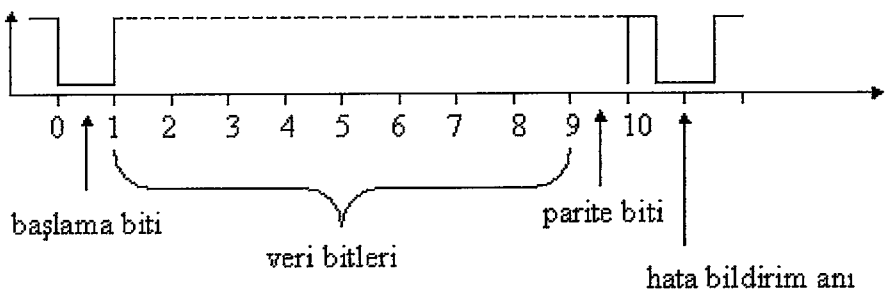
T=0 protokolünde iletilen en küçük birim bir bayttır. İletilen paket, Şekil 4.10'da görüldüğü gibi başlık ve veri bölümlerine ayrılır. Başlık kısmı komutun sınıfı, komutun kendisi, P1 ve P2 komut parametreleri ve iletilecek ya da alınacak veri bölümü uzunluk bilgisinden oluşur.



Şekil 4.10 T=0 protokolü komut yapısı

Bayt yönelimli bir protokol olduğu için, hata algılanması durumunda yalnızca hatalı olan baytın tekrar gönderilmesi yeterli olur. Hatalı bayt parite kontrolü yapılarak algılanabilir. Dolayısıyla bir bayt içinde çift sayıda bitin hatalı gönderilmesi durumunda hata algılanamaz.

Hatalı parite algılayan taraf, Şekil 4.11'de gösterildiği gibi bayt iletim bekleme anında 1 etu kadar bir süre G/Ç hattını düşük voltaja çeker. Bunun üzerine gönderici taraf yalnızca hatalı olan baytı tekrarlar.



Şekil 4.11 T=0 protokolünde hata bildirim prosedürü

T=0 protokolü akıllı kartın, terminal tarafından sağlanan EPROM programlama gerilimini açıp kapatmasını yönetmesine imkan verir. Kart, terminalden gelen komuta 1 ekleyerek elde ettiği değeri terminale göndererek terminalin V_{pp} voltajını anahtarlamasını ister. Fakat günümüz kartları EEPROM kullandığından ve gerekli voltajı besleme geriliminden sağlayabildiğinden bu özelliğe ihtiyaç duymaz.

T=0 protokolü kullanan kartların uygulama katmanı haberleşmesi şu şekilde olmaktadır:

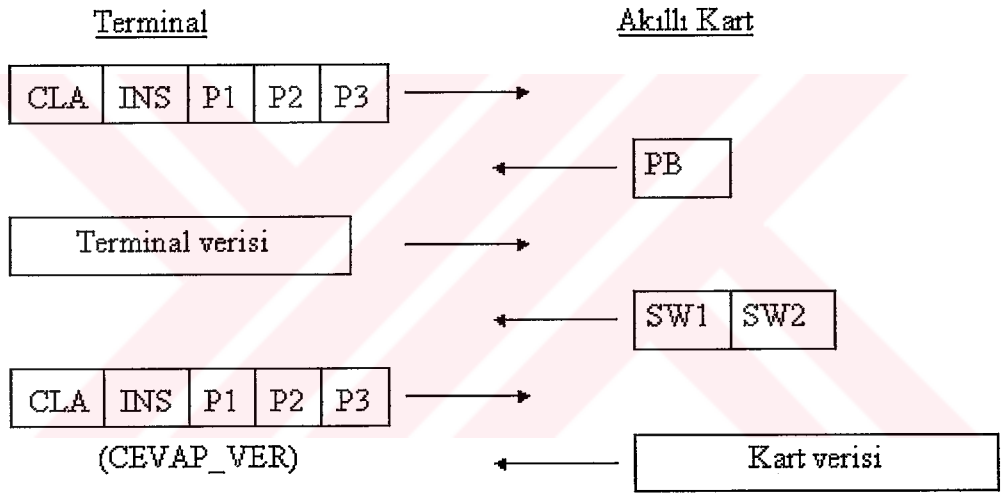
- Teminal 5 baytlık komut başlığını akıllı karta gönderir. Eğer komutun bir veri kısmı varsa terminal P3 baytını veri kısmının uzunluğu olacak şekilde iletir.
- Kart bu bilgiyi aldığını terminale bir prosedür baytıyla (PB – Procedure Byte) bildirir. Normal şartlarda kart INS (Instruction) baytını PB baytı olarak terminale gönderir.
- Eğer varsa terminal P3 bayt uzunluktaki veri bölümünü karta gönderir.

- Kart komuta karşılık olarak veri iletimi yapacaksa SW2 (Session Word) baytını bu bilginin boyu olarak ayarlar ve "SW1, SW2" cevabını gönderir. Komut karşılığında bilgi gönderilmeyecekse bu komuta ait iletişim burada sonlanır.

- Komuta cevap verilecekse terminal, CEVAP_VER (GET_RESPONSE) komutunu P3 baytı SW2 olacak şekilde karta gönderir.

- Karşılık olarak kart ilgili veriyi iletir ve bu komuta ait iletişim burada sonlanır.

Şekil 4.12'de, hem terminalin hem de akıllı kartın iletilecek bilgi baytları olması durumunda gerçekleşecek komut alışverişi gösterilmektedir.



Şekil 4.12 T=0 protokolü komut alışverişi

T=0 protokolü kartın, terminalin gönderdiği veriyi birer birer almasına imkan sağlar. Kart PB olarak INS baytının ters çevrilmiş halini gönderirse terminal ilk veri baytını gönderir. Kart bu PB'yi tekrar göndererek bir sonraki baytı ister. Tüm veri birer birer alınabileceği gibi herhangi bir anda INS baytının kendisi PB olarak gönderildiğinde terminal kalan veri baytlarını bir seferde iletir.

T=1 veri iletişim protokolü

Asenkron, yarı çift yönlü ve blok temelli bir protokoldür. Yani iletilen en küçük veri birimi bir bloktur. Bu protokol, haberleşme katmanları mimarisi özelliklerini gösterir

ve OSI referans modelinin veri bağlantı katmanına karşılık gelir. Blok transferinin iki temel amacı vardır. Bunlar, uygulama verisinin saydam olması ve iletim hatalarının düzeltilmesidir.

Terminal karta ilk bloğu gönderir, kart gerekli işlemleri yapar ve terminale cevap bloğunu iletir. Haberleşme bu şekilde devam eder. T=1 haberleşmesi terminal ve akıllı kart arasında kullanıldığı gibi terminaller arasında da kullanılmaktadır. Şekil 4.13'te T=1 protokolünün blok yapısı gösterilmektedir.

Başlangıç alanı			Bilgi alanı	Sonlandırma alanı
NAD	**PCB**	**LEN**	**APDU**	**EDC**
1 bayt	1 bayt	1 bayt	0 – 254 bayt	1 - 2 bayt

Şekil 4.13 T=1 protokolü blok yapısı

T=1 haberleşme protokolünde bir blok başlangıç, veri ve sonlandırma kısımlarından oluşur. Başlangıç bölümünün ilk baytındaki üçer bit, blokların kaynak ve hedef adreslerini içerir. Kalan iki bit ise akıllı kartın V_{pp} EPROM programlama gerilimini kontrol etmesi için kullanılmaktadır.

T=1 protokolünde üç tip blok vardır. Bunlar bilgi blokları, paket alındı bilgisi blokları ve sistem bloklarıdır. Başlangıç bölümünün ikinci baytı blok tipini belirtmek için kullanılmaktadır.

Bilgi blokları (I bloks) uygulama katman verisini iletmek için, paket alındı bilgisi blokları (R blocks) gönderilen komuta olumlu ya da olumsuz cevap vermek için, sistem blokları da (S blocks) protokolle ilgili kontrol bilgisini iletmek için kullanılır.

Başlangıç bloğunun üçüncü baytı, veri bloğunda iletilecek olan bilginin uzunluğunu gösterir. Bu değer 0 ile 254 arasında olabilir. Veri bloğunda iletilen uygulama katmanına ait bilgi tamamen saydamdır, yani üzerinde hiçbir işlem yapılmadan iletilmektedir.

Protokolün sonlandırma kısmında ise LRC (Longitudinal Redundancy Check) ya da CRC (Cyclic Redundancy Check) hata kontrol baytları iletilmektedir.

Bu protokolde her bilgi bloğu bir bitlik paket gönderme sıra numarası içerir. Sırasıyla 0 ve 1 olan bu değer sayesinde paket alışverişi düzenlenmiş ve hatalı gönderilen blokların tekrar edilmesi sağlanmış olur.

Veri iletişiminin daha sağlıklı yapılabilmesi için T=1 protokolüne çeşitli bekleme süreleri eklenmiştir.

- Karakter bekleme süresi (CWT – Character waiting time) : Bir blok içinde ard arda gönderilen iki karakterin başlangıç zamanları arasında olabilecek en fazla bekleme süresidir. Bu süre geçince ya iletilecek veri bitmiş ya da iletimde bir problem olmuş demektir.

- Blok bekleme süresi (BWT – Block waiting time) : Karta gönderilen bloğun son baytının başlama zamanı ile kartın cevap olarak gönderdiği ilk baytın başlama zamanı arasında olabilecek en fazla bekleme süresidir. Bu süre zarfında kart cevap vermezse haberleşme sonlandırılır.

  Eğer kart işlemlerini tamamlamak için BWT'den daha fazla bir süreye ihtiyaç duyarsa S bloklarını kullanarak fazladan süre talep edebilir. Terminal bu süreyi vermek zorundadır.

- Blok koruma süresi (BGT – Block guard time) : Karta gönderilen bloğun son baytının başlama zamanı ile kartın cevap olarak gönderdiği ilk baytın başlama zamanı arasında olabilecek en az bekleme süresidir. Bu şekilde, bloğu gönderen terminalin alıcı konuma geçmesi için yeterli süre verilmiş olur.

T=1 protokolünün önemli özelliklerinden birisi de blok zincirlemedir. Taraflardan birisi alımı ya da gönderimi bir defada yapılamayacak kadar uzun olan bir bilgi bloğunu zincirleyebilir. Yalnızca bilgi blokları (I blocks) için izin verilen bu yöntemde uygulama katmanı verisi bölümlere ayrılır ve her bölüm birbirinden bağımsız olarak iletilir.

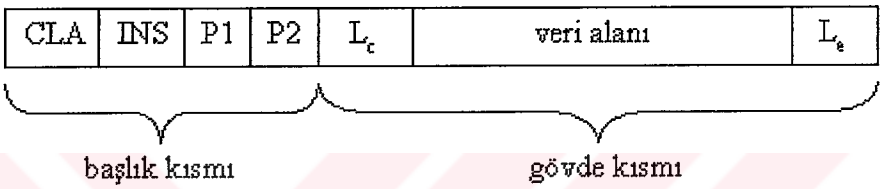
4.3.5 Mesaj yapıları (APDU)

Terminal ve akıllı kart arasındaki tüm veri alışverişi uygulama katmanı APDU (Application Protocol Data Unit) paketleri kullanılarak yapılır.

ISO/IEC 7816-4 standardıyla belirlenmiş olan APDU yapısı iletim protokolünden bağımsızdır yani farklı iletim protokolleri kullanılması durumunda APDU yapısı değişmeyecektir.

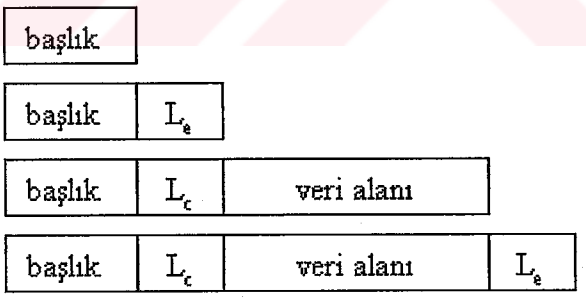
APDU komut yapısı

APDU komutu Şekil 4.14'te gösterildiği gibi, başlık ve gövde kısımlarından oluşur. 4 baytlık başlık kısmında uygulamanın sınıfı (CLA – Class), komut kodu (INS – Instruction), P1 ve P2 komut parametreleri bulunur. Değişken uzunluklu gövde kısmında ise komutun veri bölümünün uzunluk bilgisi (L_c), belirlenen uzunluktaki veri ve akıllı kartın cevabında beklenen verinin uzunluğu (L_e) bilgileri bulunur.



Şekil 4.14 APDU komut yapısı

Komutun başlık kısmının gönderilmesi zorunludur fakat gövde kısmı her zaman mevcut değildir. Olası APDU komut yapıları Şekil 4.15'te gösterilmiştir.



Şekil 4.15 Olası 4 APDU terminal komut yapısı

APDU cevap yapısı:

Şekil 4.16'da gösterildiği gibi, APDU cevabı komutta gönderilen L_e uzunluğunda bir gövde ve 2 baytlık durum (SW1 , SW2) kısmından oluşmaktadır. Gövde kısmı her zaman olmasa da komutun işlenme biçimini belirten durum kısmının gönderilmesi zorunludur.

veri alanı	SW1 SW2

Şekil 4.16 Akıllı kart APDU cevabı yapısı

Standartlaşmış durum bildirim baytları bulunmasına rağmen uygulamalar genellikle kendi belirledikleri özel kodları kullanır. Bununla birlikte, hemen hemen tüm uygulamalar evrenselleşmiş başarılı komut işletimi belirteci olan "9000h" değerini içermektedir.

4.4 Akıllı Kart İşletim Sistemleri

İşletim sistemi, bilgisayar sistemini oluşturan donanım ve yazılım nitelikli kaynakları, programlar arasında kolay, hızlı ve nitelikli bir işletim hizmetine olanak verecek biçimde paylaştırırken bu kaynakların kullanım verimliliğini en üst düzeyde tutmayı amaçlayan yazılım sistemidir.

Akıllı kart işletim sistemlerinin ana görevleri şunlardır:

- Akıllı karttan terminale ve terminalden akıllı karta veri iletimini sağlamak.
- Komutların çalışmasını kontrol etmek.
- Dosyaları yönetmek.
- Kriptografik algoritmaları yönetmek ve çalıştırmak.

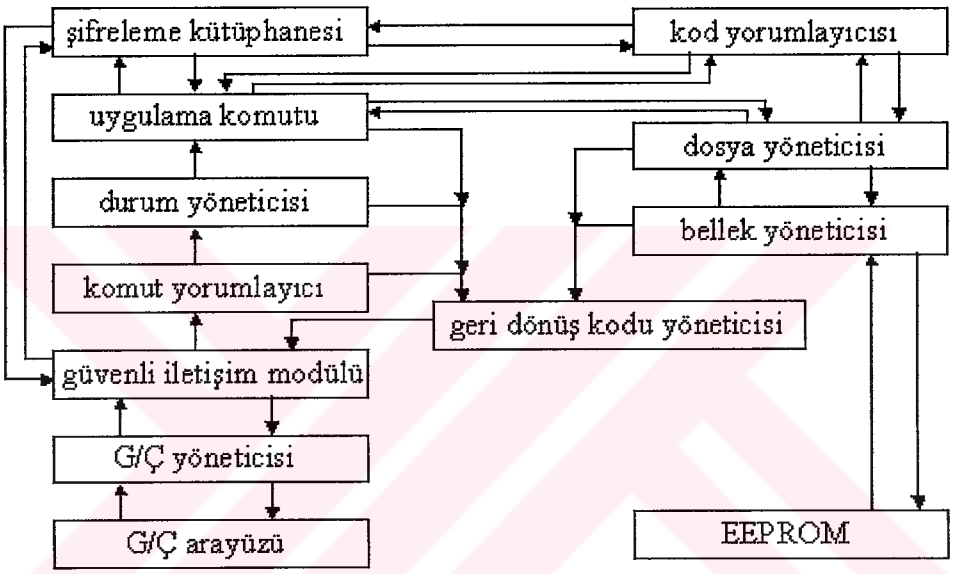
Günümüzde akıllı kartlar için geliştirilmiş genel ya da özel amaçlı bir çok işletim sistemi vardır. En bilinen işletim sistemleri şunlardır:

İşletim sistemi	Üreticisi
CardOS	Infineon
Cyberflex, Multiflex, Payflex	Schlumberger
MFC	IBM
Micado	Orga
MULTOS	Maosco

OSCAR	Oki
PCOS, MCOS, MPCOS	Gemplus
STARCOS	Giesecke & Devrient
TB, CC, Odyssey	Bull CP8
TCOS	Telesec

4.4.1 Komut işleyişi

Akıllı kartlarda komut işleyişi Şekil 4.17'de gösterildiği şekilde yapılmaktadır.



Şekil 4.17 Akıllı kart komut işleme mekanizması

Akıllı kart tüm komutları G/Ç arayüzünden alır. G/Ç yöneticisi hata algılama ve düzeltme işlemlerini gerçekleştirir. Hatasız bir komut ulaştığında güvenli mesajlaşma yöneticisi mesajın bütünlüğünü kontrol eder ve eğer mesaj şifreliyse şifreleme kütüphanesini kullanarak mesajı deşifre eder. Eğer güvenli bir iletişim yoksa bu birimde işlem yapılmaz.

Daha sonra komut yorumlayıcısı komutu çözmeye çalışır. Eğer bu mümkün olmazsa geri dönüş kodu yöneticisi çağrılır ve uygun bir hata kodu ile G/Ç arayüzünden terminale cevap gönderilir. Eğer komut doğru şekilde yorumlanırsa durum yöneticisi, komutun akıllı kartın durumuyla uyumlu bir komut olup olmadığını kontrol eder.

Uyumlu bir komut değilse uygun bir hata koduyla geri dönülür. Durumla uyumlu komutlar yorumlanarak işletilir.

Eğer komut bir dosyaya ulaşmayı gerektiriyorsa dosya yöneticisi çağrılır. Dosya yöneticisi, EEPROM'un yönetiminden sorumlu olan düşük seviyeli bellek yöneticisi vasıtasıyla dosyalara ulaşır. Fiziksel adreslere yalnızca bellek yöneticisi ulaşabilir. Bu da işletim sisteminin güvenlik ve portatifliğini arttırır.

Komut işletimi sonucuna uygun olarak geri dönüş yöneticisi uygun bir cevap kodu hazırlar.

4.4.2 Bellek organizasyonu

Akıllı kartta üç tür bellek bulunmaktadır: RAM, ROM ve EEPROM.

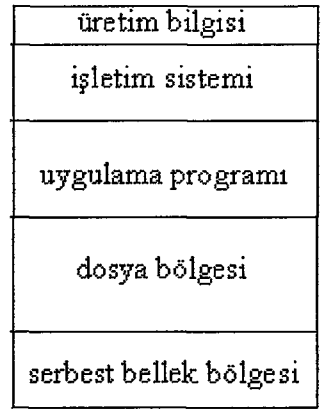
ROM bellek akıllı kartın üretimi esnasında bir defaya mahsus olarak doldurulur ve bir daha değişmez. ROM, temel işletim sistemi dosyaları ve uygulamaların kullandığı temel kütüphane dosyalarını içerir.

RAM bellek sınırsız defa yazılabilen ve silinebilen geçici bellek bölgesidir. Üzerindeki verinin kaybolmaması için besleme gerilimi sürekli olarak sağlanmalıdır.

RAM bellekte Şekil 4.18'de belirtildiği gibi, yazmaçlar, yığın bölgesi, genel değişkenler, kriptografik fonksiyon alanı, G/Ç tamponları bulunur. Mevcut RAM belleğin yetersiz olması durumunda EEPROM bellekten faydalanılabilir.

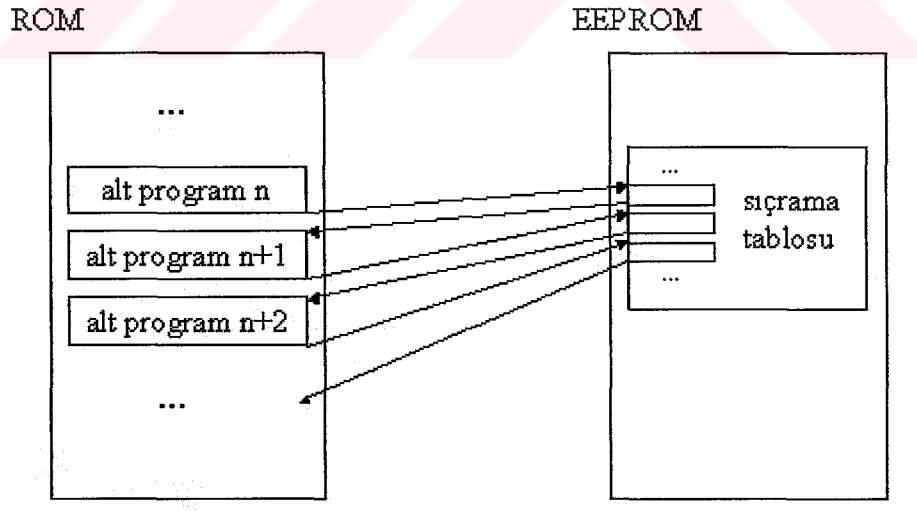
Şekil 4.18 RAM bellek yapısı

EEPROM bellek besleme gerilimi kesilse bile üzerindeki veriyi saklayabilmektedir. Bununla birlikte yazılıp silinme işleminin sınırlı sayıda yapılabilmesi ve çok yavaş olması (yaklaşık olarak RAM belleğin 1/10000'i kadar) EEPROM'un dezavantajlarıdır. EEPROM bellek yapısı Şekil 4.19'da gösterilmiştir.



Şekil 4.19 EEPROM bellek yapısı

Kart üretilirken EEPROM'un başlangıcına temel üretim bilgilerinin hemen sonrasında, genellikle 16-32 bayt uzunluğunda olan tablolar ve işletim sistemi göstericileri yüklenir. Bu tablo ve göstericiler, ROM'daki programlar ile birlikte Şekil 4.20'de gösterildiği gibi, akıllı kartın işletim sistemini yükler.



Şekil 4.20 Akıllı karta işletim sisteminin yüklenmesi

EEPROM'da işletim sistemi bölümünden sonra uygulama programı bölümü bulunur. Bu bölümün yazılabilir olması sayesinde akıllı kartın üretimi sonrasında çeşitli uygulamalar EEPROM belleğe yüklenebilmektedir.

Daha sonra dosya bölümü vardır. Akıllı kartın kalıcı bilgileri çeşitli formatlarda dosyalar halinde bu bölümde bulunmaktadır.

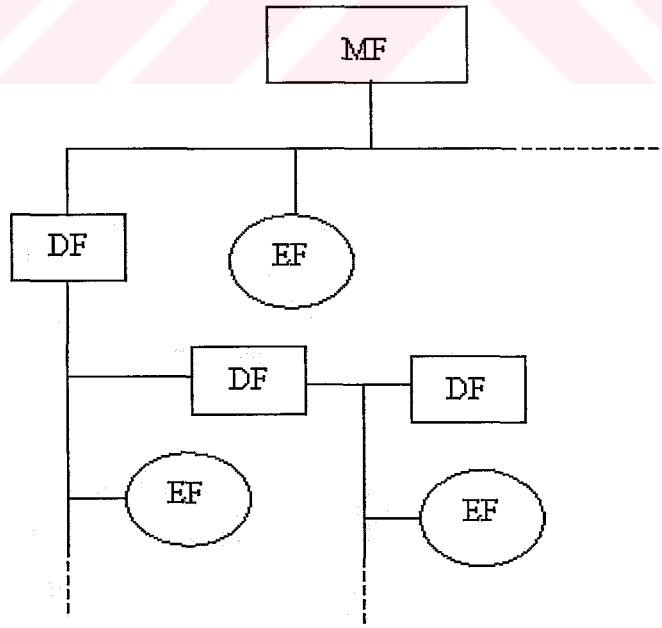
4.4.3 Dosya yapıları

Akıllı kart dosya sisteminin nesne yönelimli bir yapısı vardır. Yani dosya ile ilgili tüm bilgiler yine dosyanın içinde tutulur. Bu yüzden bir dosya ile ilgili işlem yapılmadan önce dosyayı seçmek gerekmektedir.

Akıllı kart dosya sisteminde bir dosya başlık ve gövde kısımlarından oluşur. Başlık kısmında dosya yapısı ve erişim koşulları vardır. Gövde kısmında ise dosyanın sakladığı veri bulunmaktadır.

Başlık ve gövde kısımları farklı bellek bölgelerinde saklanır. Böylece dosya başlık kısmı doğrudan okuma ve yazmaya karşı korunabilmektedir.

Bilgisayar işletim sistemlerindeki dosyalar uygulamaya yönelik yapılara sahiptirler. Örneğin Microsoft Word dosyasının kendine ait bir biçimi vardır. Akıllı kart dosya sistemlerinde ise Şekil 4.21'de görüldüğü gibi, yalnızca standart dosya yapıları bulunmaktadır.



Şekil 4.21 Akıllı kart dosya yapıları

Akıllı kart dosya sisteminin kök klasörü "ana dosya" (MF - Master File) olarak adlandırılır. Sistemde bir tane MF vardır ve ilklendirme sonrası otomatik olarak seçilir. MF tüm dosya sistemini içerir.

Dosya sisteminde ikinci seviyede "adanmış dosyalar" (DF – Dedicated File) bulunmaktadır. DF doğrudan MF'nin altında ya da bir başka DF'nin altında bulunabilirken başka EF (Elementary File) ve DF'leri içerebilir. Teorik olarak sonsuz sayıda DF birbirini içerebilir fakat sınırlı hafıza nedeniyle iç içe iki DF'den fazlasına pek rastlanmaz.

Uygulamalar için gerekli olan bilgilerin saklandığı dosyalar "temel dosyalar"dır (EF – Elementary File). EF genellikle DF'lerin altına koyulduğu gibi doğrudan MF'nin altında da bulunabilir. EF'nin uygulamadan bağımsız, kendine ait bir yapısı vardır. Bu yönüyle bilgisayar işletim sistemleri uygulama dosyalarından ayrılır.

Uygulamalara ait bilgilerin yanısıra işletim sistemi dosyaları, anahtar dosyaları ve kod dosyaları da EF dosyası formatında saklanır.

DF dosyaları uygulamaya özel oluşturulur. Yani bir uygulamaya ait tüm dosyalar bir DF dosya altında saklanmaktadır. Bu sayede düzenli bir dosya sistemi tutulabilmektedir.

Akıllı kartlarda 4 standart EF dosya yapısı vardır.

- Şeffaf dosyaların bir iç yapısı yoktur. Dosyadaki veriler baytlar ya da bloklar halinde okunup yazılabilir. Bu dosyalarda herhangi bir biçimi olmayan veriler, örneğin kart sahibinin dijital fotoğrafı saklanabilir. İkili okuma / yazma komutlarıyla (READ_BINARY, WRITE_BINARY, UPDATE_BINARY) kullanılırlar.

- Lineer sabit dosyalarda sabit uzunluklu kayıtlar bulunur. Ulaşılabilecek en küçük birim bir kayıttır. Bir kayıt en fazla 254 bayt olabilir ve bir dosyada en fazla 254 kayıt bulunabilir. Kayıtlar kayıt okuma / yazma komutlarıyla (READ_RECORD, WRITE_RECORD, UPDATE_RECORD) yönetilir.

- Lineer değişken dosyalar da kayıtlardan oluşur fakat kayıtların boyutu değişkendir. Bu yüzden her kayıt için bir de kayıt uzunluğu bilgisi saklanır.

Değişken uzunluklu veriler için, örneğin telefon defterindeki isim bilgileri, uygun dosya yapısıdır. Kayıtlar kayıt okuma / yazma komutlarıyla (READ_RECORD, WRITE_RECORD, UPDATE_RECORD) yönetilir.

- Döngüsel dosyalar lineer sabit dosyalara benzer, yani her kayıt sabit uzunluktadır. Dosya ek olarak son kaydı gösteren bir gösterici içerir. Bu kaydın numarası 1'dir. Bir önceki tutulan kaydın numarası 2'dir. n kayıt bulunan bir dosyadaki en eski kaydın numarası n'dir. Dosyanın döngüsel bir yapısı vardır, yani dosyanın sonuna ulaşılınca tekrar başa dönülür ve eski kayıtların üzerine yazılır. Kayıtlar kayıt okuma / yazma komutlarıyla (READ_RECORD, WRITE_RECORD, UPDATE_RECORD) yönetilir.

Dosya isimleri :

Akıllı kart işletim sisteminde dosyalar mantıksal isimlerle işaret edilir. Her dosyanın 2 baytlık "dosya tanımlayıcısı" (FID – File Identifier) vardır. Dosyalara tanımlayıcı değer atamada çeşitli kısıtlamalar bulunmaktadır:

- '3F00h' dosya tanımlayıcısı değeri MF için ayrılmıştır.
- Aynı DF içinde aynı FID değerine sahip birden fazla dosya bulunamaz. Ayrıca bir DF ile hemen altındaki bir EF'nin FID değerleri aynı olamaz.
- Bir klasör altındaki tüm EF'ler farklı FID'lere sahip olmalıdır.
- İç içe bulunan DF'ler aynı FID'ye sahip olabilirler.

Hızlı işlem gerektiren durumlarda EF dosyaları için 5 bit uzunluğunda kısa FID değerleri kullanılabilmektedir.

Bilgi amaçlı olarak, adanmış dosyaların gövde bölümlerine 1 ile 16 bayt arası uzunlukta olabilen "DF isimleri" eklenebilir.

Dosyaların seçilmesi :

Akıllı kart işletim sistemlerinde bir dosya ile ilgili bir işlem yapılmadan önce ilgili dosya seçilmelidir. Belirli bir zamanda en fazla bir dosya seçili olabilir. Yeni bir dosyanın seçilmesi bir önceki seçimi iptal eder.

MF, FID değeri eşsiz olduğu için dosya ağacında herhangi bir yerden doğrudan seçilebilir. MF'nin hemen altındaki DF'ler MF'den ya da aynı seviyedeki bir başka DF'den sonra seçilebilir. DF'ler FID değerlerinin yanısıra DF isimleri kullanılarak da seçilebilir.

Açıkça seçmenin yanısıra dosyalar, komutların içerisinde kısa FID değerini kullanarak üstü kapalı bir şekilde de seçilebilmektedir. Bu sayede iki aşamalı işlem bir komutla gerçekleştirilebilir.

4.5 Akıllı Kart Uygulama Geliştirme Ortamları

Akıllı kartlar üretim amaçlarına yönelik fonksiyonları içerecek şekilde üretilir. Bununla beraber üretim sonrası kartlara uygulama yükleme imkanı veren çeşitli geliştirme ortamları vardır.

4.5.1. Java card uygulama geliştirme ortamı

Java platformdan bağımsız bir dildir. Bu dilde yazılan bir uygulama Java derleyicisinde derlenir ve ikili (binary) Java kodu elde edilir. Bu ikili kod, Java yorumlayıcısı bulunan her ortamda çalışabilir.

Java Card teknolojisi, Java programlama dilinde yazılan uygulamaların sınırlı kaynaklara sahip olan akıllı kartlarda çalıştırılmasına imkan tanır. Java Card üç bölümden oluşur:

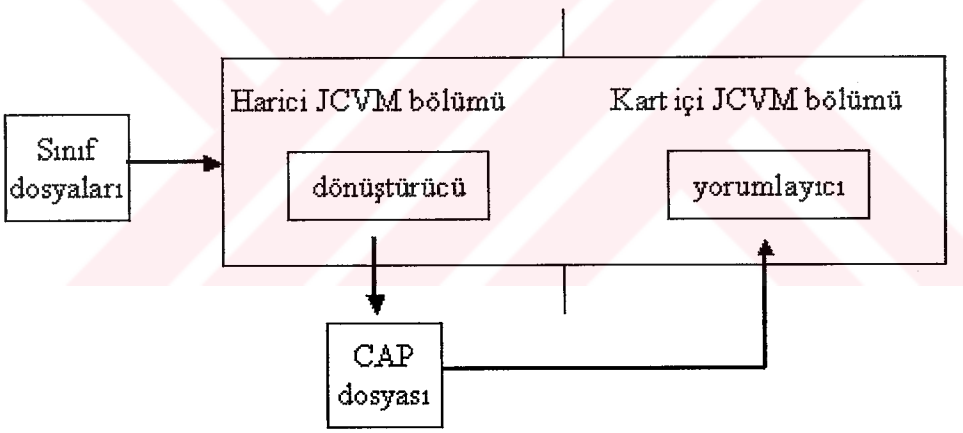
- Java Card Sanal Makinası (JCVM – Java Card Virtual Machine), Java dilinde yazılmış olan kaynak kodun ikili koda dönüştürülmesi, bu kodun java uyumlu akıllı karta yüklenmesi ve çalıştırılması bölümlerini gerçekleştirir.

- Java Card Çalışma Zamanı Ortamı (JCRE – Java Card Runtime Environment), java uyumlu akıllı kartın bellek yönetimi, uygulama yönetimi gibi çalışma zamanı davranışlarını kontrol eder.

- Java Card Uygulama Geliştirme Arayüzü (API – Application Protocol Interface), akıllı kartlar üzerinde uygulama geliştirecekler için tasarlanmış temel Java paketleri ve sınıflarından oluşmuştur.

Java card sanal makinası (JCVM)

Java kaynak kodu herhangi bir Java derleyicisiyle derlenir ve uygulamaya ait Java sınıfı üretilir. Daha sonra Şekil 4.22'de görüldüğü gibi, dönüştürücü bir program bu sınıfı akıllı kartın desteklediği CAP (Converted Applet) dosya formatına dönüştürür ve dosya akıllı karta yüklenir. Akıllı karta yüklenen bu ikili dosya bir diğer program tarafından yorumlanmak suretiyle çalıştırılır.



Şekil 4.22 JCVM yapısı

Kaynak tüketimi fazla olduğu için kart dışında çalıştırılan dönüştürme programı ve çalışma zamanında başvurulduğu için kart içinde olması gereken yorumlama programı birlikte Java Card Sanal Makinası'nı oluştururlar. Yani JCVM biri kart dışında ve diğeri kart içinde olmak üzere iki kısımdan oluşmaktadır.

Java card çalışma zamanı ortamı (JCRE)

JCRE akıllı kartın içinde çalışmakta olan Java Card sistem bileşenlerinden oluşur. JCRE, kart kaynak yönetimi, şebeke haberleşmesi, uygulamaların güvenliği ve çalıştırılmasından sorumludur. Yani JCRE akıllı kartın işletim sistemi gibi davranmaktadır.

JCRE kart ömründe sadece bir defa başlatılır. Besleme kesildiğinde JCRE tamamen durmaz, askıya alınır ve JCVM'nin durumu saklanır. Karta tekrar enerji geldiğinde JCRE, JCVM'yi kaldığı yerden başlatır. Sıfırlama sırasında yarım kalmış bir işlem varsa, tekrar enerji gelme anında bu işlem tamamlanır.

Karta enerji geldikten ve sıfırlama işlemleri gerçekleştirildikten sonra JCRE konuk bilgisayarından gelecek olan komutları beklemek üzere döngüye girer. Gelen komutla ilgili uygulamayı etkin hale getirir ve komutu bu uygulamaya gönderir. Uygulamanın verdiği yanıtı konuk bilgisayara iletir.

Java card uygulama geliştirme arayüzü (API)

Java Card API'ları ISO 7816 standardını destekleyecek uygulamalar yazmaya imkan sağlayacak şekilde geliştirilmiş sınıflardan oluşmuştur. Bu API'larda üç ana paket ve bir genişletme paketi bulunmaktadır.

"Java.lang" paketi temel Java programlama dili özelliklerini içermektedir. "Javacard.framework" paketi uygulama çalıştırmakta ve çalışma esnasında uygulamanın JCRE ile etkileşimini sağlamaktadır. "Javacard.security" paketi temel kriptografik fonksiyonları desteklemektedir. Bir genişletme paketi olan "Javacardx.crypto" ise gelişmiş kriptografik fonksiyonları sağlamaktadır.

Java card dili alt kümesi

Küçük hafıza boyutu nedeniyle Java Card platformu dikkatlice seçilmiş, akıllı kartlara göre uyarlanmış bir Java dili alt kümesini içermektedir. Bu alt küme aynı zamanda nesne yönelimli programlama tekniğini de barındırmaktadır.

Desteklenen bazı Java dili özellikleri şunlardır:

- Küçük boyutlu temel veri tipleri (boolean, byte, short),
- Bir boyutlu diziler,
- Java paketleri, sınıfları, arayüzleri ve istisnaları,
- Süpürme fonksiyonları (bazı Java Card'lar desteklemektedir),
- Nesne yönelimli özellikler (miras alma, sanal fonksiyonlar, fonksiyon aşırı yükleme, dinamik nesne oluşturma vb.)

Desteklenmeyen bazı Java dili özellikleri şunlardır:

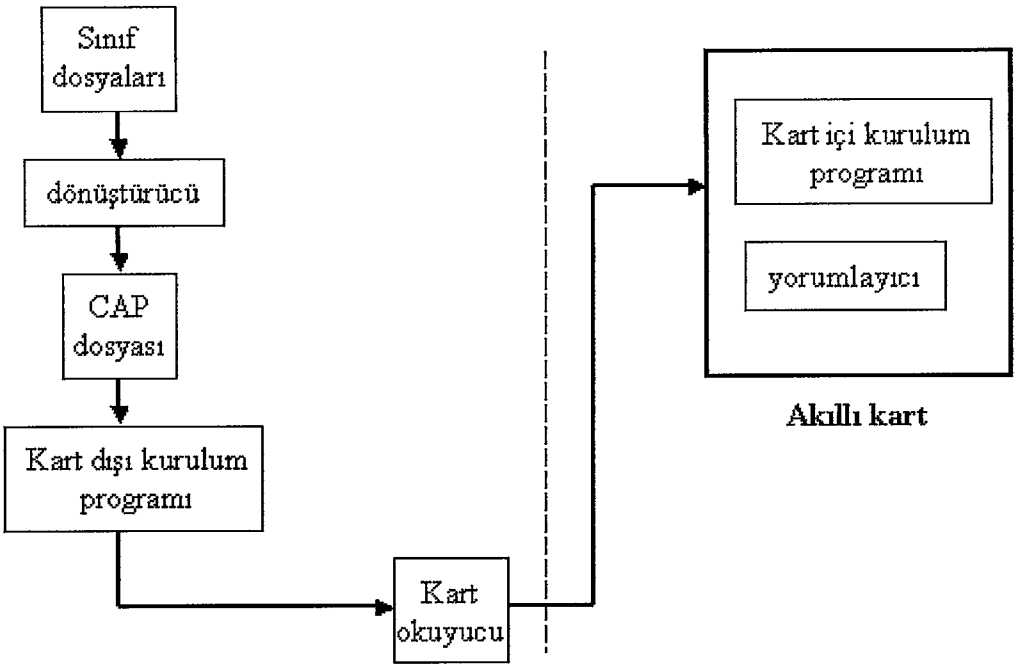
- Büyük boyutlu temel veri tipleri (long, double, float),
- Karakterler ve karakter dizileri,
- Çok boyutlu diziler,
- Dinamik sınıf yükleme,
- Çoklu görevli çalışma,
- Nesne kopyalama,
- Nesne sonlandırma fonksiyonları.

Akıllı karta uygulama yüklenmesi aşamaları

Java Card uygulama yükleyici programı kartın içinde bulunur. Bu program kart dışındaki bir diğer yükleme programıyla birlikte çalışır. Kart dışı yükleme programı CAP dosyasının içeriğini kart okuyucusu aracılığıyla kart içi yükleme programına iletir. Yükleme programı bu bilgiyi kartın hafızasına yazar ve JCRE tarafından kullanılacak olan veri yapılarını oluşturur.

Yorumlayıcı ve yükleyici programları ayırmak yükleyici programlar için esneklik sağlarken yorumlayıcı programı da küçük boyutlu tutmaktadır.

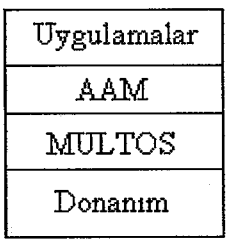
Akıllı karta uygulamanın yüklenmesi aşamaları Şekil 4.23'te gösterildiği gibidir.



Şekil 4.23 Java Card uygulama yükleme aşamaları

4.5.2 MULTOS uygulama geliştirme ortamı

MULTOS işletim sistemi, çoklu uygulamalı akıllı kartlar için alt katmanda gerçekleştirilen haberleşme, bellek yönetimi, uygulama yönetimi işlemlerini düzenler. Uygulama yükleme, uygulama silme, uygulamaya APDU komutlarını gönderme ve APDU cevaplarını terminale yönlendirme işlemleri yine MULTOS işletim sistemi tarafından yapılmaktadır. MULTOS akıllı kart yapısı Şekil 4.24'te gösterildiği gibidir.



Şekil 4.24 MULTOS akıllı kart yapısı

MULTOS uygulama özet makinası (AAM - Application abstract machine)

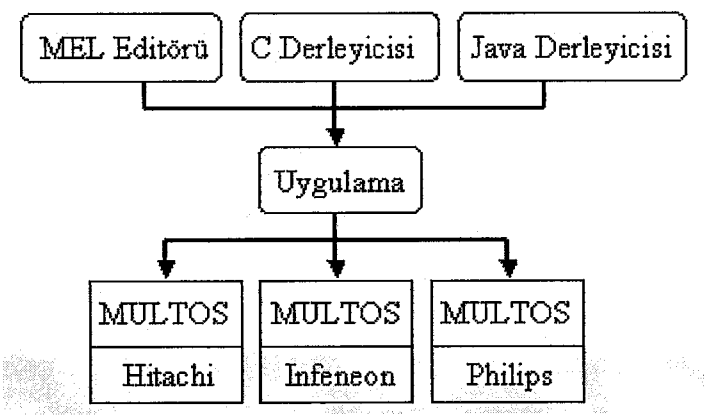
AAM, MULTOS işletim sistemine sahip akıllı kartlar için geliştirilmiş olan uygulama yorumlayıcıdır. AAM için yazılmış olan MULTOS uygulamaları her MULTOS kart için aynı şekilde çalışacaktır. Uygulama geliştiriciler alt katmandaki değişikliklerden etkilenmez.

MULTOS AAM, tüm MULTOS platformları için aynı şekilde çalışacak olan komut ve kütüphane fonksiyonlarından oluşmuş standart bir uygulama geliştirme arayüzü sağlamaktadır. MULTOS uygulamaları MEL (MULTOS Executable Language) diye adlandırılmış olan bayt kodunda yazılır. Uygulamalar MEL koduna dönüştüren derleyici kullanmak kaydıyla yüksek seviyeli C ve Java dillerinde de yazılabilmektedir.

MULTOS AAM, her uygulamanın kod ve bellek bölgelerini ayrı ayrı tahsis eder. Bir uygulama bir başka uygulamanın bellek bölgesine erişemediği gibi kendi kod belleğine de erişemez. Tüm uygulamaların erişebildiği ortak kullanımlı bellekler bulunmaktadır. Bu bellekler sayesinde uygulamalar birbirleriyle ve kartın bağlı bulunduğu terminal ile haberleşebilmektedir. Bu mimari sayesinde hatalı çalışan bir uygulamanın diğer uygulamaları etkilemesi önlenmiş olur.

MULTOS uygulama geliştirme yöntemleri

MULTOS uygulamaları doğrudan MEL dilinde yazılabileceği gibi C ya da Java dillerinde yazılıp derlenerek de MEL diline dönüştürülebilir. MEL, kırmık üzerinde çalışan sonuç koddur ve Şekil 4.25'te de görüldüğü gibi, platform bağımsızdır.



Şekil 4.25 MULTOS uygulama geliştirme yöntemleri

MULTOS uygulama geliştiriciler için standart bir uygulama geliştirme arayüzü (API) sağlar. MULTOS API'da bulunacak komut ve fonksiyonlar MULTOS Konsorsiyum tarafından kararlaştırılır. MULTOS ayrıca DES, 3DES, RSA şifreleme ve şifre çözme, rastlantısal sayı üretme gibi kriptoloji uygulamaları için kullanılmak üzere temel fonksiyonlar sağlamaktadır.

MULTOS dinamik uygulama yükleme mekanizması

MULTOS kartlara çalışma ömürleri boyunca herhangi bir zamanda uygulama yüklenebilir ya da üzerlerindeki bir uygulama kaldırılabilir. Bunun ötesinde uygulamalar internet ya da telefon hattı gibi açık ve güvensiz ortamlarda güvenli bir şekilde yüklenip kaldırılabilir.

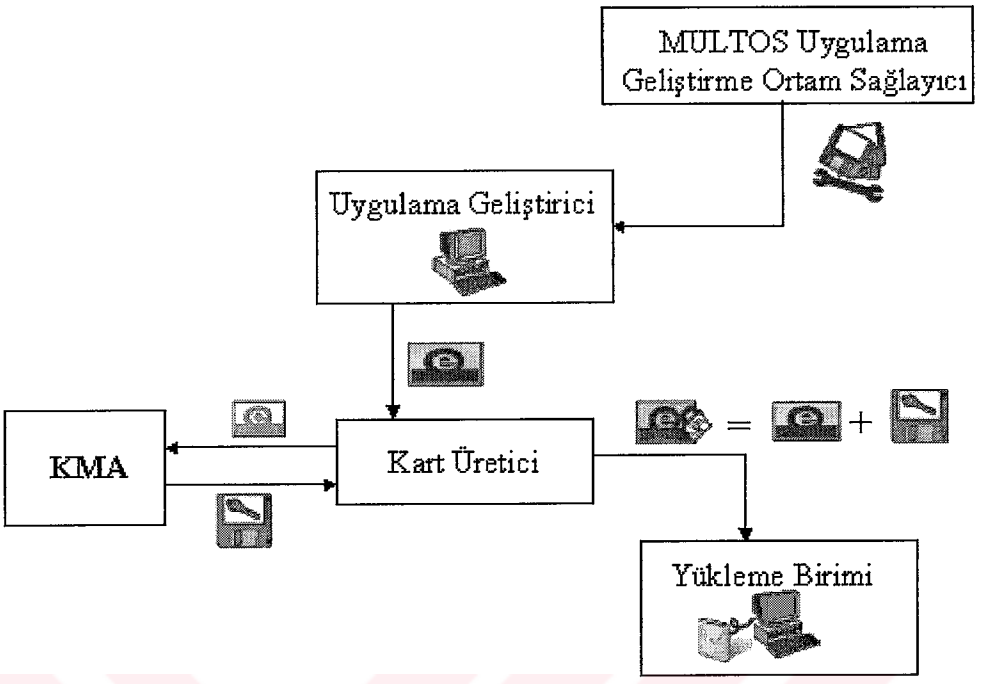
Kart üreticisi şifreleme ve imzalama mekanizmalarından faydalanarak, karta yüklenecek olan uygulamanın güvenliğini ve bütünlüğünü sağlayabilir. Veri ve uygulama, yükleme yapan tarafta şifrelenir ve kartın içinde bu şifre çözülür. Bu işlem uçtan uca tam bir güvenlik sağlamaktadır.

MULTOS uygulama yükleme ve kaldırma alt yapısı asimetrik anahtar şifreleme algoritmalarıyla gerçekleştirilmektedir. Karta uygulama yüklemek için uygulama yükleme sertifikası (ALC - Application Load Certificate) ve karttaki uygulamayı kaldırmak için uygulama kaldırma sertifikası (ADC - Application Delete Certificate) gerekmektedir. Bu sertifikalar güvenilen üçüncü bir kuruluş tarafından sağlanır. MULTOS akıllı kartları için bu kuruluş MULTOS Anahtar Yönetim Merkezi'dir (KMA - Key Management Agent).

Her sertifika MULTOS KMA'nın gizli anahtarıyla imzalandığı için her MULTOS kart MULTOS KMA'ya ait olan açık anahtar bilgisini içerir. Üreticiyi tanımlayan numara sertifikadan elde edilen numarayla karşılaştırılır. Her iki numara birbirinin aynısı ise akıllı kart uygulamanın yüklenmesine izin verecektir.

MULTOS KMA sertifikayı üretirken uygulamanın çırpma değerini de dikkate alır. Böylece uygulamanın ekleme, çıkarma, değiştirme yapılmaksızın bir bütün olarak yüklenmesi sağlanmış olur.

Karta uygulama yükleme aşamaları Şekil 4.26'da gösterildiği gibidir.



Şekil 4.26 MULTOS uygulama geliştirme aşamaları

Uygulama geliştirici bir geliştirme ortamı sağlayıcısı ile irtibat kurulup MULTOS uygulama geliştirme ortamını elde etmelidir. Uygulama geliştirme doğrudan MEL dilinde yapılabileceği gibi, Java ya da C dilinde yapılıp bu kod derleyici yardımıyla MEL diline de dönüştürülebilir. Uygulama geliştirici, yazmış olduğu programı kart üreticisine teslim eder.

Kart üreticisi, MEL dilinde yazılmış olan kodun çırpma değerini ve kendi üretici tanımlayıcısını MULTOS KMA'ya gönderilir. MULTOS KMA çırpma değeri ve üreticiyi tanımlayan bilgiyi kendi kapalı anahtarıyla imzalar ve kart üreticisine geri gönderir. Bu gönderilen imzalı veri, yüklenmeye hazır MULTOS uygulamasına eklenir. MULTOS kart, kendisine gönderilen MULTOS KMA imzalı uygulamanın yüklenmesine izin verir.

5. UYGULAMA

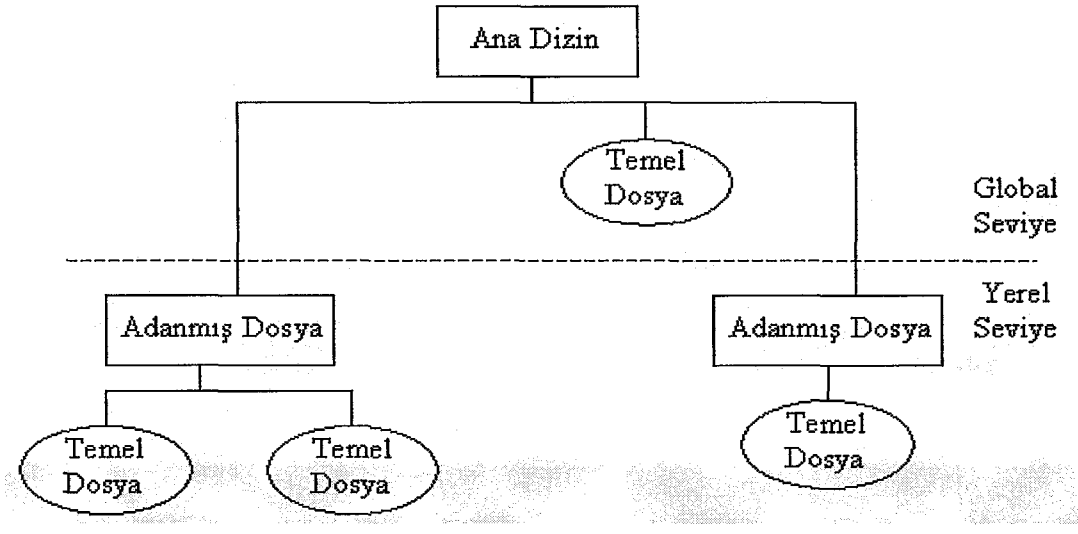
MPCOS akıllı kartlar kullanılarak bir şebeke güvenliği uygulaması geliştirilmiştir. Bu bölümde öncelikle uygulamada kullanılan kartların işletim sistemi incelenmiş, bir uygulama yapılmış, geliştirme aşamaları açıklanmıştır.

5.1 MPCOS İşletim Sistemi

MPCOS (Multi-application Payment Chip Operating System), Gemplus firması tarafından geliştirilmiş bir akıllı kart işletim sistemidir. ISO 7816 standardı ile uyumlu veri yapıları, komutları ve geri dönüş kodları vardır.

5.1.1 Dosya yapıları

MPCOS dosyaları 2 seviyeli bir yapıda değerlendirilir. Ana dosya (master file) ve doğrudan ana dosyaya bağlı temel dosyalar (elementary file) global seviyeyi oluşturur. Adanmış dosyalar (dedicated file) ve bunlara bağlı temel dosyalar da yerel seviyeyi oluşturur. MPCOS işletim sisteminin dosya yapısı Şekil 5.1'de gösterildiği şekildedir.



Şekil 5.1 MULTOS dosya yapısı

Ana dosya, MPCOS dosya yapısının kök dizinidir. Her kartta bir tane bulunur ve toplam en fazla 63 tane olmak üzere temel dosyalar ve adanmış dosyalar içerebilir.

Adanmış dosya, belirli bir uygulama ile alakalı tüm temel dosyaların tutulduğu klasör yapısıdır. Bir adanmış dosyada en fazla 63 temel dosya tutulabilir ve bir adanmış dosya bir başka adanmış dosyayı içeremez. Her adanmış dosya 16 baytlık bir tanımlayıcı kısım ve en fazla 16 bayt uzunluğunda olabilen bir gövde kısmından oluşur. Gövde bölümü adanmış dosya isim bilgisini içerir.

Temel dosyalar MPCOS dosya yapısının uygulama verilerinin tutulduğu dosyalardır. 16 baytlık bir tanımlayıcı kısımları ve verilerin bulunduğu bir gövde kısımları bulunur. Gövde kısmında taşınan verinin hitap ettiği kullanıcıya göre temel olarak iki gruba ayrılırlar. Bunlar, kart tarafından değerlendirilen ve kullanım amacına göre belirli formatlarda bulunan hassas dosyalar ile kullanıcıya ait bilgilerin tutulduğu, standart dosya formatlarından oluşan veri dosyalarıdır.

Hassas dosyalar

Bu dosyalar akıllı kart işletim sistemi tarafından değerlendirilir. Kullanıcı bu dosyalara doğrudan erişemez fakat bunları dolaylı olarak kullanır. Kullanım amaçlarına göre çeşitli formatları vardır:

- Anahtar dosyaları, MPCOS şifreleme fonksiyonları tarafından kullanılan anahtarların tutulduğu dosyalardır. Ana dizin ve her bir adanmış dosya bir ya da daha çok anahtar dosyası içerebilir. Her bir anahtar dosyası en fazla 4 adet 3DES şifreleme algoritması anahtarı içerebilmektedir.

- Gizli kod dosyaları, ana dizinde ve her bir adanmış dosyada en fazla 1 adet olabilir. Bir gizli kod dosyası en fazla 8 gizli kod içerebilmektedir. Bazı dosyalara çeşitli erişim izinleri bu gizli kodlardan bazılarının bilinmesi şartıyla verilebilir.

Veri dosyaları

Bu dosyalar uygulama tarafından değerlendirilir. Kullanıcı bu dosyalara doğrudan erişebilmektedir. Veri saklama şekillerine göre çeşitli formatları vardır:

- Saydam dosyaların belirli bir biçimleri yoktur, sıralı bayt dizilerinden oluşurlar. Bu dosyalardaki verilere, dosya başından itibaren göreli konum değerleri ile ulaşılmaktadır.

- Doğrusal sabit dosyalar, sabit uzunluklu kayıtlardan oluşur. Her kaydın oluşturuluş sırasına göre bir kayıt numarası bulunur ve kayıtlara bu numaralarla erişilir. En fazla 255 kayıt tutulabilir. Kayıt uzunluğu dosya oluşturma sırasında belirlenir.

- Doğrusal değişken dosyalar, değişken uzunluklu kayıtlardan oluşur. Kayıtlara kayıt numaralarıyla erişilmekte, bellek daha etkin olarak kullanılmaktadır. Fakat kayıtlara ulaşma süresi doğrusal sabit kayıtlara oranla daha uzundur.

- Çevrimsel temel dosyalar, sabit uzunluklu kayıtlardan oluşur ve en fazla 32 kayıt tutulabilir. Bu sınır aşıldığında eski kayıtların üzerine yazılacaktır. Kayıt numaraları en son girilen kayıttan itibaren verilmekte ve her kayıt girişinde güncellenmektedir. Yani en son girilen kayıt 1 numaralı kayıttır. Kayıt uzunluğu dosya oluşturma sırasında belirlenir.

5.1.2 Dosya tanımlayıcıları

MPCOS işletim sisteminde her dosyanın Şekil 5.2'de gösterilen yapıda, dosya tanımlayıcısı denilen 16 baytlık bir başlık kısmı vardır. İşletim sistemi bu tanımlayıcı kısımdaki bilgileri kullanarak dosya işlemlerini yönetir.

<table>
<tr><td>FP</td><td>FN</td><td colspan="2">Dosya Tanıtıcı</td></tr>
<tr><td>FDB</td><td>RL</td><td colspan="2">Gövde Uzunluğu</td></tr>
<tr><td colspan="2">AC 1</td><td colspan="2">AC2</td></tr>
<tr><td colspan="2">AC3</td><td>RFU</td><td>CS</td></tr>
</table>

Şekil 5.2 MULTOS dosya tanımlayıcı yapısı

FP (File Pedigree) bölümü, işletim sisteminin dosyayı hiyerarşik yapıda doğru bir şekilde yerleştirebilmesini sağlar. Şekil 5.3'te gösterilen 1 baytlık bu bölümün ilk biti olan F, dosyanın çeşidini belirtir. Adanmış dosyalar için bu değer 1 iken temel

dosyalar için 0 değeri verilir. İkinci bit olan L, dosyanın seviyesini belirtir. Yerel dosyalar için bu değer 1 iken global dosyalar için 0 değeri verilir. Kalan 6 bitlik bölüm, dosyanın bulunduğu hiyerarşide bir üst seviyedeki adanmış dosyanın dosya numarasını içermektedir.

F	L	Üst Seviye Dosya Numarası

Şekil 5.3 FP bölümü yapısı

FN (File Number) bölümü dosya numarası bilgisini içerir ve 1 bayttır. Bu bölümün son 6 biti dosyanın oluşturulma sırasına göre değer alır. FP – FN çifti tüm dosya yapısı içinde her dosya için farklıdır ve işletim sistemi bu 2 baytlık bilgiye göre dosyaları ayırt etmektedir.

Dosya tanıtıcı bölüm 2 bayt uzunluktadır ve işletim sistemi komutlarında dosyayı belirtmek için bu bölgedeki değer kullanılır. Bu bölümün son 5 biti kısa dosya tanıtıcı bölümdür.

FDB (File Description Byte) bölümü dosya tanımlayıcı bayttır ve Şekil 5.4'te görüldüğü gibi, dosyanın çeşidine göre değer almaktadır.

b7	b6	b5	b4	b3	b2	b1	b0	Dosya Tipi
0	0	1	1	1	0	0	0	Adanmış dosya
x	x	1	0	1	x	x	x	Anahtar dosyası
x	x	1	0	0	x	x	x	Gizli kod dosyası
x	x	x	x	x	0	0	1	Saydam dosya
x	x	x	x	x	0	1	0	Doğrusal sabit dosya
x	x	x	x	x	1	0	0	Doğrusal değişken dosya
x	x	x	x	x	1	1	0	Çevrimsel dosya

Şekil 5.4 FDB bölümü yapısı

RL (Record Length) bölümü sabit uzunluklu kayıtlara sahip doğrusal dosyalar için kullanılır ve kayıt uzunluk bilgisini taşır.

Gövde uzunluğu bölgesi 2 bayt uzunluktadır ve dosyanın gövde bölümünün uzunluğu bilgisini taşır. Adanmış dosyalar gövde kısmında dosya isimlerini bulundurduklarından bu dosyalar için bu bölüm dosya isim uzunluğudur.

Her biri 2 bayttan oluşan 3 AC (Access Condition) erişim koşulu bölgesi vardır. Bu bölgeler dosyalar üzerinde çeşitli işlemler yapmadan önce girilmesi gereken kodların bulunduğu gizli kod dosyalarını ya da komutları oluştururken 3DES şifreleme algoritmasından geçirmek gerekiyorsa ilgili anahtarların bulunduğu anahtar dosyalarını belirtmektedir.

RFU (Reserved For Future Use) bölgesinin bir işlevi yoktur, daha sonra kullanılmak üzere boş bırakılmıştır.

CS (Checksum) bölgesi, dosya tanımlayıcısının ilk 15 baytındaki 0 bitlerinin toplamıdır ve hata kontrolü amacıyla kullanılmaktadır.

5.1.3 Anahtar dosyaları

Anahtar dosyaları, tüm MPCOS şifreleme fonksiyonları tarafından kullanılan anahtarların tutulduğu dosyalardır. Adanmış ve temel dosyalar altında istenilen sayıda anahtar dosyası bulunabilir. Her bir dosyada en fazla 4 adet 3DES anahtarı saklanabilir.

Temel olarak 3 çeşit anahtar vardır. Yönetimsel anahtarlar güvenli mesajlaşmada, doğrulama anahtarları kart ya da terminalin yetkisini ispatlamasında, imza anahtarları ise imzalamada kullanılır.

3DES anahtarları, iki DES anahtarının anahtar dosyasına ard arda kaydedilmesiyle saklanır. Şekil 5.5'te görüldüğü gibi, her bir anahtar 8 bayt ve her anahtarın başlık kısmı 4 bayt olmak üzere bir 3DES anahtarı için toplam 24 bayt yer ayrılır.

Anahtar Tipi 1	00h	Kv	Cks
K16	K15	K14	K13
K12	K11	K10	K9
Anahtar Tipi 2	00h	Kv	Cks
K8	K7	K6	K5
K4	K3	K2	K1

Şekil 5.5 3DES anahtarı saklama biçimi

Anahtar tipi, Şekil 5.6'da gösterildiği şekilde kodlanmalıdır. 3DES anahtarını oluşturan her iki DES anahtarı aynı tip olmalıdır.

b7	b6	b5	b4	b3	b2	b1	b0	Anahtar Tipi
x	x	x	0	x	x	0	0	Yönetimsel anahtar
x	x	x	1	x	x	x	x	İmza anahtarı
x	x	x	0	x	x	1	1	Doğrulama anahtarı

Şekil 5.6 Anahtar tipinin kodlanması

Kv (Key Version) bölümü 1 baytlık anahtar versiyon numarasıdır. Terminal anahtarı değiştirince yeni anahtar versiyonunu bu bölgeye yazabilir.

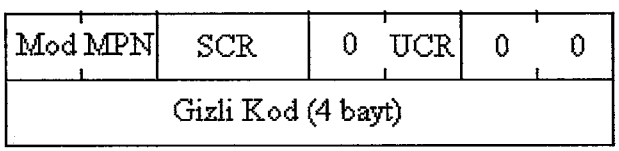
Cks (Checksum) bölümü 1 baytlık hata kontrol bölümüdür. Kendinden önceki 3 baytın dışlamalı VEYA (exclusive OR) işleminden geçirilip sonucun tersinin alınmasıyla elde edilir.

K16 – K1 değerleri 16 baytlık 3DES anahtarıdır.

5.1.4 Gizli kod dosyaları

Ana dizinde ve her bir adanmış dosyada en fazla 1 gizli kod dosyası olabilir. Birden fazla olması durumunda ilk oluşturulan dosya geçerlidir. Gizli kod dosyaları en fazla 8 gizli kod içerebilir.

Şekil 5.7'de gösterildiği gibi, her gizli kod 4 bayt başlık ve 4 bayt kod bölümü olmak üzere 8 bayttan oluşur. Kod bilgisine doğrudan erişime izin verilmez. Gizli kodlara, oluşturulma sırasına göre 0 ile 7 arasında bir adres verilir ve ulaşım bu adresler kullanılarak yapılır.



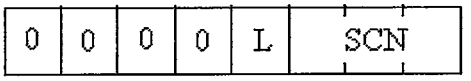
Şekil 5.7 Gizli kod saklama biçimi

Mod bölümü 4 bitlik bir bölümdür ve gizli kodun doğrudan mı yoksa şifrelenmiş halinin mi girilmesi gerektiğini belirtir.

MPN (Maximum Presentation Number) bölümü 4 bitliktir ve ard arda kaç defa yanlış girilme olursa bu kodun kilitleneceği bilgisini taşır. Bu bölümün son 3 biti kullanılarak 1 ile 7 arası bir giriş yapılır.

SCR (Secret Code Ratification), ard arda girilen yanlış kod sayısını tutan 1 baytlık sayaçtır. Her yanlış kod girişinde bu sayaç bir arttırılırken her doğru kod girişinde bu sayaç sıfırlanır. MPN değerine ulaşıldığında gizli kod kilitlenir.

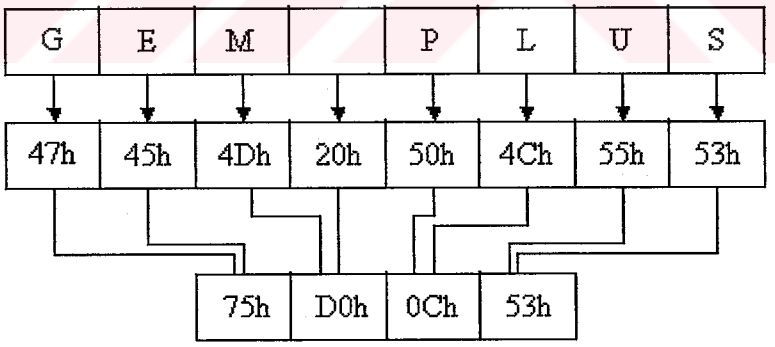
UCR (Unlock Code Reference) bölümü, kilitlenen bir gizli kodu açmak için gerekli olan gizli kodun bulunduğu yeri belirtir. 1 baytlık bu bölüm Şekil 5.8'de belirtildiği gibi kodlanmaktadır.



Şekil 5.8 UCR bölümü yapısı

SCN (Secret Code Number), kilitlenmiş kodu açmak için kullanılacak gizli kodun sıra numarasını gösterirken L biti ilgili dosyanın seviyesini belirtir. L biti 1 ise ilgili dosya yerel, 0 ise globaldir.

Gizli kod 4 baytlık bir değerdir fakat terminal her seferinde 8 baytlık bir veri girmelidir. MPCOS işletim sistemi, girilen her bir baytın düşük anlamlı 4 bitini alarak 4 baytlık gizli kod değerini elde eder. Şekil 5.9'da girilen 8 baytlık veriden 4 baytlık gizli kod değerinin elde edilişi örneği verilmiştir.



Şekil 5.9 8 baytlık girişten 4 baytlık gizli kod elde edilmesi örneği

5.1.5 Erişim koşulları

MPCOS işletim sistemi, dosya erişiminin gizli kodlar ve şifreleme anahtarlarıyla güvenli hale getirilmesine imkan tanır. Bu işlemler gerçekleştirilirken gizli kod dosyaları ve anahtar dosyaları kullanılmaktadır.

Terminal bir dosyadaki veriye ulaşmak istediğinde, kart, ilgili dosyanın tanımlayıcısında bulunan erişim koşullarında belirtilen yetki yazmacını kontrol eder, ilgili bitler '1' yapılmışsa dosyaya erişime izin verir. MPCOS kartların her biri 8 bit olan iki yetki yazmacı vardır:

- Global yetki yazmacı, ana dizin seçiliyken girilen gizli kodların sonuçlarının tutulduğu yazmaçtır. Bir defa 1 yapılan bitler kart tekrar başlatılana kadar sıfırlanmaz.

- Yerel yetki yazmacı, adanmış bir dosya seçiliyken girilen gizli kodların sonuçlarının tutulduğu yazmaçtır. Bu yazmaç her yeni adanmış dosya seçildiğinde sıfırlanır. Aktif olan adanmış dosya tekrar seçilse bile sıfırlama yapılır.

Terminal, 0 ile 7 arasında bulunan gizli kod numarasını da belirterek ilgili gizli kodu girer. Kart girilen değeri kendi gizli kod dosyasındaki değerle karşılaştırır. Gizli kodlar uyuşmazsa kart SCR değerini 1 arttırır ve hata kodu döndürür. Gizli kodlar uyuşursa kart SCR değerini sıfırlar ve ilgili yetki yazmacı bitini 1 yapar.

Erişim koşulları Şekil 5.10'da görülen ikişer baytlık yapılardır.

Şekil 5.10 Erişim koşulları kodlanma yapısı

Val bölümü, grup koruma seviyesini belirten 2 bitlik geçerlilik bölümüdür ve şu anlamları vardır:

- 00b Gizli kod koruması yoktur.

- 01b Dosya, SCN1 ile belirtilen gizli kod ile korunmaktadır.

- 10b Dosya, SCN1 ve SCN2 ile belirtilen gizli kodlar ile korunmaktadır.

- 11b Dosyaya erişim izni yoktur.

L1 biti, anahtar dosyasının bulunduğu seviyeyi belirtmektedir. 0 değeri için global, 1 değeri için yerel anahtar dosyası anlaşılır.

Anahtar dosyası bölümü 5 bitlik bir bölümdür ve L1 ile belirtilen seviyedeki anahtar dosyasının kısa tanımlayıcı değeridir. Bu bölümde sıfırdan farklı bir değer varsa, ilgili anahtar dosyasında bulunan 3DES anahtarı kullanılarak güvenli haberleşmeye geçilir. Sıfır olması durumunda normal, şifre korumasız erişim geçerlidir.

L2 ve L3 bitleri ilgili gizli kod dosyalarının seviyelerini belirtir. 0 değeri için global, 1 değeri için yerel gizli kod dosyaları anlaşılır.

SCN1 ve SCN2 bölümleri üçer bitten oluşur ve ilgili gizli kod dosyalarının hangi numaralı kodlarının geçerli olduğunu belirtir.

Adanmış dosyalar için 2, temel dosyalar için 3 grup erişim koşulu vardır. Erişim koşullarının ilişkili olduğu izinler Şekil 5.11'de gösterilmiştir.

Erişim Grupları	Adanmış Dosya İzinleri	Temel Dosya İzinleri
AC 1	Hassas dosya işlemleri	Güncelleme
AC 2	Veri dosyası işlemleri	Yazma ve ekleme
AC 3	Kullanılmıyor	Okuma

Şekil 5.11 MPCOS erişim koşulu grupları

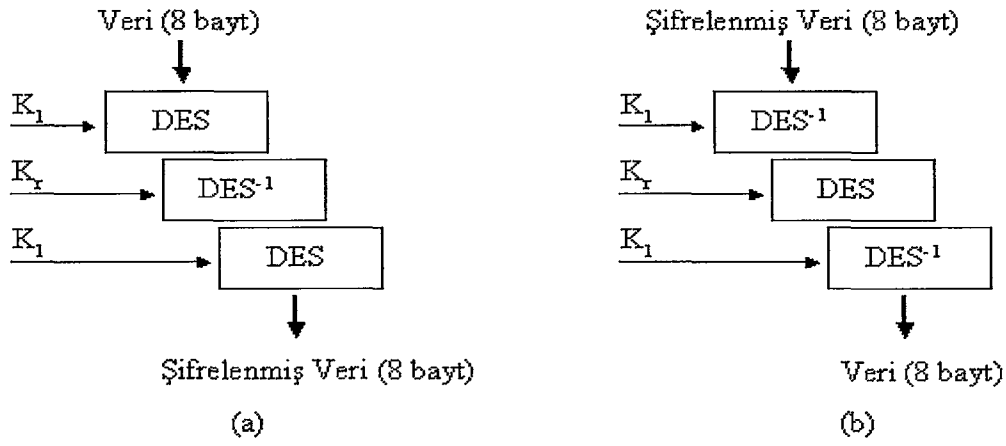
5.1.6 Kriptografi alt yapısı

MPCOS işletim sisteminin DES simetrik anahtar şifreleme yöntemine dayalı bir kriptografi altyapısı vardır.

3DES algoritması :

Bu algoritma, DES algoritmasının "şifreleme – şifre çözme – şifreleme" sırasıyla 3 defa uygulanması ile gerçekleştirilir. Her biri 8 baytlık 2 DES anahtarı kullanıldığı için, algoritmanın anahtar uzunluğu 16 bayttır. Şifreleme ve şifre çözme yapıları Şekil 5.12'de gösterildiği gibidir.

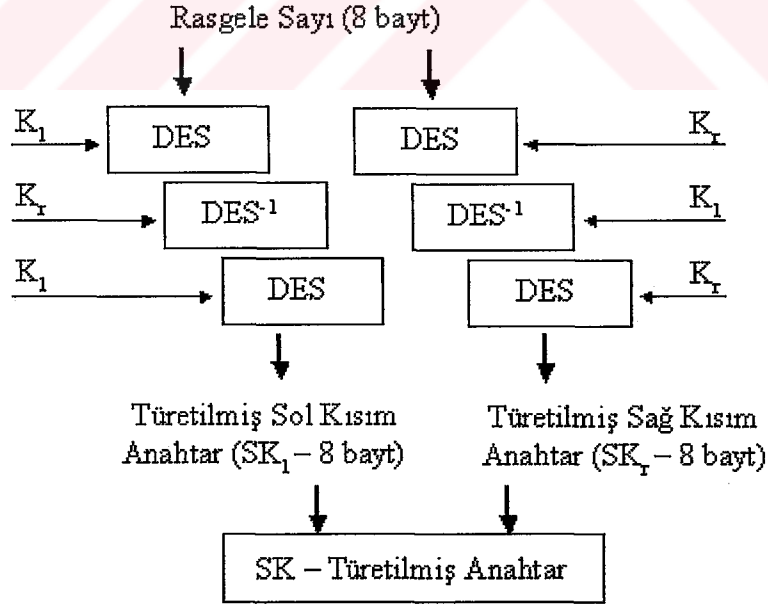
K_l, 16 baytlık K anahtarının yüksek anlamlı 8 baytı iken, K_r, düşük anlamlı 8 bayt değeridir. 3DES algoritmasının şifre çözme aşaması, şifreleme aşamasındaki işlemlerin tersten yapılması ile gerçekleştirilir.



Şekil 5.12 3DES şifreleme ve şifre çözme yapıları

Anahtar türetme :

MPCOS işletim sistemi 3DES anahtarlarını anahtar dosyalarında saklamaktadır. Fakat işletim sistemi komutları bu anahtarları doğrudan kullanmaz. Kart tarafından üretilen 8 baytlık rasgele sayı ile, ilgili anahtar bir türetme işleminden geçirilir, 16 baytlık geçici 3DES şifreleme anahtarı elde edilir ve komutlarda artık bu anahtar kullanılır. Bu şekilde 3DES anahtarının güvenliği arttırılmış olur. 3DES anahtarı türetme aşamaları Şekil 5.13'te gösterildiği gibidir.



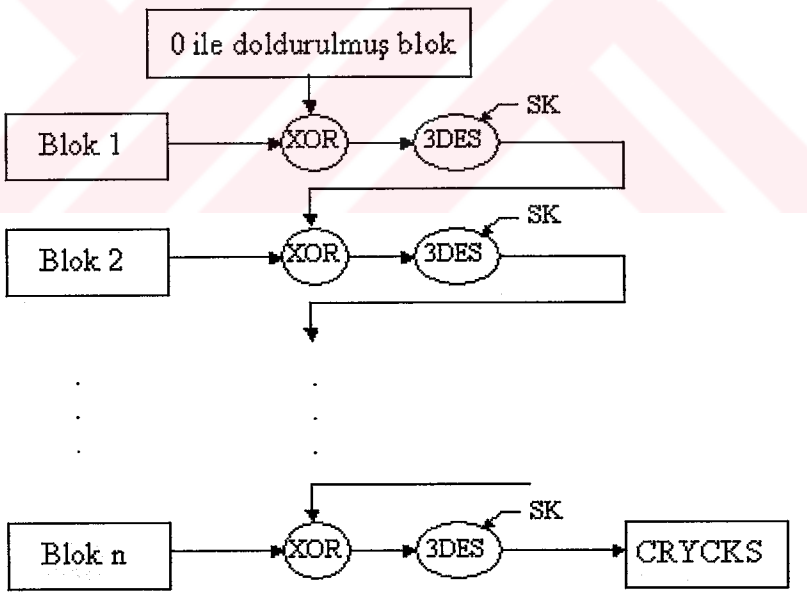
Şekil 5.13 3DES anahtarı türetme yapısı

SK_l, 16 baytlık SK (Session Key) geçici 3DES anahtarının yüksek anlamlı 8 baytı iken, SK_r, düşük anlamlı 8 bayt değeridir.

Kriptografik sağlama toplamı (CRYCKS – Cryptographic checksum) :

MPCOS işletim sistemi, güvenli haberleşme işlemlerinde kriptografik sağlama toplamı değerini kullanmaktadır. Bu değer, kart ile terminal arasında iletilen tüm uygulama katmanı verisi kullanılarak elde edilir.

Kriptografik sağlama toplamı mekanizması Şekil 5.14'te gösterilmiştir. Tüm komut 8 baytlık bloklara bölünür, ilk blok SK geçici anahtarıyla 3DES algoritmasından geçirilir ve bir sonraki blokla mantıksal dışlamalı VEYA (XOR – Exclusive OR) işlemine hazır 8 baytlık sonuç elde edilir. Bir sonraki aşamaya bu mantıksal işlem ile başlanır. Bu işlemler tüm bloklar bitene kadar tekrarlanır. En son blok için elde edilen değer, tüm bloğun 8 baytlık sağlama toplamı değeridir. Sağlama toplamı değeri bulunacak komutun uzunluğunun 8'in tam böleni olmadığı durumlarda son bloğun sonuna sıfır değerleri eklenir.



Şekil 5.14 CRYCKS değerinin hesaplanması

Kart / terminal doğrulama mekanizmaları :

MPCOS kartların, terminal ve kart tarafından bilinen gizli bir anahtarın kullanılması temeline dayanan doğrulama mekanizmaları vardır. Bu algoritmalarda kullanılacak olan 3DES anahtarı doğrulama tipinde olmalıdır.

Dahili doğrulama (internal authentication) mekanizmasında kart terminale kendi doğruluğunu ispatlar. Bunun için kart, terminal tarafından verilen rasgele değeri ve ilgili gizli anahtarı kullanarak 8 baytlık doğrulama değerini elde eder ve bunun son 4 baytını terminale iletir. Aynı hesaplamayı yapan terminal bu değerleri karşılaştırır ve eşitlik durumunda kartın doğruluğundan emin olur.

Harici doğrulama (external authentication) mekanizmasında terminal karta kendi doğruluğunu ispatlar. Bunun için terminal, kart tarafından verilen rasgele değeri ve ilgili gizli anahtarı kullanarak 8 baytlık doğrulama değerini elde eder ve bunun ilk 4 baytını karta iletir. Aynı hesaplamayı yapan kart bu değerleri karşılaştırır ve eşitlik durumunda terminalin doğruluğundan emin olur.

Dahili ve harici doğrulamada hesaplanan değerlerin iletilen kısımları farklıdır. Bu şekilde, harici doğrulamada terminalin kendi doğruluğunu ispatlarken aynı yapıda bir başka kartı kullanması ihtimali engellenmiş olur.

Güvenli mesajlaşma :

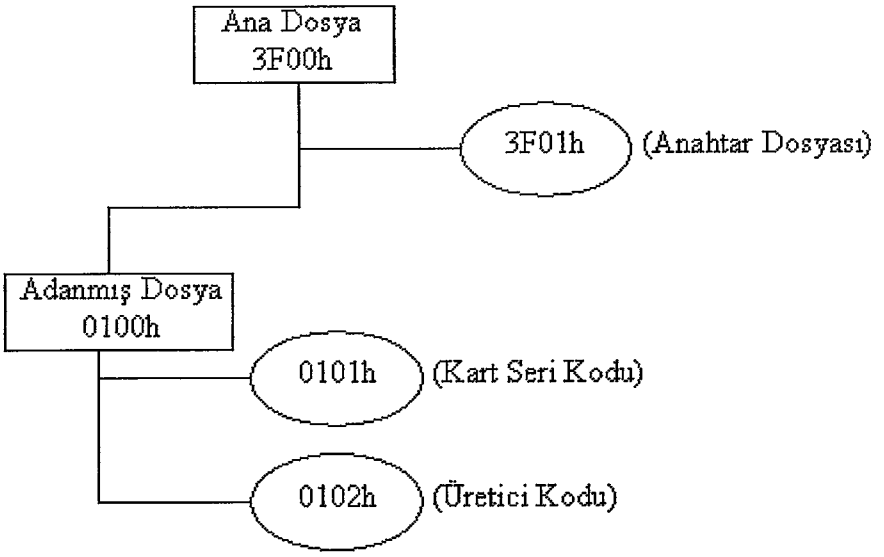
MPCOS işletim sistemi kart ve terminal arasındaki yönetimsel komutları güvenli mesajlaşma mekanizmasıyla korumaktadır. Güvenli mesajlaşma iki şekilde olabilir:

- Karta gönderilen komut şifrelenebilir. Her komut kendine has bir şifreleme yöntemi kullanır, bu yüzden ilgili komutun tanımına bakılmalıdır.

- Alıcı tarafın komutu doğru şekilde aldığından emin olmak için kriptografik sağlama toplamının 3 baytlık bölümü komutun sonuna eklenerek gönderilebilir.

5.1.7 Başlangıç dosya konfigürasyonları

MPCOS kartların başlangıç dosya yapıları Şekil 5.15'te görüldüğü gibidir.



Şekil 5.15 MPCOS başlangıç dosya yapısı

Dosya hiyerarşisinin en üst noktasında 3F00h tanıtıcı değere sahip olan ana dosya bulunmaktadır. Ana dosya altında adanmış ya da temel dosya oluşturmak için, 3F01h anahtar dosyasında bulunan 3DES anahtarı ile güvenli mesajlaşma protokolü kullanılmalıdır.

3F01h tanıtıcı değere sahip temel dosya, global seviye anahtar dosyasıdır. Bu dosyadaki sıfır numaralı anahtar, yönetimsel 3DES anahtarıdır. Ana dosya erişim koşullarında, bu dosyadaki anahtarı kullanan güvenli mesajlaşma protokolü belirtilmiştir. Dolayısıyla kart üzerinde yeni dosyalar oluşturmak için mutlaka buradaki anahtar bilinmelidir. Örnek uygulama geliştirme kartları için bu değer "TEST KEYTEST KEY" karakter dizisindeki her bir harfe karşılık gelen ASCII değeridir.

0100h tanıtıcı değere sahip adanmış dosya, sistem temel dosyalarını içerir ve erişim koşulu olarak 3F01h global seviye anahtar dosyasındaki sıfır numaralı anahtar kullanılarak yapılan güvenli haberleşme protokolü belirtilmiştir.

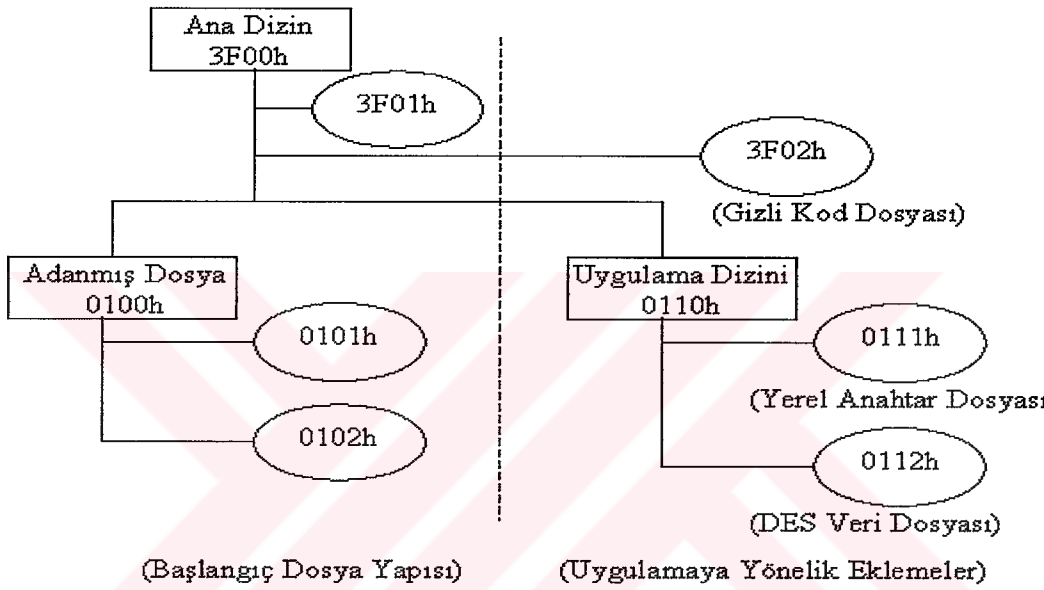
0101h numaralı temel dosya, her kart için farklı olan kart seri numarasını ve 0102h numaralı temel dosya, kart üretici kodunu içermektedir.

5.2 Uygulama Aşamaları

Tezde, akıllı kartın konfigüre edilmesi ve bu kart kullanılarak güvenli bir haberleşme protokolü sağlanma aşamalarından oluşan bir uygulama yapılmıştır.

5.2.1 Uygulama kartı konfigürasyonu

MPCOS işletim sistemine sahip akıllı kart, güvenlik uygulamasında kullanılmak üzere Şekil 5.16'da belirtildiği gibi konfigüre edilmiştir.



Şekil 5.16 Uygulama kartı dosya yapısı

- 3F02h tanıtıcı değere sahip olan temel dosya global seviye gizli kod dosyasıdır ve 0112h numaralı veri dosyasına erişmek için girilmesi gereken gizli kodu içerir.

- 0110h tanıtıcı değere sahip adanmış dosya, güvenli şebeke haberleşmesinde kullanılacak dosyaları içeren adanmış dosyadır.

- 0111h numaralı dosya, doğrulama tipinde 3DES anahtara sahip olan yerel bir anahtar dosyasıdır. Bu anahtar, şebeke haberleşmesi yapan tarafların kimlik doğrulaması işlemleri için kullanılmaktadır.

- 0112h numaralı dosya şifreli şebeke haberleşmesinde kullanılacak olan 8 baytlık DES şifreleme anahtarını içeren veri dosyasıdır. Bu dosyadan okuma

yapmadan önce, 3F02h numaralı gizli kod dosyasındaki gizli kod değeri başarılı bir şekilde girilmelidir.

5.2.2 Uygulamada kullanılan güvenlik protokolleri

Yapılan uygulamada iki çeşit güvenlik protokolü kullanılmıştır. Bunlar, haberleşen tarafların birbirlerinden emin olması için, istedikleri zaman ve sıklıkta uygulayabildikleri kimlik doğrulama ve haberleşmenin izinsiz olarak dinlenmesini engellemek için istenilen anda, iletilen veri üzerinde uygulayabildikleri DES şifreleme mekanizmasıdır.

Kimlik doğrulama mekanizması :

Haberleşme esnasında taraflar birbirlerinin kimliklerinden emin olmak isteyebilir. Karşı tarafı doğrulamak isteyen tarafın, program menüsünden doğrulama seçeneğini seçmesi yeterli olacaktır. Bu durumda sırasıyla şu işlemler yapılır:

- Doğrulama yapan taraf, 8 baytlık rasgele bir sayı üretir ve akıllı karta, bu sayıyı içeren bir dahili doğrulama komutu gönderir. Kart, bu sayıyı ve 0111h tanıtıcı değere sahip yerel anahtar dosyasındaki sıfır numaralı 3DES anahtarını kullanarak 4 baytlık bir doğrulama değeri hesaplar ve döndürür. Program bu değeri kaydeder. 8 baytlık rasgele seçilmiş değer, bir doğrulama isteği çerçevesi içinde karşı tarafa gönderilir.

- Doğruluğunu ispatlaması istenen taraf, ulaşan rasgele seçilmiş değeri alır ve kendi sistemindeki akıllı karta dahili doğrulama komutu gönderir. Akıllı kartın verdiği cevabı karşı tarafa iletir.

- Doğrulama isteğine cevap verilen taraf, gelen doğrulama değeri ile kendi bulduğu değeri karşılaştırır. Değerler uyuşursa karşı tarafın doğruluğundan emin olur.

Değerlerin uyuşması için, her iki akıllı kartın yerel anahtarlarında saklanan 3DES doğrulama anahtarları aynı olmalıdır. Bu mekanizma sayesinde, gizli anahtar tehlikeye atılmadan güvenli bir doğrulama işlemi gerçekleştirilmektedir.

Şifreleme mekanizması :

Taraflar istedikleri anda şifreli haberleşmeye geçebilir. Bunun için program menüsünden DES anahtarını kullanma seçeneği seçilmelidir. Şifrelemede kullanılan DES anahtarı, 0112h numaralı DES veri dosyasında saklanmaktadır. Bu dosyadan okuma yapmanın ön koşulu, 3F02h global gizli kod dosyasındaki sıfır numaralı gizli kodu doğru bir şekilde girmektir. Bu sayede, kartın yetkisiz bir kimsenin eline geçmesi durumunda, DES anahtarının elde edilmesi önlenir. Kriptolu haberleşmeye geçme aşamaları şunlardır:

- İleteceği veriyi şifrelemek isteyen taraf, program menüsünden gizli kod girme seçeneğini seçer. Karşısına gelen mesaj kutusuna gizli kodu girer. Bu kod 3F02h numaralı gizli kod dosyasındaki sıfır numaralı gizli koda karşılık gelmektedir.

- Bu aşamada kullanıcı, program menüsünü kullanarak istediği anda şifreli iletime geçebilir, istediği anda şifresiz iletime geri dönebilir.

- Şifreli ve şifresiz veri, farklı mesaj çerçeveleriyle iletilir. Bu sayede alıcı taraf gelen verinin şifreli olup olmadığını anlar. Gelen şifreli veriyi çözmek için alıcı tarafta da 3F02h numaralı gizli kod dosyasında saklanan gizli kodun girilmesi gerekmektedir.

Gizli kod ve DES anahtarının birlikte kullanılması sayesinde, şifreli mesajları çözmek için hem akıllı karta sahip olmak hem de DES anahtarını koruyan gizli kodu bilmek gerekmektedir.

Geliştirme ortamı :

Akıllı kart kullanan bu güvenlik uygulaması, Windows2000 işletim sisteminde, Microsoft Visual C++ geliştirme ortamında, Windows API fonksiyonları kullanılarak geliştirilmiştir. Uygulamada, Gemplus MPCOS akıllı kartlar ve GemPC410 seri akıllı kart okuyucuları kullanılmıştır.

6. SONUÇLAR ve TARTIŞMA

Bilişim teknolojilerinin gelişimi, bilgiye ulaşma hızını arttırdığı gibi hassas bilginin çalınması riskini de beraberinde getirmiştir. Özellikle, evrensel bilgisayar ağı internet ortamında bulunan organizasyonlar, şebeke güvenliğine gereken önemi vermemeleri durumunda önemli ölçüde zarar görebilmektedir.

İnternette kullanılan iletişim protokolü olan TCP/IP'nin geliştiriliş amacı, aynı amaç için çalışan bir grup arasında etkin veri paylaşımını sağlamaktı. Bu yüzden, geliştirilme aşamasında kötü niyetli kullanıma karşı güvenli olmasına ihtiyaç duyulmayan TCP/IP protokolünün bir çok zayıf noktası bulunmaktadır. Şebekelere yapılan saldırıların birçoğu bu protokolün zayıflıklarını kullanarak gerçekleştirilmektedir. TCP/IP protokolünün zayıf yönleri, aktif şebeke cihazlarında özel konfigürasyonlar yapılarak, kriptografik fonksiyonlar yardımıyla haberleşilen tarafların kimliklerini sık sık doğrulayarak ve iletilen önemli bilgileri şifreleyerek giderilmeye çalışılmaktadır.

Bir sistemde çalışan bütün yazılımlar potansiyel birer saldırı noktasıdır. Gereksiz yere açılan portlar, kontrol edilmeyen parametre giriş uzunlukları, uygulamalara has zayıflıklar, sistemleri saldırganlara karşı daha zayıf hale getirmektedir. Bu yüzden sistem üzerinde çalıştırılacak yazılımlar dikkatle seçilmeli, yazılım üretici firmalar araştırılmalı, kullanılmayan yazılım ve servisler mutlaka kaldırılmalıdır.

Şebeke güvenliğinden sorumlu uzmanlarının bulunmaması ya da bu kişilerin yeterli bilgi ve deneyime sahip olmaması da bir organizasyonu saldırılara karşı zayıf düşürecek önemli açıklardan birisidir. Kurulan sistem sürekli olarak izlenmeli, çalışan yazılımların versiyonları takip edilmeli, bulunan zayıflıklara karşı hızla önlemler alınmalı, güvenlik politikaları güncel tutulmalıdır. Güvenlik bir defaya mahsus olarak yapılıp bitecek bir iş değil, üzerinde önemle durulması gereken bir süreçtir.

İnternet üzerinden gelecek saldırılar çok çeşitli olabileceği gibi, kurumu bunlardan korumak için seçilebilecek yöntemler de çeşitlidir. Bu çalışmada, şebeke güvenliğini sağlamada akıllı kartların kullanılması yöntemi üzerinde durulmuştur.

Üzerlerindeki veriyi güvenli bir şekilde saklayabilmeleri ve kriptografik algoritmaları gerçekleştirebilmeleri açısından akıllı kartlar, şebeke güvenliği konusunda ideal bir uygulama ortamı sunmaktadır. Bu kartlar küçük ve kolayca taşınabilir olduklarından, günlük hayatta herkes tarafından bulundurulabilmekte ve her yerde kullanılabilmektedir.

Bu çalışmada akıllı kartlar kullanılarak bir güvenlik uygulaması yapılmıştır. Yapılan uygulamanın güçlü noktası, doğrulama ve veri şifrelemesinde kullanılan anahtarların hiçbir şekilde üçüncü bir şahıs tarafından ele geçirilememesidir. Bu noktada akıllı kartların veriyi güvenli saklama yeteneği ve uygulama pratiklikliği özelliklerinden faydalanılmıştır.

Uygulamada kullanılan doğrulama mekanizmasında, kendisini doğrulaması istenen tarafa 8 baytlık rasgele bir veri iletilir ve bu sayıya karşılık gelen doğrulama değerini göndermesi beklenir. Doğrulama değerlerini akıllı kart hesapladığından, ilgili 3DES anahtarı sadece kart tarafından bilinmektedir.

Uygulamada kullanılan veri şifreleme mekanizmasında ilgili DES anahtarı yine kartın içindedir. Taraflar şifrelenmiş veriyi çözebilmek için ilgili akıllı karta ihtiyaç duyar. Yani simetrik anahtar şifreleme yönteminin en zayıf yönü olan anahtar alışverişi problemi, akıllı kartlar kullanılarak çözülmüştür. Bu anahtarın bulunduğu dosya gizli bir kod ile korunduğu için, ilgili akıllı kartın yetkisiz bir kimsenin eline geçmesi durumunda bile anahtar ele geçirilemez. 4 baytlık bu gizli kod ard arda 7 defa yanlış girildiği takdirde, erişime izin veren gizli kod dosyası kilitlenir ve DES anahtarına bir daha ulaşılamaz.

Kullanılan akıllı kartların işletim sistemi MPCOS, uygulamanın ihtiyaç duyduğu güvenlik ortamını gizli kod ve anahtar dosyaları ile karşılamıştır. Bu dosya yapıları kullanılarak etkin güvenlik uygulamaları pratik bir şekilde gerçekleştirilebilmektedir.

Sağladığı kolaylıkların yanısıra güvenlik konusunda akıllı kart kullanmanın dezavantajları da bulunmaktadır. Öncelikle böyle bir sistemde her uç noktada bir

akıllı kart ve bir akıllı kart okuyucu bulunmalıdır. Getirdiği ek maliyetin yanısıra kart okuyucu bilgisayarın bir portunu sürekli meşgul edecektir. Ayrıca sisteme eklenen her yeni kullanıcıya bir kart verilmelidir. Geniş bir bölgeye yayılmış şebekeler için kullanıcılara kart ulaştırma, kartı geri alma, değiştirme gibi yönetimsel işlemler de oldukça zor olacaktır. Örneğin bir kullanıcıya çabucak ve kısa süreli sistem giriş izni verilmesi gibi bir işlem pratik olarak gerçekleştirilemeyecektir.

Fakat akıllı kart teknolojisinde yaşanan gelişmeler, onların her gün biraz daha hayatımıza girmesi, donanımlarının ucuzlaması ve uygulama geliştirme ortamlarının geliştirilmesi, akıllı kartların getirdiği dezavantajları zamanla yok edecektir. Akıllı kart okucuyularının ucuzlaması ve giderek temel bir sistem bileşeni haline gelmesiyle, kullanıcılar kendi okuyucularını alabilecek, uygulama geliştirme ortamları sayesinde kendilerine gönderilen yazılımı yine kendi aldığı karta yükleyebilecektir. Bu sayede uçtan uca erişim dezavantajı zamanla ortadan kalkacaktır.

Günümüzde akıllı kartlar pek çok alanda kullanılmakta ve bu alanlar gün geçtikçe artmaktadır. Şebeke güvenliğinden ödemeli televizyon yayınlarına, paralı otoyollardan cep telefonlarına kadar pek çok alanda akıllı kartlar kullanılmaktadır. Birbirleriyle oldukça ilgisiz gözüken bu alanların hepsinin ortak bir noktası bulunmaktadır: kriptografik algoritmalar kullanılarak sağlanan güvenlik.

Anahtar temelli iki çeşit şifreleme algoritması bulunmaktadır. Simetrik algoritmalar az işlem gücü isteyen, oldukça hızlı algoritmalardır. Bununla beraber anahtar yönetim problemleri bulunmaktadır. Asimetrik algoritmalarda anahtar yönetim problemi yoktur fakat bunlar çok işlem gücü isteyen ve yavaş algoritmalardır. Akıllı kartlar, az işlem gücü isteyen ve hızlı olan simetrik anahtar algoritmalarının anahtar yönetim problemini ortadan kaldırmaktadır. Bu özellikleri, bundan 10 yıl öncesine kadar sadece telefonlarda ve güvenli giriş/çıkış sistemlerinde kullanılan akıllı kartların bugün birçok uygulamada karşımıza çıkmasının önemli nedenlerinden biridir.

Tüm bu avantajlarının yanısıra popüler işletim sistemlerinin de akıllı kart ortamlarını desteklemesi, uygulama geliştiricilere arayüz fonksiyonlarını sunması akıllı kartların gelişimlerine önemli katkılar sunmaktadır.

KAYNAKLAR

[1] **Kenneth H. Rosen,** 1995, Cryptography: Theory and Practice, CRC Press

[2] **Michael Welschenbach,** 2001, Cryptography in C and C++, Apress

[3] **W.Rankl, W.Effing,** 2000, Smart Card Handbook, Wiley Computer Publihing

[4] **Zhiqun Chen,** 2000, Java Card Technology for Smart Cards, Addison Wesley

[5] **Java Card Special Interest Group,** 2003, Introduction to Java Card, http://www.javacard.org

[6] **MAOSCO Ltd,** 2003, MULTOS Technology, http://www.multos.com

[7] **Smart Valley,** 2001, Güvenlik, Denetim ve Kontrol Konferansı, http://www.smartwalley.com

[8] **T.J.Klevinsky, Scott Laliberte, Ajay Guppa,** 2002, Hack I.T. – Security Through Penetration Testing, Addison Wesley

[9] **Eric Cole,** 2002, Hackers Beware, New Riders Publishing

[10] **CERT/CC,** 2001, Security Practices Structure, http://www.cert.org

[11] **Fatih Özavcı,** 2002, Saldırı Tespit Sistemleri, www.guvenlikhaber.com

[12] **Türkiye Kriptografi Sayfaları,** 2002, Kriptolojiye Giriş, http://gsu.linux.org.tr

[13] **Henry Dreifus, J.Thomas Monk,** 1998, Smart Cards – A Guide To Building And Managing Smart Card Applications, Wiley Computer Publihing

[14] **Herbert Schildt,** 2003, The Complete Reference Java 2, MC Graw Hill

[15] **ISO-7816,** Part I-IV, http://www.iso.org

[16] Gemplus, 1999, MPCOS-Reference Manual, http://www.gemplus.com

[17] Brial McKeon, 2001, A Comparative Study of the Major Smart Card Platforms, Keycorp Limited

[18] Brial McKeon, 2000, The Fundemental Requirements of Smart Cards, Keycorp Limited

[19] Samuel Chanson, 2003, Guide to Smart Card Technology, http://www.cyber.ust.hk

[20] TÜBİTAK – BİLTEN Elektronik Kimlik Hizmetleri, 2002, Genel Kriptografi, http://e-kimlik.bilten.metu.edu.tr

ÖZGEÇMİŞ

Tarık TAKTAKÇI, 1975 yılında İstanbul'da doğdu. 1993 yılında İstanbul Polis Koleji'nden mezun oldu. Aynı yıl Ankara Üniversitesi Elektronik Mühendisliği Bölümü'ne girdi. 1997 yılında Elektronik Mühendisi ünvanını aldı. 1998 yılında İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Yüksek Lisans Programı'na başladı. Eylül 1997 – Aralık 1999 yılları arasında İstanbul Emniyet Müdürlüğü Bilgi İşlem Şube Müdürlüğü'nde, Ocak 2000 – Mayıs 2001 yılları arasında Alcatel Teletaş Arge Departmanı'nda, Kasım 2001 – Ekim 2002 yılları arasında Milli Savunma Bakanlığı Arge Kışlası'nda (askerlik hizmeti), Kasım 2002 – Temmuz 2003 tarihleri arasında Neta Elektronik AŞ'de, Ağustos 2003 – Ekim 2003 tarihleri arasında Elmak Elektronik AŞ'de çalışmış olup Kasım 2003 tarihinden itibaren ST Microelectronics SetTopBox Departmanı'nda görev yapmaktadır.