AKILLI KARTLAR İÇİN GÜVENLİK SİSTEMİ
(Elektronik Kimlik ve Pasaport Sistemi)

Giriş

Günümüzde teknolojik gelişmeye paralel olarak akıllı sistemler ve bu sistmlerde daha fazla güvenlik, daha fazla hız önemli olmaya başlamıştır. Özellikle ***ÖdeGeç*** (para işlemlerinde, akıllı bilet v.b.) veya ***OnaylaGeç*** (güvenlik ve geçiş sistemleri, Pasaport, giriş kartı v.b.) olarak adlandıracağımız sistemler basit bir kart ile gerçekleşmektedir. Bu kartlarda istenen özellikler;

- Hızlı tepki süresi,
- Çok fazla güvenlik,
- Olabildiğince fazla veri depolayan bellek.
- Ucuz maliyet.

Yukarıdaki dört kriter göz önüne alındığında günümüzde kişinin sahip olduğu kimlik (kendini ispat), para ve özel bilgileri taşıyan ancak kişinin kendisi tarafından kullanılabilen bir araca ihtiyaç vardır. Bu araç son zamanlarda hayatımıza daha fazla giren akıllı kartlar veya onların türevleridir. Bu kartlar çok az yer işgal ettiğinden kolay taşınır ancak kolay da kaybedilirler. Bu nedenle güvenlik kartlarla birlikte kartların ilişkili olduğu merkezi birimlerde sağlanmaya çalışılmaktadır. Bu durumda da kişinin güvenliği kendi elinde taşıdığı kart kadar merkezi birimlere ve onların üzerinde çalışan yazılımlara bırakılmıştır.

Kart ve Güvenlik

Günümüzde bilgi işlem hırsızlığının yazılımlar üzerinden yapıldığı düşünülürse güvenlik mekanizmasının büyük çoğunluğunun merkezi sistemlere bırakılması da iyi bir düşünce gibi görünmemektedir. Buna karşılık güvenlik doğrudan kişinin taşıdığı kartlarda da olması kartların güvenliğini gündeme getirmektedir. Bilindiği gibi kartların elde edilmesi kolay ve ucuzdur. Hatta son bir kaç yıl içerisinde kartlar ile ilgili taklit edilme veya buna benzer yöntemler kullanılarak güvenliğin zedelenmesi kartların hayatımızdaki yerini tartışmaya açmıştır. Doğal olarak üretim maliyetinin düşüklüğü güvenliği de en aza indirmektedir. Gelişmeleri kısaca hatırlayacak olursak; ilk kartlar sadece üzerindeki görsel şekiller ve barındırdığı kabartma rakamlar ile sağlanırdı, bunların taklidi oldukça kolay ve üretimi son derece basitti. Daha sonra özellikle kredi kartlarının yaygınlaşması ile birlikte ek bir güvenlik olarak manyetik şerit ile güvenlik sağlanmaya çalışıldı. Bu yöntemde de bilgi sabit veri (*Static Data*) olarak manyetik şerite yazılıyordu. Bu yöntem de sonuçta teknolojiyi bilenler tarafından kolayca taklit edilebilmekte ve güvenlik sorunu yaratabilmekteydi (örneğin papağan denen bir cihaz ile kredi kartları kopyalanıp bir çok kopyası oluşturulabilmekteydi). Maliyetin bir miktar arttırılmasına karşılık güvenliğin çok daha fazla artması düşüncesi ile akıllı kartlar ortaya çıktı ve en önemli uygulama alanı olarak kredi kartlarında Chip&PIN olarak adlandırılan akıllı yonga barındıran kartlar üretildi.

Bu kartların taklit edilmesi hemen hemen imkansız olmakla birlikte kaybedilmesi halinde uygulamaların PIN (*Personal Identifier Number*) olarak adlandırılan kişiye ait bir numaranın uygulamalarda sorgulanması ile güvenlik sağlanması fikri ortaya çıktı. Manyetik şeritli uygulamda da bu bilgi bulunmasına karşılık bilgi manyetik şerit üzerinde açık bir şekilde tutulduğundan kopyalanması oldukça kolaydı. Ancak yonga barındırıan kartlarda bu bilgi kart içerisindeki yongada şifreli olarak saklandığından ve yonganın içerisinden bu bilgiyi elde etmek imkansız olmakla birlikte PIN doğrulaması yonga tarafından yapıldığından (akıllı kart denmesinin de en önemli nedeni de budur) ihtiyaç duyulan güvenlik sağlanmış oldu.

Akıllı Kartlarda Maliyet, Güvenlik ve Kullanım kolaylığı

Yukarıdaki bölümde kartların daha güvenli olması düşüncesi ve ortaya çıkan elektronik (yonga barındıran) kartların yaygınlaşmasının önündeki en önemli engelin kullanım zorluğu ve maliyet yüksekliğidir. Çünkü bazı uygulamalar maliyet, bazı uygulamalar kullanım kolaylığı bazı uygulamalar iyi daha fazla güvenliğe ihtiyaç duyar. Bu nedenle yapılan uygulamaya bağlı olarak değişik kart tipleri ortaya çıkmaktadır. Kullanım kolaylığı kart sahibinin kendisi ile iletişime geçerek bir sorulama yapmayan kartların üretilmesini gündeme getirmiş ve Philips firması Mifare olarak adlandırılan içerisinde simetrik şifreleme tekniğini de kullanarak temas uçları bulunmayan bir yonga tasarlamıştır. Bu yonga, üzerinde basit bir işletim sistemi ve basit bir simetrik şifreleme tekniğini barındırmaktaydı. Ancak bu yonga işletim sisteminin DES simetrik şifreleme yöntemini kullanmasından dolayı ortaya kullanımının kolay olmasına karşılık kolay kırılabilmesi sorununu ortaya çıkardı. Bundan dolayı güvenlik bir kez daha tartışmaya açıldı.

Maliyetin güvenlikle ters orantılı olduğunu düşünürsek en önemli gelişme kullanım kolaylığı ve güvenlik içeren bir yonganın tasarlanması olacaktır. Burada maliyet olarak kabul edilebilir bir maliyet ancak kullanımı kolay ve güvenliği oldukça yüksek bir işletim sistminin tasarlanması planlanmaktadır. Kullanım kolaylığı olarak temassız bir kartın (veya dual interface) işletim sisteminin tasarlanması planlanmaktadır.

Bu tip bir yonga işletim sistemi ile pasaport veya benzeri uygulamaları da daha kolay ve güvenli çalıştıracak bir işletim tasarlanmış olacaktır. Günümüzde kullanılmakta olan standartlara uyumlu, var olan uygulamalara da destek verebilen ancak bazı ek komut ve yöntemlerle dış uygulamalara da ek güvenlik önlemleri getirmiş olan bir işletim sistemi tasarlanmış olacaktır. Üzerinde bu işletim sistemini barındıran yongalar aynı zamanda atanmış sistemler için de bir çeşit TPM (Trusted Platform Modul) işevi görebilecektir.
.....

Yukarıdaki felsefeden yola çıkarak atanmış sistemleri bir araç ve kaynak olarak kullanacak kişilerin ihtiyaç duydukları kaynak erişimlerini kolaylaştırmak ve sistem üzerinde kolay uygulama geliştirmelerine yardımcı olmak için atanmış sistemlerin seri veri yollarını giriş/çıkış arayüzü olarak kullanan ve EPROM, RAM ve I/O modullerinin kullanımını kolaylaştıran üzerinde koşan uygulamaların bağımsız olarak çalışmasına olanak sağlayan temel bir işletim sistemi önerisidir.

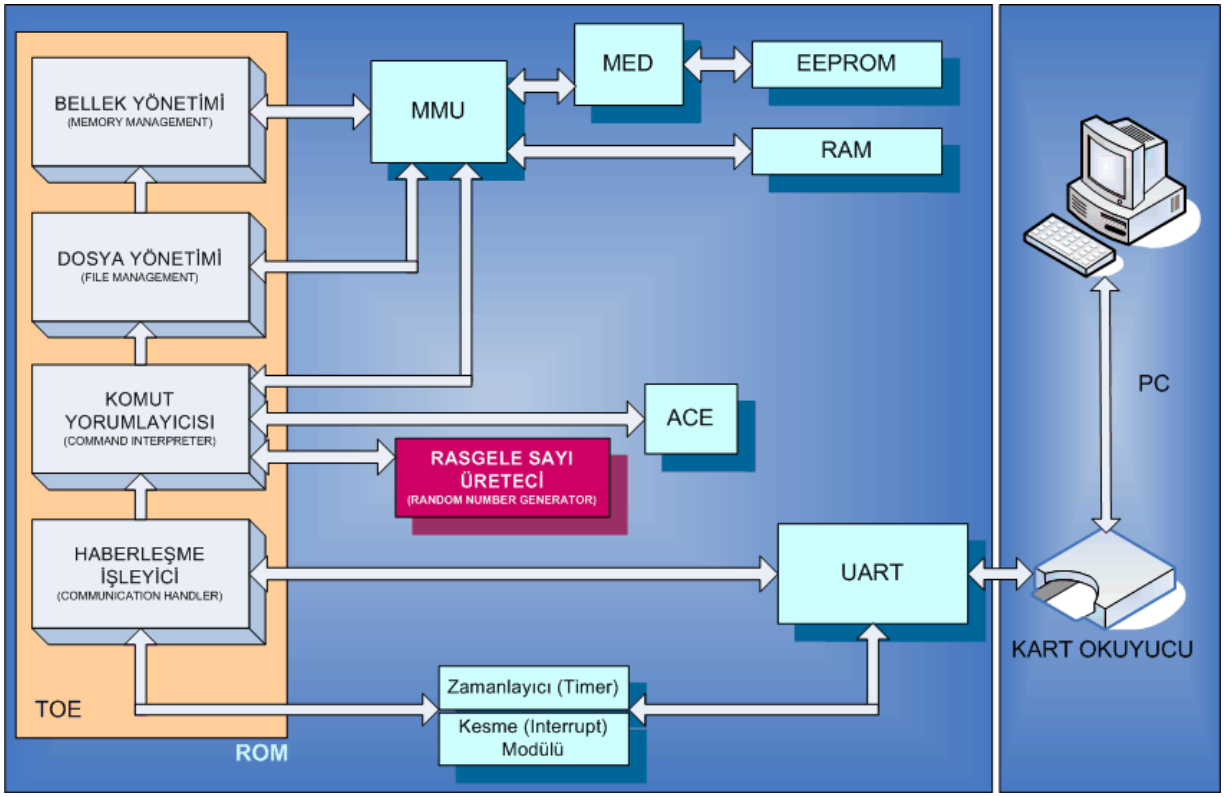
Bu işletim sisteminde uygulama geliştirici uygulamalarına ait verileri (açık bilgileri, gizli bilgileri ve anahtarları) TPM'in (Akıllı Kartın) EEPROM alanına yükler ve

kendisine ait güvenlik işlevini çalıştırarak güvenliğini sağlamış olur. Bu tip kartın en temel özelliği ilklendirme sırasında sisteme ait bazı güvenlik verilerinin yanısıra güvenlik işlevinin de kart üzerine yüklenebilir olmasıdır. Böylece işletim sistemi hem gizli veriler ve anahtarlar üzerinden hem de uygulamacı tarafından yüklenen güvenlik işlevi ile güvenliği sağlamış olur. Buradaki yenilik yüklenebilen güvenlik işlevine sahip bir mekanizmanın olması. Böylece yukarıdaki bölümlerde kartların güvenliği ile ilgili gelişim süreci anlatılırken,

- Fiziksel güvenlik (kart basımı, barkod v.b.)
- Manyetik ortamda sabit veri ile sağlanan güvenlik,
- Akıllı kartlarda yonga üzerindeki sabit veriler ve anahtarlar aracılığıyla getirilen güvenlik (SDA ve DDA kartlarda veriler kullanılmakta). *Karşılaştırma ile sağlanan güvenlik*
- Güvenlik işlevi yüklenerek sağlanan *Değiştirilebilir, Koşturulabilir güvenlik*.

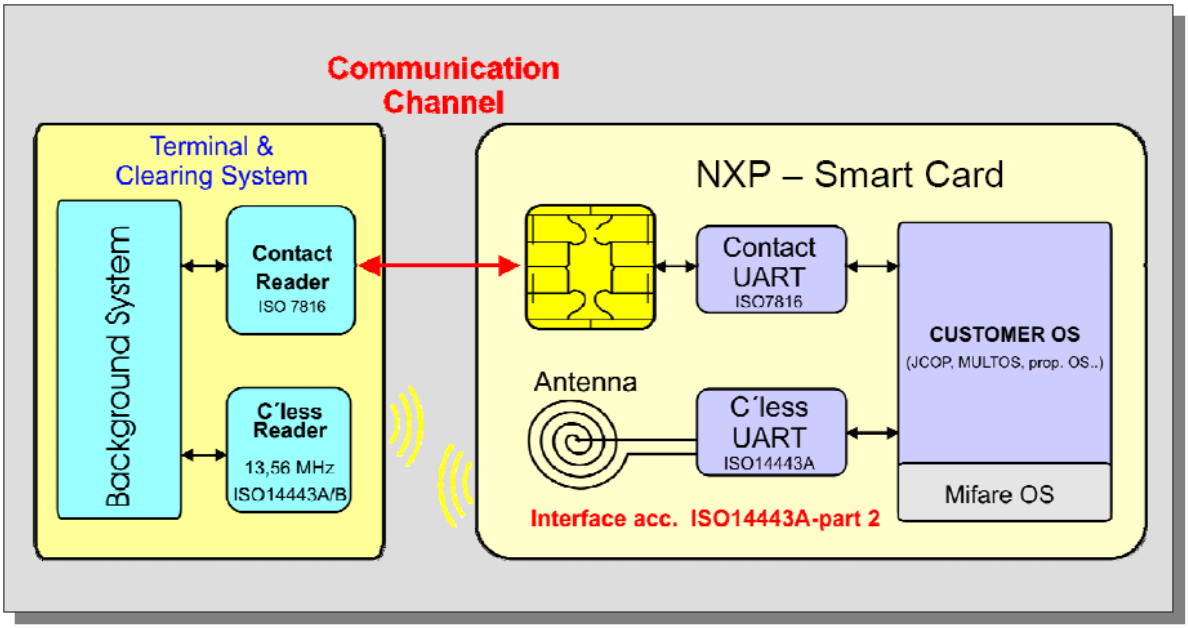
Yukarıdaki maddeler bir önceki maddeyi kapsayarak ilerlemektedir. Böylece maliyet, güvenlik ve kullanım kolaylığı arasındaki tüm analizler yukarıdaki maddelerde tanımlı durumları kapsayacak şekilde yapılabilir. Ancak bu çalışmada birinci ve ikinci madde devre dışıdır çünkü en temel ihtiyaç olaran güvenliği içermez.

Tasarlancak atanmış, gömülü yazılımın ilişkili olduğu donanım birimleri aşağıda gösterilmiştir.

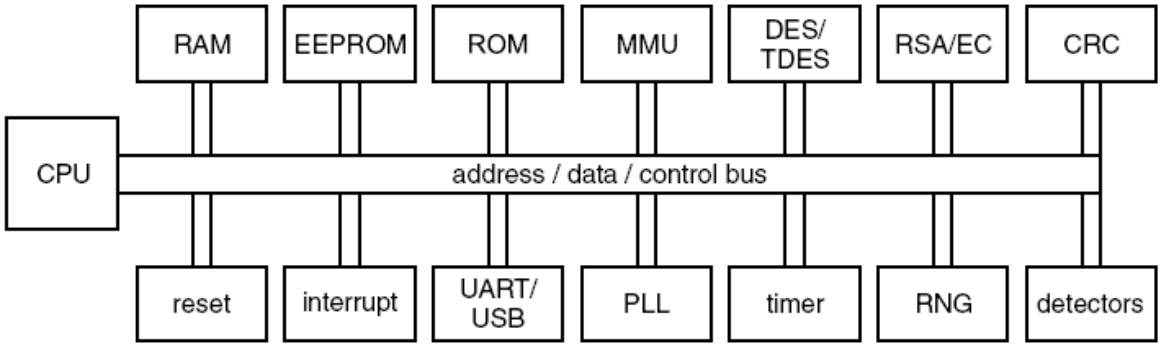


Şekil-1 Tasarlanacak yazılım ve ilişkili olduğu donanım birimleri

Yukarıda Akıllı Kart içerisindeki donanım birimleri ile ilişkili yazılım modülleri gösterilmiştir. Aşağıdaki şekillerde ise doğrudan donanım modülleri ve bu modüller ile ilişkili işletim sistemi (Operating System) bağlantısı gösterilmektedir.



Şekil-2 Bir Akıllı Kartın dış ve iç bileşenleri



Şekil-3 Bir Akıllı Kartın iç bileşenleri

Yukarıdaki şekilden de görüleceği üzere geliştirilecek sistemin üzerinde koştuğu donanım biriminde aşağıdaki birimler bulunmak zorundadır;

- **CPU :** Merkezi İşlem Birimi, kendisine elektriksel olarak bağlı olan çevre birimleri ile birlikte Akıllı Kartın akıllı olmasına neden olan yazılımı işleten birimdir. Bu birim bazı yazılım komutlarını ROM ve EEPROM bellekten alır ve işler. Elde ettiği sonucu yine çevre birimleri aracılığıyla ya saklar yada kullanıcının programı doğrultusunda dış dünyaya aktarır. Akıllı kartlarda kullanılan Merkezi İşlem Birimi ilk aşamalarda 8 bit daha sonraki zamanlarda işlem gücü artarak 16 ve 32 bit olarak tasarlanmışlardır. Daha kısa olarak gerek işletim sistemi gerekse uygulama yazılımlarının üzerinde koştuğu işlemci birimi.
- **ROM :** ROM bellek, genellikle Merkezi İşlem Biriminin yazılım komutlarını ve Akıllı Kartın İşletim sistemini üzerinde barındırır. Yazılım kodları Yonga üretilmesi aşamasında ROM bellek üzerine kazınır. İşletim Sisteminin ve uygulamanın ihtiyaç duyduğu kütüphanelerin de bulunduğu donanım birimi.
- **RAM :** Merkezi İşlem Birimi işlemlerini gerçekleştirirken geçici olarak ihtiyaç duyduğu bilgiler için yonganın RAM belleğini kullanır. Böylece geçici işlemler için kullanılan bellek Akıllı kartın RAM belleğidir. Diğer bir deyişle, programların geçici olarak kullandıkları ve değişkenlerin bulunduğu birim.
- **EERPOM :** EEPROM bellek Akıllı kartın kalıcı bilgi saklama belleğidir ve kart sahibinin açık ve gizli bilgileri bu bellek içerisinde saklanır. Akıllı kart elektriksel olarak beslenmese bile bu bellekteki bilgiler kalıcı olarak tutulmaktadır. Örneğin kart sahibinin adı kartın içerisinde açık olarak yer almakta ve asla silinmemektedir. Bu bilgi işletim sistemi aracılığıyla dış dünyadan alınır ve bir daha değiştirilmemek veya üzerine yazılmamak üzere EEPROM bellek alanında sabitlenir. EEPROM bellek boyu ilk aşamalarda 4Kbyte sonraları ihtiyaçlar doğrultusunda 8K, 16K, 32K ve 64K olarak arttırılmıştır. Kart sahibinin (pasaport sahibinin) verilerinin üzerinde tutulduğu kalıcı bellek. Özellikle güvenliğin sağlanması gereken birimdir.
- **ACE :** Şifreleme makinesi (Crypto Engine) olarak adlandırılan birimdir. Bu birimde daha çok matematiksel işlemler ve şifreleme/şifre çözme işlemleri gerçeklenmektedir. Akıllı kart yongası üzerinde yer alan ve Merkezi İşlem Biriminin ihtiyaç duyduğu matematiksel işlemler ile şifreleme/deşifreleme ile sayısal imza işlevlerinin tek başına gerçekleştiren birim "Kripto yardımcı işlemcisi" dir. Matematiksel işlemler genellikle çok daha karmaşık ve zaman alıcı işlemler olduğundan Merkezi İşlem Biriminin işlem yükünü paylaşması amacıyla Akıllı Kart yongasının içerisinde Kripto yardımcı işlemcisi bulunmaktadır.
- **UART :** Asenkron seri bilgi alışverişinin gerçeklendiği donanım modülü.
- **MMU :** Bellek yönetiminin kullandığı donanımsal firewalların işletildiği donanımsal birim. Yazılacak uygulama ve yazılımlar bu donanım öğesini kullanacaktır.

Bu işletim sisteminde aşağıdaki yazılım modülleri bulunacaktır;

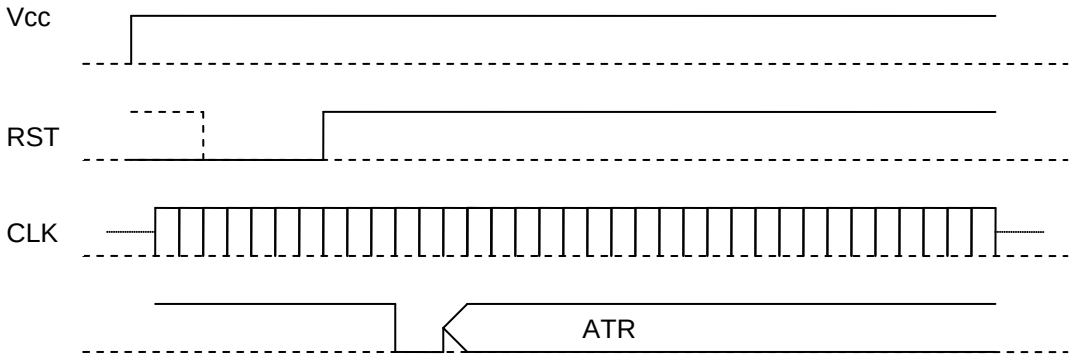
- EEPROM üzerinde etkili bir Bellek yönetim modülü (Memory Manager),
- EEPROM üzerinde etkili bir Dosya Yönetim modülü (File Manager),
- Seri bilgi giriş çıkışlarını denetleyen bir SGÇ(SIO) yönetim modülü (Comm. Handler ve Command Interpreter),
- Zamanlayıcı denetim modülü (Timer Manager),
- Güvenlik yönetim modülü
- İlklendirme ve kişiselleştirme yönetim modülü,

Akıllı Kart Veri İletişimi

Akıllı Kartlar dış dünya ile Şekil-2 de gösterilen CLK (*clock*) ve I/O (*input/output*) olarak belirtilen iki fiziksel uç üzerinden iletişim kurmaktadır. CLK ucu, akılllı kartın üzerinde takılı olduğu kart okuyucu tarafından yonga içerisindeki birimlerin ihtiyaç duyduğu saat darbelerinin verildiği uçtur. I/O ucu ise çift yönlü bir uç olup bazen yongadan kart okuyucuya bazen de kart okuyucudan karta bilgi taşımada kullanılır. Akıllı kartla kart okuyucu birimi arasındaki iletişim için yonga içerisinde UART olarak adlandırılan *"asenkron alıcı verici birimi"* bulunmaktadır. Akıllı Kart yongasının seri iletişim arayüzü olan UART'ı yonga içerisinde bulunan akıllı kart işletim sistemi denetlemektedir. Bu arayüzdeki iletişim hızı kart okuyucunun verdiği 3.5MHz saat darbesine bağlı olmakla birlikte yazılımsal olarak en fazla 115 kbit/s'a kadar çıkabilmektedir. Daha fazla hız istenmesi durumunda ya UART'ın CLK hızının arttırılması yada PLL olarak adlandırılan saat işaretinin çarpma devresi ile hızın arttırılması gerekir. Ancak günümüzdeki yonga ve işletim sistemleri ile ~380 kbit/s hızına ulaşılmıştır.

Akıllı kartın veri iletişimi donanımsal olarak UART birimi üzerinden CLK, I/O uçları ile gerçeklenmesine karşın kart okuyucu ile el sıkışmadan birbirleri ile anlaşması olası değildir. El sıkışma işleminin ilk adımı yongada bulunan RST (*reset*) ucu ile sağlanmaktadır. Aşağıdaki şekilde gösterildiği gibi kart okuyucu ilk olarak Vcc ucundan yongaya enerji verir ve yonganın RST ucunu belirli bir süre 0V potansiyel seviyesinde tutar. Daha sonra RST ucunu tekrar 5V gerilim seviyesine yükseltir. Bu işlem yongadaki işlemcinin durum makinasını sıfırlar ve işlemcinin koşturduğu işletim sisteminin ilk duruma gelmesine neden olur.

El sıkışma işleminin ikinci adımı yonga ve işletim sisteminin doğru koşullandırılıp kart okuyucu ile ilk mantıksal temasın gerçeklenmesidir. Yukarıda açıklanan donanımsal işlemden sonra işletim sistemi, yonganın doğru başlatıldığını ve beraberinde kendi çalışma koşullarını da içeren bilgileri ATR (Answer To Reset) verileri olarak kart okuyucuya gönderir (ilk mantıksal temas).



Şekil – 4 Akıllı Kart Fiziksel arayüz uçları

ATR bilgisi akıllı kartlar için temel bir işlemdir ve olmazsa olmaz bir bilgidir. ATR verileri beş ayrı gruba ayrılmıştır.

- Başlangıç Karakteri (TS),
- Format Karakteri (T0),
- Arayüz Karakterleri (TA, TB, TC, TD, v.b),
- Tarihsel Karakterler (T1... TK),
- Kontrol Karakteri (TCK)

Yukarıdaki verilerden TS, T0 ve TCK zorunlu bilgiler diğerleri seçimli bilgilerdir. Arayüz karakterleri ve Tarihsel karakterler akıllı kartın çalışma hızını, parametre değiştirme olanağını, üretici ile ilgili bilgileri ve iletişim sorunlarını çözmeye yönelik bilgiler içerir.

ATR içerisindeki TS başlangıç ve kodlama yöntemini, T0 karakteri ise iletişimin formatını belirler. TS karakteri AKiS gibi birçok işletim sisteminde 0x3B (Doğrudan Kural) olarak tanımlanmıştır. T0 iletişim formatını belirlediğinden bu sekizlide bulunan en anlamlı dört bit (b8, b7, b6, b4) ATR'de TA(1), TB(1), TC(1) ve TD(1) sekizlilerinin olup olmadığını, geri kalan dört bit ise tarihsel verilerin sayısını belirtir. Böylece tarihsel karaterlerin de en fazla 15 tane sekizliden oluşacağı ortaya çıkmış olur.

ATR içerisinde bulunan karakterler ve bunlarla ilgili daha ayrıntılı bilgiler ISO 7813-3 dokümanında açıklanmıştır.

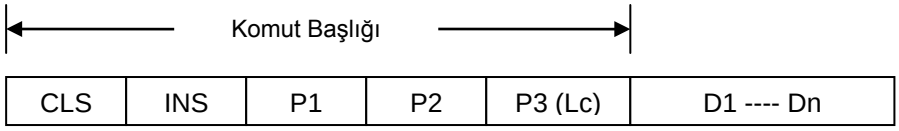
Akıllı Kart İletişim Protokolleri

Bir önceki başlık altında kart okuyucu tarafından akıllı kartın RTS ucuna gönderilen başlama işareti (reset darbesi) sonrasında akıllı kartın içerisinde bulunan mikroişlemci ve diğer devreler ile birlikte yonganın işletim sisteminin başlama koşullarına getirilip kart okuyucuya ATR bilgisi göndererek ilk kalp atışını verdiğini açıklamıştık. Bu aşamadan sonra yonga ile kart okuyucu aralarında bir iletişim protokolü ile bilgi alışverişinde bulunmaya başlar. Bu iletişim protokolü ATR'de bulunan bilgi doğrultusunda ya T=0 veya T=1 protokolüdür. Başkaca iletişim protokolleri olsa bile bunlar daha özel iletişim protokolleri olarak işlem görmektedir. Bu nedenle biz burada T=0 ve T=1 protokollerini açıklamaya çalışacağız.

T=0 İletişim Protokolü

Akıllı kartlarda T=0 iletişim protokolü, seri asenkron yarı çift yönlü karakter iletişim protokolüdür. Akıllı kartlarda reset ve ATR işleminin başarılı bir şekilde gerçekleşmesinden sonra ilk işlem olarak gönderilecek sekizlilerin sekteye uğraması veya veri gecikmesinin algılanması için ATR'de tanımlı özel bir sekizli olan TC(2) arayüz sekizlisi içerisinde belirtilen karakter bekleme zamanı parametresi ile protokolde bekleme zamanı hesaplanır. Hesaplama işlemi sonrasında arayüz birimi komut gönderek iletişimi başlatır. Akıllı kart ise gelen komuta komutta istenen verileri ve bu veriler ile birlikte standartlarda tanımlı sonuç durum bilgisini geri döndürerek cevap verir. T=0 protoklünde komut yapısı aşağıdaki şekilde gösterildiği gibi komut başlığı ve komut verisinden oluşur.

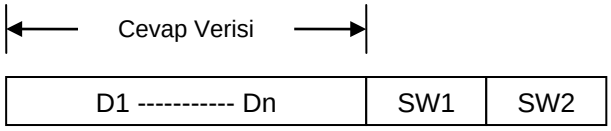
T=0 protokolünde komut başlığı, ilk sekizli CLS (class) olarak adlandırılan sınıf sekizlisi, daha sonra gelen INS (instruction) olarak adlandırlan Komut sekizlisi ve bunlardan sonra gelen P1 ve P2 olarak adlandırılan iki tane parametre sekizlileri ve son olarak P3 veya komut veri boyu olarak adlandırılan Lc sekizlisinden oluşur. Eğer Lc = 0 ise arayüz birimi tarafından gönderilecek herhangi bir verinin olmadığı anlamı taşır. Buna karşılık Lc ≠ 0 ise boy kadar bilgi akıllı kart yongasına iletilir.



T=0 protokolü komut veri yapısı

Şekil – 5 T=0 Protokolü

Akıllı kart yongası yukarıdaki şekilde gönderilen komutu aldıktan sonra cevap olarak eğer geri döndüreceği bir veri varsa bu veri ve veri ile birlikte en son iki sekizliden oluşan durum bilgisini de arayüz birimine iletir.



Şekil – 6 Cevap Veri Yapısı

SW1 ve SW2 durum sekizlileri prosedür sekizlisi olarak 3 tiptir;

1. NULL (0x60) karakter.
2. ACK sekizlileri, bunlar veri aktarımını denetlemekte ve Vpp durumu hakkında bilgi vermekte kullanılır.
3. Durum sekizlileri, bunlardan SW1, 0x6X ve 0x9X değeri alır (0x60 hariç). SW2 üzerinde herhangi bir kısıt yoktur.

T=1 İletişim Protokolü

Akıllı kartlarda T=1 iletişim protokolü, asenkron yarı çift yönlü blok iletişim protokolüdür. Bu protokolde karta özel denetimler ile birlikte veri aktarım akış denetimi, zincir blok mesaj aktarımı ve hata denetimi tanımlanmaktadır. Protokol ATR işleminden sonra veya başarılı bir PPS işleminden sonra (Parameter Selection Process) başlar. Protokolün temel karakteristiği;

- Protokol, arayüz cihazının gönderdiği ilk blok ile başlar, blok gönderme hakkını değiştirerek devam eder.
- Bir blok, aktarılan en küçük veri birimidir. Bir blok protokolden bağımsız uygulama verisini veya aktarım hatalarını kotaran aktarım denetim verilerini içerir.
- Blok yapısı taşınan verinin işlenmesinden önce alınan bloğun denetlenmesini sağlar.

T=1 protokolünde karakter çerçevesi daha önce açıklandığı üzere ATR verileri içerisinde tanımlanır. Blok çerçevesinin yapısı ise aşağıdaki şekilde gösterilmiştir.

Prolog Alanı			Bilgi Alanı	Epilog Alanı
NAD	PCB	LEN	INF	EDC (LRC, CRC)

Blok Çerçeve Yapısı

Şekil – 7 Çerçeve Yapısı

Bir blok çerçevesinde üç değişik alan mevcuttur;

- Prolog alanı; T=1 protokolünde bu alanın bulunması zorunludur ve *“Node ADres”*, *“Protocol Control Byte”* ve *“LENgth”* olmak üzere üç sekizli uzunluğundadır.
- Bilgi alanı bulunması zorunlu bir alan değildir ve varsa en fazla 254 sekizli uzunluğundadır.
- Epilog alanı; T=1 protokolünde bulunması zorunlu bir alandır hata denetimi tipine göre (LRC veya CRC) bir veya iki sekizli uzunluğundadır.

T=1 protokolünde üç değişik tip blok tanımlanmıştır;

- *Information Blok (I-blok).* Bu blok uygulama katmanları arasında veri taşır.
- *Receive Ready Blok (R-blok).* Bu blok pozitif veya negatif alındı bilgisi taşır.
- *Supervisory Blok (S-blok).* Bu blok ile kart ve arayüz cihazı arasında denetim bilgisi taşınır.

Bir blok çerçevesindeki Prolog alanında ilk sekizli NAD bilgisidir. NAD bilgisi ile mesajın kaynağı ve hedeflenen adres bilgisi tanımlanır. Bu sekizlideki b1’den b3’e kadar olan bitler de kaynak adresi (SAD), b5’ten b7’ye kadar olan bitler de ise hedef adres (DAD) tanımlanır. Eğer adresleme kullanılmayacaksa bu alan ‘00’ olarak doldurulur.

Bir blok çerçevesindeki Prolog alanında ikinci sekizli PCB’dir. PCB *(Protocol Control Byte)* olarak adlandırılan sekizlide iletişimin denetimine ilişkin bilgiler taşınır. PCB blok tipine göre I-blok, R-blok veya S-blok olarak tanımlanmıştır. Aşağıdaki şekillerde bu blokların kodlamaları ayrıntılı olarak gösterilmiştir.

- Herbir I-blok için PCB'nin b8 biti "0" olarak ayarlanır.

0	b7	b6	b5	b4	b3	b2	b1

b8 ..... 0 (**PCB I-blok**)

b7 ..... Sequence Number, N(S)

b6 ..... More Data Bit, M-bit

b5b4b3b2b1 ..... RFU

Şekil – 8 I-Blok

- Herbir R-blok için PCB'nin b8 ve b7 bitleri "1" ve "0" olarak ayarlanır.

1	0	b6	b5	b4	b3	b2	b1

b8b7 ..... 10 (**PCB R-blok**)

b6b5b4b3b2b1

0-N(r)-0000 ..... Hatasız

0-N(r)-0001 ..... EDC veya eşlik hatası

0-N(r)-0010 ..... Diğer Hatalar

Şekil – 9 R-Blok

- Herbir S-blok için PCB'nin b8 ve b7 bitleri "1" ve "1" olarak ayarlanır.

1	1	b6	b5	b4	b3	b2	b1

b8b7 ..... 11 (**PCB S-blok**)

b6b5b4b3b2b1 ..... b6 response bit

00000 ..... RESYNCH request

10000 ..... RESYNCH response

00001 ..... IFS request

10001 ..... IFS response

00010 ..... ABORT request

10010 ..... ABORT response

00011 ..... WTX request

10011 ..... WTX response

Şekil – 10 S-Blok

Çerçevedeki LEN sekizlisi, bilgi alanında bulunan bilgi uzunluğunu gösterir. Eğer bu sekizli ‘00’ ise çerçevede bilgi bulunmamaktadır. ‘FF’ haricindeki 1-254 arasındaki değer ise çerçevedeki veri uzunluğunu göstermektedir. ‘FF’ değeri ise daha sonra kullanılmak üzere ayrılmıştır.

Çerçevedeki INF alanı, R-blok çerçevesinde hiçbir bilgi bulundurmaz. I-blok çerçevelerinde uygulama verisini, S-blok çerçevesinde ise uygulama verisi olmayan iletişim yönetimine ilişkin veriler barındırır.

Blok çerçevesinin epilog alanında ise blok hata algılama kodu olarak EDC bilgisi taşınır. Bu alandaki bilgi kullanılan yönteme bağlı olarak ya LRC veya CRC bilgisidir. EDC bilgisi, LRC *(Longitudinal Redundancy Check)* yönteminde bir tek sekizli, değeri ISO/IEC 3309’da tanımlanmış CRC *(Cyclic Redundancy Check)* yönteminde iki sekizliden oluşur.

T=1 protokolünde hata algılaması T=0 protokolüne göre daha iyidir. Bunun için T=1 protokolünde karakter bekleme zamanı, blok bekleme zamanı, hata kotarıcı değişik blok yapıları gibi bazı ek parametreler tanımlanmıştır. Bu parametreler kart tarafından ATR içerisinde tanımlandığı gibi arayüz cihazında da tanımlanabilmektedir.

T=0 ve T=1 protokolüne ilişkin daha ayrıntılı bilgiler ISO 7816-3 dokümanında verilmiştir.

T=0 karakter tabanlı, T=1 protokolü ise çerçeve tabanlı bir protokoldür. .

Terminal - Akıllı Kart İletişim Arayüzü

Bir önceki bölümde Akıllı kart ile dış birimlerin iletişiminde kullanılan fiziksel arayüz protokolleri açıklandı. Bu bölümde fiziksel arayüz üzerindeki protokoller kullanılarak iletilen mesajların yorumlanması üzerinde durulacaktır.

Gerek T=0, gerekse T=1 protokolünde dış birimler akıllı kartlar ile ISO7816’da APDU olarak tanımlı komutlar ile iletişim kurmaktadır.

Uygulama Protokol Verileri (Application Protocol Data Unit, APDU), terminal ile akıllı kart arasında veri aktarımında kullanılan bir yapıdır ve OSI modelinin katmanlı yapısında 7. katman olan uygulama veri katmanındaki veri birimi olarak tanımlanmıştır. Ancak akıllı kartlarda bu katman doğrudan veri transfer katmanının hemen üzerinde yer almaktadır. Bu katmandaki veri yapısı ise TPDU (transmission Protokol Data Unit) olarak adlandırılmıştır.

APDU komut veri yapısı

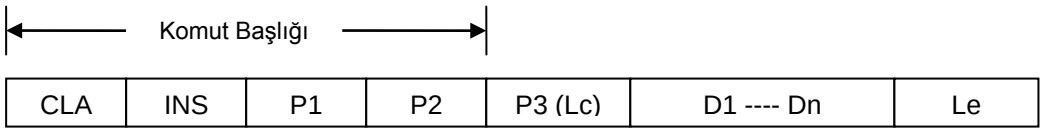
APDU komut veri yapısı başlık (header) ve gövde (body) olmak üzere iki bölümden oluşur. APDU komut başlığı sabit olmakla birlikte komut gövdesi değişken uzunlukta olabilir veya hiç olmayabilir. Başlık CLA, INS, P1 ve P2 olmak üzere dört veri elemanından oluşur. CLA sekizilisi uygulamayı ve uygulamaya ait özel komut kümesini tanımlar. Örneğin ‘A0’ GSM’e özgü komutları, ‘8X’ kuruluşa özgü (özel) komutlar için ayrılmıştır. ISO tabanlı komutlar için CLA sekizlisi ‘0X’ olarak ayrılmıştır. Bunlara ek olarak standartta CLA sekizlisi güvenli iletişim ve mantıksal kanallar oluşturmada da kullanılmaktadır.

Komuttaki bir sonraki sekizli komut (instruction, INS) sekizlisidir ve komut işlevini tutar.

Daha sonraki P1 ve P2 sekizlileri APDU komutu ile daha fazla bilgi taşımak için oluşturulmuş sekizlilerdir. Bu sekizliler ile komutta daha fazla seçenek oluşturmak mümkün olmaktadır.

Örneğin SELECT FILE komutu ile değişik seçenekler oluşturmak mümkün olabileceği gibi READ BINRY komutu ile okunacak alanın offset bilgisi P1 ve P2 ile gönderilmektedir.

APDU yapısı aşağıdaki şekilde gösterildiği gibidir.



Şekil – 5 APDU yapısı

Bu yapıda CLA, Sınıf bilgisi, INS komut bilgisi, P1 ve P2 parametre bilgisidir ve komut başlığını oluşturur. P3 veya Lc ise komut uzunluğu bilgisidir. D1.....Dn sekizlileri ise komutun verileri, Le ise karttan beklenen cevap verilerinin uzunluğudur. Lc, D1.....Dn ve Le bilgileri komutun gövdesini oluştururlar.

Dört değişk APDU tipi bulunmaktadır. Bu tiplere ilişkin bir tablo aşağıda verilmiştir.

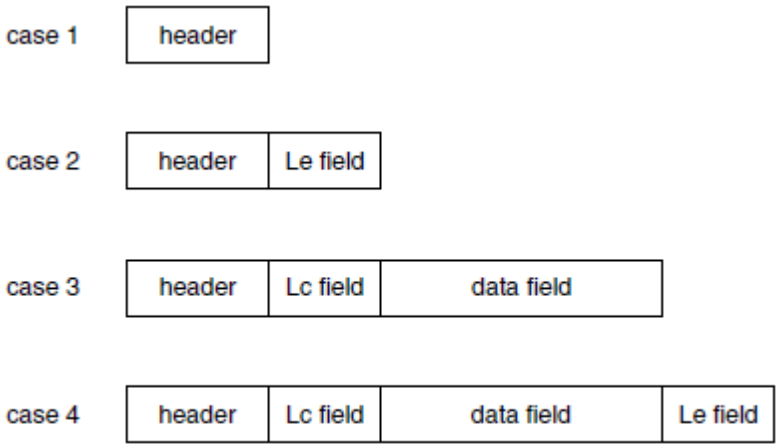


Figure 6.41 The four possible command APDU cases

APDU cevap veri yapısı

APDU cevabı, kart tarafından gelen APDU komutuna karşılık dış birime (terminale) gönderilir. Bu bilgi seçimli olan cevap verisi ve zorunlu olan iki sekizlik durum bilgisini kart gönderilen APDU komutuna kaşılık dış birime (terminale) gönderir. Bu durum aşağıdaki şekilde ayrıntılı olarak gösterilmiştir.

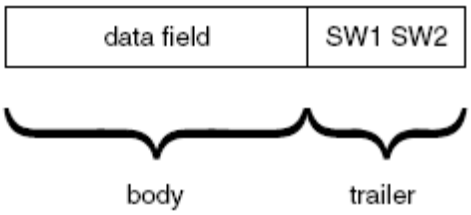


Figure 6.42 Structure of the response APDU

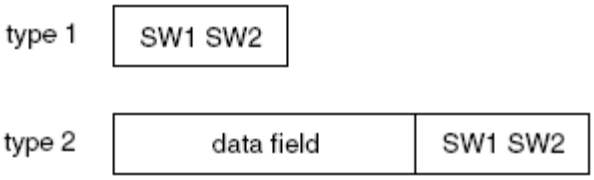
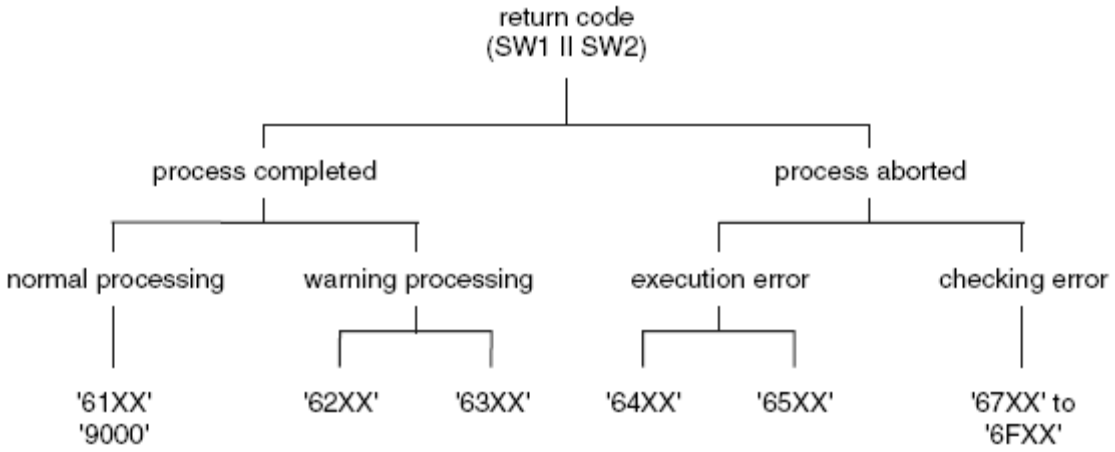


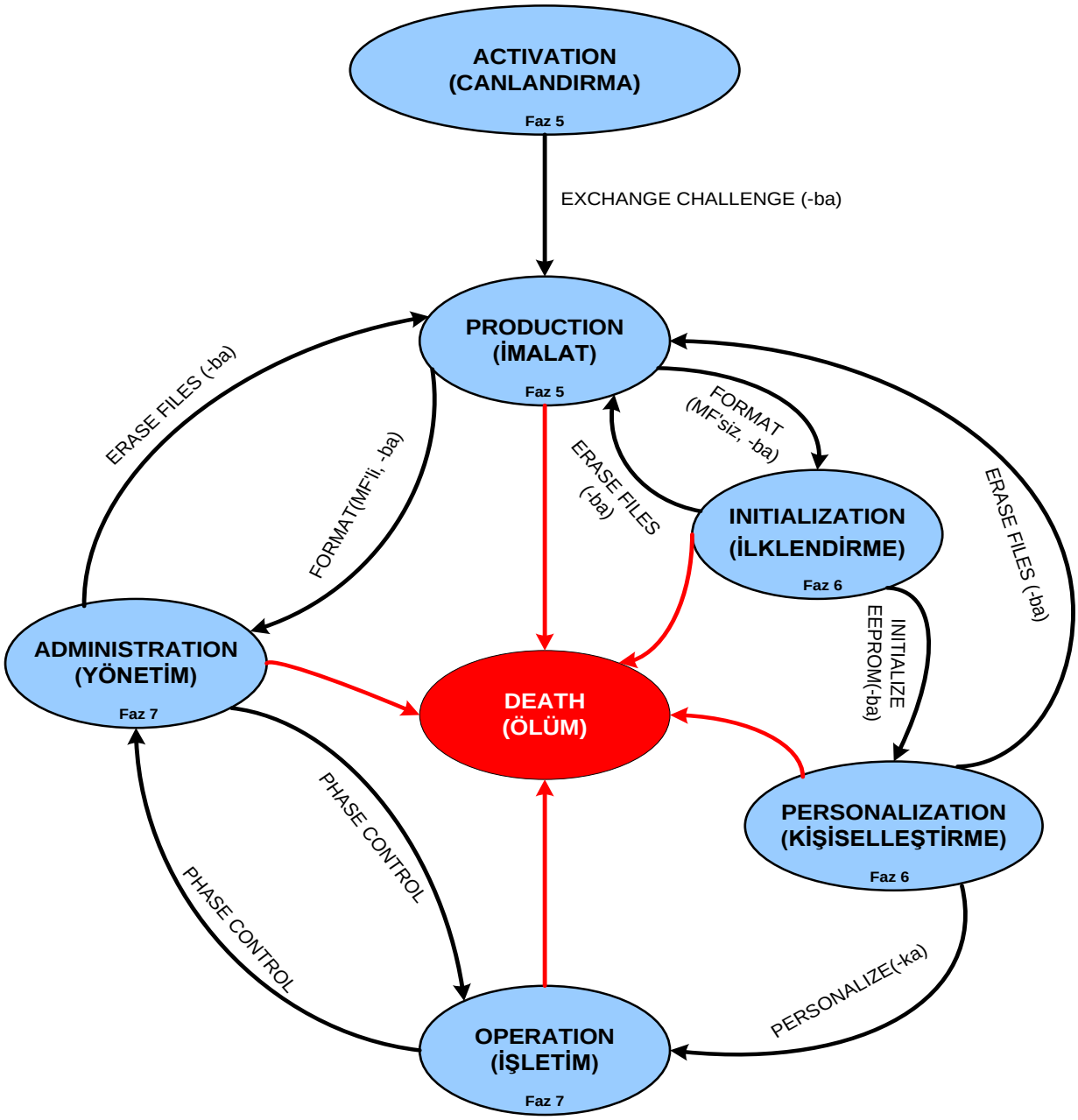
Figure 6.43 The two types of response APDUs

Kart gönderilen APDU komutuna mutlaka cevap bilgisi göndermektedir. Bu cevap bilgisi ISO 7816'da tanımlı standarta uygun olmak zorundadır. Gönderilecek zorunlu bilgi alanı aşağıdaki yapıda olmak zorundadır.



Akıllı Kart Yaşam Evreleri

Akıllı Kart üretiminden kullanıcıya ulaşmasına ve daha sonra tüketilmesine kadar bir çok evreden geçer. Bu geçiş evrelerinde akıllı kart üzerinde çeşitli işlemler yapılır. Örneğin kart fiziksel olarak üretildikten sonra CANLANDIRILMADAN asla kullanılamaması gerekir, canlandırılan bir kart üretim evresinde tıpkı bir bilgisayar harddisk'i gibi FORMAT komutu ile şekillendirilecektir. Daha sonra ILKLENDIRME komutu ile ilklendirilmek üzere ilklendirici firmaya oradan da KİŞİSELLEŞTİRME komutu ile kişiselleştirilmek üzere kişiselleştiriciye aktarılacaktır. Akıllı kartın bu yaşam döngüsü aşağıdaki şekilde verilmiştir.



Şekil – 2 Akıllı Kart yaşam döngüsü

Akıllı kart yaşam döngüsünde evre değişimine neden olan bazı komutlar vardır. Bu komutlar aşağıdaki komutlar bölümünde açıklanmaktadır.

Akıllı Kart Komutları

Dış birim veya terminal ile Akıllı Kart arasındaki iletişim Usta – Çırak (Master – Slave) çalışma prensibine göre yapılmaktadır. Bu çalışma yöntemi yarı çift yönlü iletişim prensibine dayanır ve Usta veya Efendi (Master) olarak Terminal APDU komutunu gönderir, Çırak veya Köle (Slave) olarak Akıllı kart gelen komutu alır işler ve sonucu cevap verisi olarak geri döndürür. Akıllı sorumluluk alarak kart kendi başına terminale herhangi bir cevap veya komut gönderemez, sadece "Akıllı Kart Veri İletişimi" bölümünde açıklandığı üzere ATR bu prensibin dışında tutulur çünkü ATR reset darbesine karşılık kartın çalışma ilkelerini kart okuyucuya ilettiği bir grup veriden oluşur.

Akıllı kartlar ile ilgili şu ana kadar ISO'nun hazırladığı 13 ayrı standart mevcuttur. Bu standartlar ISO 7816 olarak adlandırılan stnadartlarda mevcuttur. Akıllı kartlar ile ilgili komutların ayrıntısı ISO 7816-4 dokümanında verilmektedir. Ancak genel bir şema aşağıda gösterilmiştir.

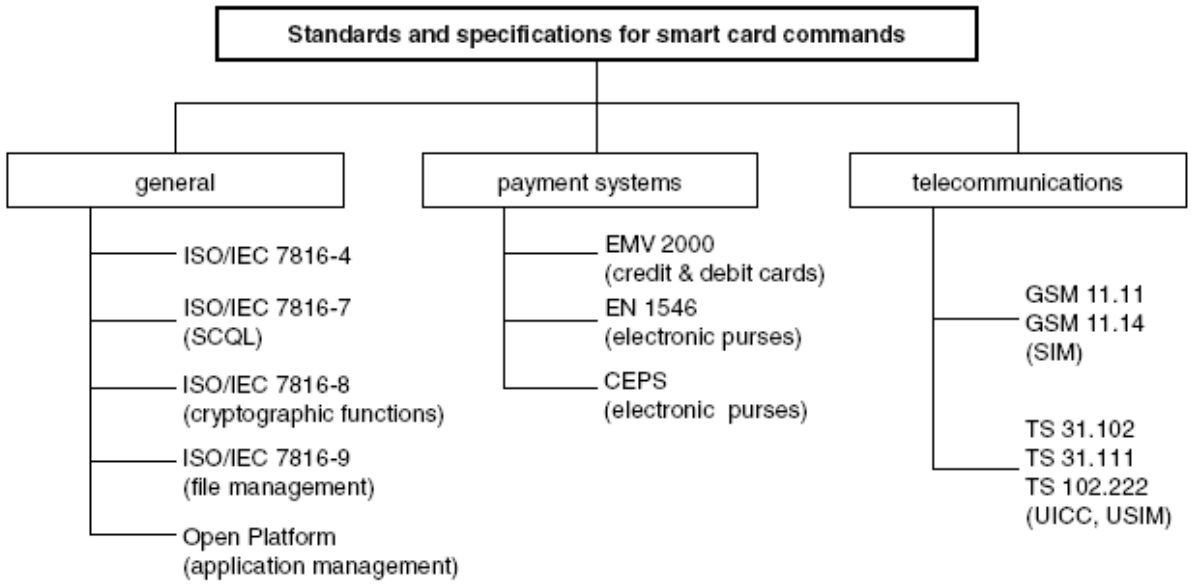


Figure 7.1 The most important standards and specifications for smart card commands

Burada akıllı kart komutları olarak aşağıdaki listede bulunan komutlar açıklanacaktır. Ayrıca listedeki komutlara ek olarak ICAO 9303'un gerçeklenmesi için ihtiyaç duyulan aşağıdaki komutlar ayrı bir başlık altında açıklancaktır.

- SELECT
- READ BINARY
- GETCHALLENGE
- EXTERNALAUTHENTICATE
- MSE
- CDS
- VERIFYCERTIFICATE

Akıllı Kartın İşletim Sisteminin dış dünya ile bağlantısı olan komutların listesi

Aşağıdaki listede bazı özel amaşlı komıutlar haricindeki tüm komutlar ISO7816-4 / 8 standartında açıklanan komutlardır.

KOMUT	INS	AÇIKLAMA
KART TEST (*)	1BH	Kod bütünlüğü kontrolu ve canlandırma kriptogramının oluşturulmasında kullanılacak chip sei no'sunu verir
EXCHANGE CHALLENGE (*)	86H	Sistem anahtarlarını ilklendirir
CHANGE KEY (*)	1CH	Sistem anahtarlarını değiştirir
FORMAT (*)	40H	Dosya dizin alanını formatlayıp kartı ilklendirme evresine geçirir
İLKLENDİRME (*)	02H	İlklendirme şablonunu yükler
KIŞISELLEŞTIRME (*)	08H	Kişiselleştirme şablonunu yükler
ERASE FILE	06H	Kartı imalar evresine döndürür
FAZ DEĞİŞTİRME (*)	09H	Yönetim ve İşletim evreleri arası geçişi sağlar
GET DATA	CAH	Karta ait bazı verileri arayüz birimine aktarır
PUT DATA	DAH	Arayüz biriminden gelen bazı verileri karta aktarır
GET CHALLENGE	84H	8 Byte rastgele sayı ürettir
INTERNAL AUTHENTICATE	88H	Uygulamayı kullanıcıya asıllar
EXT AUTHENTICATION	82H	Kullanıcıyı uygulamaya asıllar
MANAGE SECURITY	22H	Güvenlik öğesi bileşenlerini günceller
PERFORM SECURITY OPERATION	2AH	Sayısal imza, sayısal imza onaylama, şifreleme, deşifreleme, kriptografik sağlama hesaplama, kriptografik sağlama onaylama işlemlerini gerçekler
GET RESPONSE	C0H	Önceden okunan verinin devamını döndürür.
DOSYA AÇMA (*)	15H	FID veya isme göre dosya açar.
DOSYA SİLME (*)	16H	Dosya sistemindeki bir dosyayı siler.
DİZİN AÇMA (*)	10H	FID veya isme göre dizin açar.
DİZİN SİLME (*)	11H	Dosya sistemindeki bir dizini siler.
DOSYA KAPATMA (*)	17H	Seçilen dosyanın kapatılması.
SELECT FILE	A4H	Dosya veya Dizin seçme.
READ BINARY	B0H	İkili yapıdaki dosyaları okur.
UPDATE BINARY	D0/D6H	İkili yapıdaki dosyalara veri yazar veya günceller.
ERASE BINARY	0E	İkili yapıdaki dosyalardan veri siler.
READ RECORD	B2H	Kayıt yapısındaki dosyaları okur.
UPDATE RECORD	D2/DCH	Kayıt yapısındaki dosyalara yazar veya günceller.
APPEND RECORD	E2H	Kayıt yapısındaki dosyalar kayıt ekler.
VERIFY	20H	Sistem ve uygulama PIN doğrulamasını yapar.
CHANGE REFERENCE DATA	24H	PIN veya PUK değiştirmek için kullanılır.
RESET RETRY COUNTER	2CH	PIN hatasından dolayı bloke olmuş uygulamanın PUK ile blokesinin kaldırılmasını sağlar
ANAHTAR YAZ\SİL (*)	1EH	Anahtar, ID ve algoritmalarını siler, yazar.
ANAHTAR OKU (*)	1FH	Anahtarların ID, algoritma ve açık parçası okunur
DİZİN İÇERİĞİ OKUMA (*)	18H	Yol veya dizin içindeki DD dizinlerin listesini verir
DOSYA\DİZİN BAŞLIK ALANI DEĞİŞTİRME (*)	19H	İsim ve Erişim değiştirir.
OTURUM KAPATMA (*)	1AH	Reset yapılmadan başlangıç durumuna döner veya doğrulanmış uygulamanın doğrulamasını kaldırır.

1. **CHANGE KEY**

İşlevi
Başlangıç ve Kişiselleştirme anahtarlarının değiştirilmesini sağlar.

Standart / Kullanım
İşletim Sistemine özgü / İmalat, ilklendirme, kişiselleştirme ve yönetim evrelerinde kullanılır.

Parametreleri

CLA	INS	P1	P2
80H / 84H	1CH	00H	Anahtar referansı

Veri Alanı

Lc	Komut veri alanı	Le
20H	Değiştirilecek anahtarın sırasıyla eski A, eski B, yeni A ve yeni B bileşen	Yok

2. **FORMAT**

İşlevi
Kartın, dosya dizin sisteminin silinmesini, MF oluşturularak formatlanmasını ve istenilen yaşam evresine geçirilmesini sağlar.

Standart / Kullanım
İşletim Sistemine özgü / İmalat evresinde kullanılır.

Parametreleri

CLA	INS	P1	P2
80H / 84H	40H	00H	Evre

Veri Alanı

Lc	Komut veri alanı	Le
10H	Başlangıç anahtarının yine başlangıç anahtarı ile şifrelenmiş hali	Yok

3. **GET DATA**

İşlevi
Karta ait birtakım verilerin arayüz birimine aktarılmasını sağlar.

Standart / Kullanım
ISO 7816-4 / İmalat, ilklendirme, kişiselleştirme, yönetim, işletim ve ölüm evrelerinde kullanılır.

Parametreleri

CLA	INS	P1	P2
00h/ 04H	CAH	01H	İstenen bilgi

Veri Alanı

Lc	Komut veri alanı	Le
Yok	Yok	Cevap veri boyu

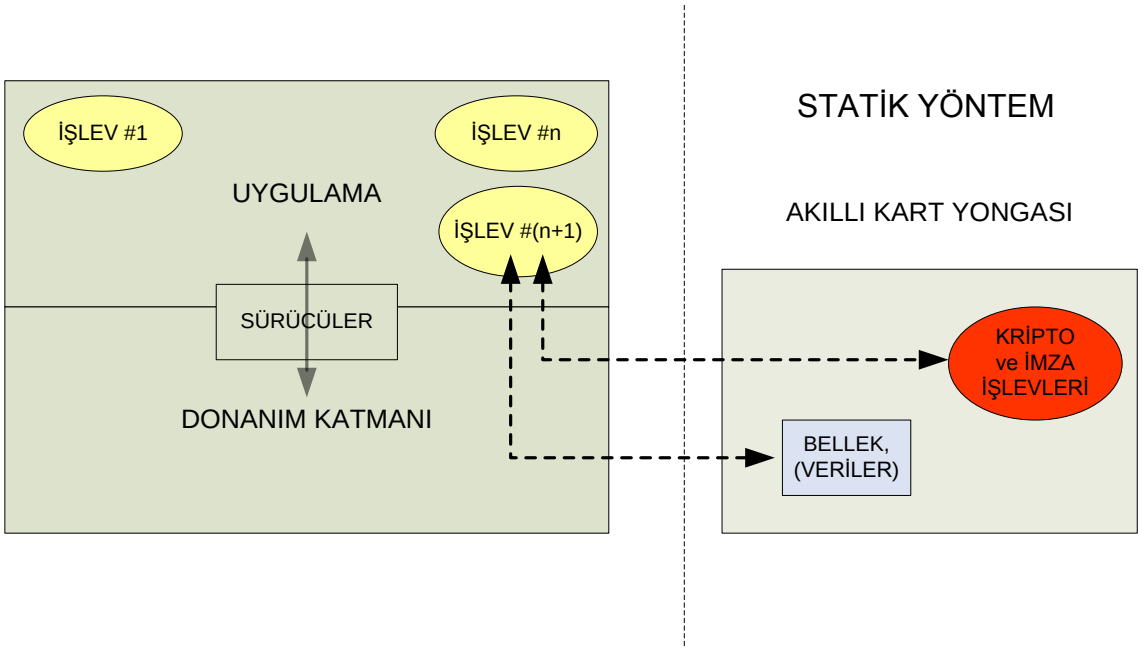
Cevap ve Dönüş kodu

Cevap veri alanı	SW1	SW2	Açıklama
Kart verisi	90H	00H	İŞLEM_BAŞARILI
Yok	67H	00H	LC_HATASI : Lc boş olmalı
Yok	6AH	86H	DESTEKLENMEYEN_PARAMETRE : P1 0x01
Yok	6AH	86H	DESTEKLENMEYEN_PARAMETRE : P2 0x01, 0x02, 0x03, 0x04, 0x05, 0x06 ya da 0x07 olmalı
Yok	6CH	00H	LE_DAHA_UZUN : Le gerkenden uzun

Asıllama (Authentication) gereksinimi

Akıllı kartı kullanan birimin güvenlik gereksinimi, bu donanım ve yazılımı kullanan kişilerin gerçek ve yetkili kişi olup olmadığıdır ve değişik şekilerde denetlenmektedir. Günümüzde kullanılan yöntemlerde veri genellike statik (değişmez) olarak sorgulanıp karşılaştırılmaktadır. Değişiklik yapılarak güvenlik gereksinimi sorgulamsında ise çoğunlukla sorgulayıcı uygulamanın veriler üzerinde belli bir algoritma ve anahtar kullanarak uyguladığı mekanizmalar ile sağlanmaktadır, ancak özü itibarı ile bu yöntemlerde de değişkenlik denetlenen verilerde değil gizlenen anahtarlardadır.

Bu yönteme ilişkin model aşağıda gösterilmiştir. Bu modelde koruma modülü içerisindeki veriler veya anahtarlar gizlidir. Ancak bu verileri denetleyen birimin ilgili işlevi bypass edilmesi durumunda gizlilik ve güvenlik devre dışı bırakılmış olur. Bu işlem bir PC üzerinde koşan yazılım ile gerçeklenecek olursa yazılıma müdahale eden bir yasadışı yazılım parçacığı güvenliği devre dışı bırakmış olur.



Kabul etmek gerekir ki burada mutlaka gizlenen bir bilgi bulunmakta ve gizlenen bilgi algoritma veya veri değil anahtarlar olmaktadır. Bilindiği kadarıyla güvenliğin en etkili olduğu sistem asimetrik şifreleme yöntemidir. Çünkü bu yöntemde asıllama ve doğrulama için dış birimde tutulmasına ihtiyaç bulunan *özel anahtar (private key)* bulunmaktadır. Ancak sistemin yanılması özel anahtarın elde edilmesi veya daha basit olan ise özel anahtar ile imzalanmış (kapatılmış) bir verinin sistemin denetlendiği yerde bypass edilmesi kolaylığıdır. Bu durumda sistem yazılım ve donanımı kullanan kişi veya otomasyon sistemini *gerçek ve asıl kabul eder*. Bu durumda sistem yanıltılmış olur. Buna alternatif bir çözüm güvenlik işlevi tanımlamasıdır.

Akıllı kartlarda güvenliğe yeni bir yaklaşım, *Güvenlik İşlevi*

Kart üzerindeki işletim sisteminin güvenlik modülü veriler üzerinden güvenlik işlerini yürütmesinin yanı sıra güvenlik işlevini de çağırarak uygulama geliştiricisinin kendisine ait güvenlik gereklerini de işletme olanağı tanımış olur. Ancak uygulamacının geliştirmiş olduğu güvenlik işlevinin mutlaka bazı güvenlik kriterlerini sağlamış olması gerekir. Bunlar;

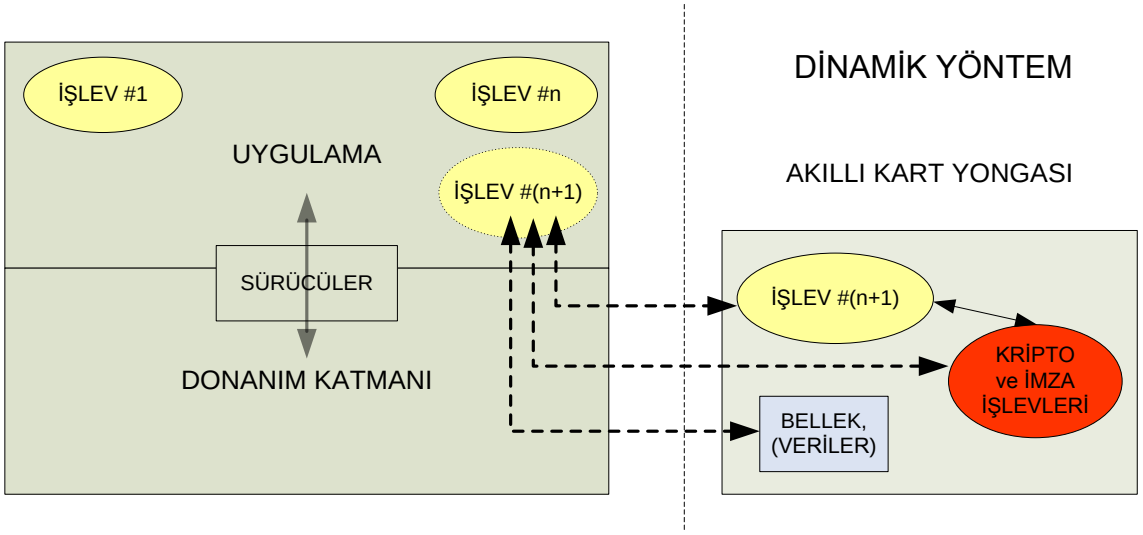
- Güvenlik işlevinin sistem güvenliğini ilgilendiren sistem anahtarlarına erişimi yasaktır (sistem güvenliğinin kullandığı alanları güvenlik işlevi paylaşamaz).
- Güvenlik işlevi ancak ve ancak sistem sertifikası ile yüklenebilir.
- Alt işlevler çağıramaz, kendisi için ayrılan alan dışına taşamaz.
- Ancak ve ancak sistemin tanımladığı veri alanlarını kullanabilir (RAM ve PROM alanı sınırlıdır).

Güvenlik İşlevi Modeli

Güvenlik işlevi, tanım olarak güvenlik gereksinimi olan bir donanım ve yazılım biriminin (sistemin) yeteneklerinin veya işlevlerinin bir kısmını bir başka birimde konuşlandırması ve sistemin çalışması için bu birimin varlığına ihtiyaç duymasına sebep olan işleve verilen addır.

Güvenlik işlevini üzerinde barındıran ve koşturan etkin sisteme de *güvenlik işlevi aracı* denmesi uygun olur. Burada güvenlik işlevi aracı olarak akıllı kart kullanılması gerek maliyet gerekse kullanılabilirlik açısından oldukça uygundur. Ancak güvenlik işlevi aracı olarak güvenlik önlemi alınmış bir donanım ve yazılımlada bu işlev gerçekleştirilebilir.

Aşağıdaki şekilde Güvenlik İşlevinin çalışmasına ilişkin bir model gösterilmektedir. Bu modelde iki ortam bulunmaktadır. Bir ortam güvenlik işlevini içerisinde barındıran yazılım ve donanımdan oluşan akıllı kart, diğer bir ortam ise akıllı kart ile irtibatlı güvenlik gereksinimi olan yazılım ve donanım birimi.



Yukarıdaki modelde güvenlik gereksinimi olan sistem, sonuç üreten kendi iç işlevlerinden birisini (tercihen en basit ve matematiksel bir formül içeren işlevi) sistem oluşturulduğunda akıllı kart veya akıllı kartlara yükler. Daha sonra bu yazılımın kullanılabilmesi için gerekli olan akıllı kart(lar) olmadan sistem çalışmaz. Ancak ve ancak sistemin çalışması için mutlaka güvenlik işlevinin bulunduğu akıllı kart mutlaka sisteme tanıtılmalıdır. Bunun için bilinen asıllamalar (sabit verilerin bütünlüğü, sertifikanın geçerliliği, ortak asıllama) tamamlandıktan sonra kart içerisinde yer alan güvenlik işlevinin çalıştırılıp verilen girdilere karşılık yazılımda kullanılacak çıktılar elde edilmesi ve elde edilen bu veriler doğrultusunda işlemlere devam edilebilmesi ile güvenlik sağlanmış olacaktır.

Güvenlik işlevi burada güvenliğin sağlanması istenen yazılımın bir parçası olarak kopyalanamayan bir birim içerisinde çalıştırılmaktadır.

Güvenlik işlevinin bir çok değişik uygulamada kullanmak olasıdır. Bunlardan bazıları yazılımın korunması, kimlik uygulamalarının korunması, v.b. gibi uygulamalardır. Örneğin geliştirilen bir yazılımın kopyalanmasını engellemek için yazılımın bir işlevi güvenlik işlevi olarak güvenlik işlevi aracına yüklenirse ve yazılım bu işleve ihtiyaç duyduğunda bu araç içerisinde bu işlevi koşturursa dış dünyadaki yazılım korunmuş olur.

Burada asıl amaç taklit edilemeyen ve kopyalanamayan elektronik bir kimlik kartı tasarlamak olduğu için güvenlik işlevinin kimlik uygulamasında kullanılması üzerinde durulacaktır.

Güvenlik işlevinin Terminal - Akıllı kart Arayüzü

Güvenlik işlevi var olan ve standartlara uyumlu bir arayüz yapısına sahiptir (ISO 7816 APDU formatı). Ancak CLA olarak var olan sınıflardan farklı bir sınıfta olacaktır (Örneğin 0xA0). Bu durumda;

Örnek bir uygulama yükleme komut yapısı

CLA : 0x8A
INS : 0x01 (İşlev yükleme)
P1 : daha önceden yüklenmiş anahtar Algo(3 bit) & No(5 bit).
P2 : Blok numarası.
Lc : Giriş veri boyu.
Data : Güvenlik işlevi (koşturulabilir kod) .
Le : 0x00

Örnek bir guvenlik işlevi koşturma komut yapısı

CLA : 0x8A
INS : 0x02 (İşlev koşturma)
P1 : daha önceden yüklenmiş anahtar Algo(3 bit) & No(5 bit).
P2 : Ilklendirmede yüklenen güvenlik işlev parametresi
Lc : Giriş veri boyu.
Data : Güvenlik işlevi giriş verileri.
Le : Beklenen çıkış verileri (P1'de belirtilen anahtar ile şifrelenmiş olarak)

Elektronik Kimlik ve pasaport sistemlerinde kullanılan Protokol ve Standartlar

ISO 14443

- RFID fiziksel iletişimi tanımlar.

ISO 7816

- Temaslı Akıllı kartlar için geliştirilmiş standart.
- APDU komut ve cevap standartlarını tanımlar.

ICAO 9303 e-pasaport standartı

- Pasaportlar için özel ISO 7816 komut ve cevapları ve ek EU standartları.
- ICAO speklerinin seçimli bölümlerini standartlaştırır.
- ICAO üzerinde ek gelişmiş güvenlik mekanizmaları.

Elektronik kimlik kartlarında veya Pasaportlarda güvenlik mekanizmaları

Günümüzde elektronik kimlik kartı uygulamalarında temel olarak ICAO 9303 standartında tanımlı güvenlik mekanizmaları kullanılmaktadır. Bu mekanizmalar;

- Pasif asıllama (Passive Authentication, PA),
  - ◊ Yongadaki pasaport verileri üzerindeki sayısal imza

- Temel Erişim Denetimi (Basic Access Control, BAC)
  - ◊ Yongaya erişim denetimi, yetkisiz erişim ve sahtecilikten koruma,

- Etkin Asıllama (Active Authentication, AA)
  - ◊ Yonganın Asıllanması (yonganın klonlanmasından koruma)

- Genişletilmiş Erişim Denetimi (Extended Access Control, EAC)
  - ◊ Yonga ve Terminal Asıllaması

Olarak tanımlanmıştır.

Pasif Asıllama (Passive Authentication)

- Pasaport yongasında DG olarak adlandırılan 16 veri grubu bulunur.
  - DG1 MRZ
  - DG2 Yüz
  - DG3 Parmak izi
  - DG 4 iris
  - .....
  - DG15 Etkin asıllama
  - .....
  - Güvenlik nesnesi
  - Veri gruplarının özetinin imzlanması.

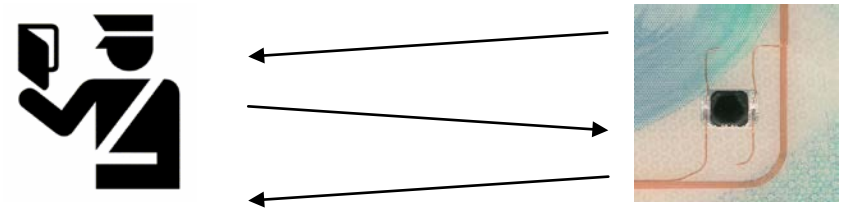
- Imzanın doğrulanması, terminaller ülkelerin imzaladığı sertifikaya ihtiyaç duyar.
- Pasif asıllama tüm elektronik pasaportlarda zorunludur.

Temel Erişim Denetimi (Basic Access Control)

- Temassız arayüzün hem avantajı hem de dezavantajı vardır.
- BAC sadece okuyucu asıllamasından sonra veri okumaya izin verir.
  - Okuyucu, pasaport MRZ bilgisinin doğruluğunu ispat eder.
- Asıllama Anahtarı, doküman numarası, doğum tarihi ve geçerlilik tarihinden türetilir.
- BAC ICAO için seçimli, EU için zorunludur.
- Birlikte çalışabilirlik,
- Brlikte Çalışabilirlik.
  - Pasaportun BAC korumalı olup olmadığı nasıl anlaşılır?
  - Deneme yoluyla anlaşılır.

Etkin Asıllama (Active Authentication, AA)

Pasaportun klonlanmasına karşı (BAC'ın yapamadığı) koruma. Örneğin pasaport yongasının asıllanması.



- Hükümet tarafından imzalanmış açık anahtar (DG15).
- Challange gönderme
- İlgili özel anahtarın bilinmesinin ispatı.

Genişletilmiş Erişim Denetimi (Extended Access Control, EAC)

Pasaport yongası tarafından terminalin asıllanmasını kapsar.

- Bu neden istenir?
  - Özel hassas bilgilerin sızmasını engellemek (örneğin hotel girişleinde).
  - Özellikle parmak izi ve iris gibi özel bilgilerin sızmasını engellemek.

- Bu nasıl yapılır?
  - Bazı terminallerin sertifikalanması ile.
  - X509 sertifikası yerine ISO 7816'da tanımlı kart doğrulama sertifkası (Card Verfiable Certificate, CVC) kullanılması ile

- Problem Nedir?
  - Sertifika iptalini gerçekleştirmek zordur.
    - Tüm pasaportlar üzerindeki sertifikaları nasıl iptal edebiliriz?
    - Pasaportun terminal sertifikanın geçerleyeceği fazla zamanı yok.
    - Yonga sadece son işlemin kayıt tarihini tutar.

Akıllı kartlarda desteklenen uygulamalar, *ICAO 9303 MRtd Standartı*

Temaslı/Temassız olarak iki değişik tipi bulunan Akıllı Kart tabanlı kimlik uygulaması hem kimlik, hemde pasaport uygulamasını desteklemektedir. Bu nedenle bu uygulamalarda standart olan ISO7816-4 ve ICAO 9303 standartlarını uygulama olarak barındırmaktadır.

ICAO 9303 'te geçerli Dosya Sistemi :

Temaslı/Temassız Akıllı kart tabanlı kimlik uygulamasının kullandığı temel dosya yapısı aşağıdaki gibi olmalıdır. Burada uygulamanın zorunlu ve seçimlik veri grup ve elemanları dosya sistemindeki ilgli dosyalarda tutulacaktır.

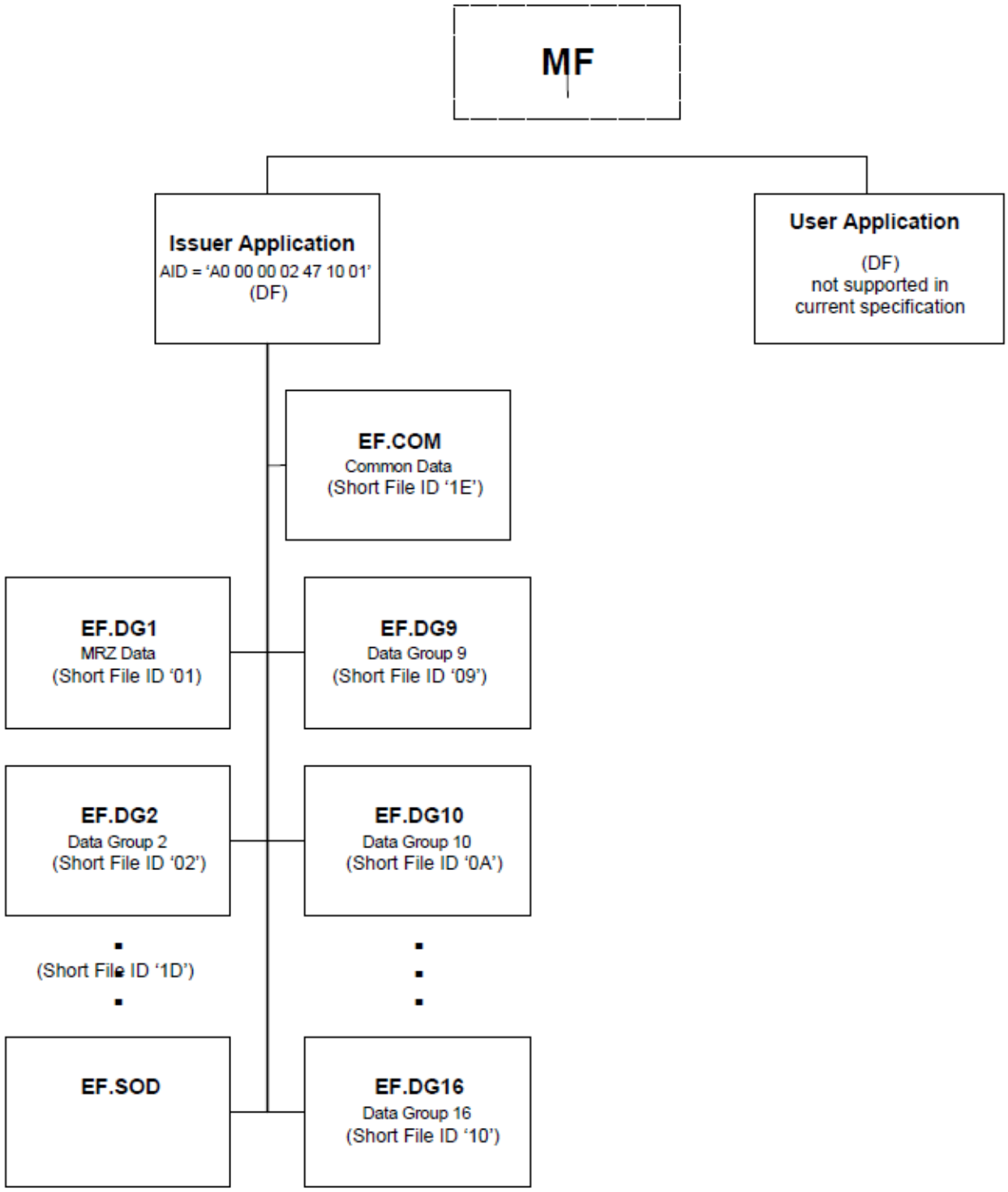


Figure A.1

Dosya Sisteminde kullanılan Dosyalar ve içerikleri

Dosya Sistemindeki dosya tanımlayıcıları standart ICAO tarafından standart olarak tnımlanmıştır ve aşağıdaki tabloda gösterilmiştir.

ISSUING STATE OR ORGANIZATION APPLICATION				
Data Group	EF Name	Short EF identifier	FID	Tag
Common	EF.COM	'1E'	'01 1E'	'60'
DG1	EF.DG1	'01'	'01 01'	'61'
DG2	EF.DG2	'02'	'01 02'	'75'
DG3	EF.DG3	'03'	'01 03'	'63'
DG4	EF.DG4	'04'	'01 04'	'76'
DG5	EF.DG5	'05'	'01 05'	'65'
DG6	EF.DG6	'06'	'01 06'	'66'
DG7	EF.DG7	'07'	'01 07'	'67'
DG8	EF.DG8	'08'	'01 08'	'68'
DG9	EF.DG9	'09'	'01 09'	'69'
DG10	EF.DG10	'0A'	'01 0A'	'6A'
DG11	EF.DG11	'0B'	'01 0B'	'6B'
DG12	EF.DG12	'0C'	'01 0C'	'6C'
DG13	EF.DG13	'0D'	'01 0D'	'6D'
DG14	EF.DG14	'0E'	'01 0E'	'6E'
DG15	EF.DG15	'0F'	'01 0F'	'6F'
DG16	EF.DG16	'10'	'01 10'	'70'
Security Data	$EF.SO_D$	'1D'	'01 1D'	'77'

Table A1
Mandatory

Kullanılacak komut kümesi

MRtd olarak adlandırılan Seyahat dokümanı en az aşağıdaki komutları destekleyecektir;

- SELECT
- READ BINARY
- GETCHALLENGE
- EXTERNALAUTHENTICATE
- MSE
- CDS
- VERIFYCERTIFICATE

Tüm komut formatları ISO/IEC 7816-4'te tanımlanmıştır.

Pasif Asıllama

Temassız IC, LDS veri gruplarının yanında SOD (Document Security Object) nesnesini de içerir. Bu nesne elektronik kimlik veya pasaportu veren devlet tarafından imzalanmıştır.

LDS içeriğinin özeti (hash) alınmış gösterimini içerir. Her devlete ait KPu_{DS}'yi (Document Signer Public Key) içeren ya da MRtd'den C_{DS}'yi (Document Signer Certificate) okuyan bir sorgulayıcı sistem SOD'yi doğrular (verify). Bu yolla SOD içeriği sayesinde LDS içeriği asıllanmış olur. Bu yöntemde IC'nin hesaplama yeteneği kullanılmaz.

Pasif asıllamada SOD'nin ve LDS'nin içeriklerinin değişmemiş olduğu kanıtlanmış olur. Ancak IC'nin toptan kopyalanıp kopyalanmadığı bilinemez.

Pasif asıllamanın yapılabilmesi için sorgulayıcı sistem, pasaportu veren devlete ait Country Signing Certificate'a (C_{CSCA}) ve C_{DS}'ye sahip olmalıdır. SOD'nin doğrulanması için C_{DS}'yi kullanmadan önce, C_{DS}'yi C_{CSCA} ile doğrular.

Pasif asıllama şu işlem adımları ile gerçekleştirilir:

- SOD temassız IC'den okunur.
- SOD'nun içinden DS okunur. (Ya da sistemde o ülkeye ait DS bulunmaktadır.)
- SOD'nin sayısal imzası KPu_{DS} ile doğrulanır. Böylece SOD'nin asıl, C_{DS}'de belirtilen ülke tarafından verilmiş ve değişmemiş olduğundan emin olunur. C_{DS}, SOD doğrulanması için kullanılmadan önce KPu_{CSCA} ile doğrulanır.
- Sorgulayıcı sistem LDS içerisinden ilgili veri gruplarını okur.
- Veri gruplarının özeti alınır ve SOD'deki özetle karşılaştırılır. Böylece verilerin asıl ve değişmemiş olduğundan emin olunur.

Aktif Asıllama

IC'nin kopyalanması aktif asıllama mekanizması ile engellenebilir. IC'de kendine ait KPr_{AA} ve KPu_{AA} (Active Authentication Key Pair) anahtarları tutulur. KPu_{AA}'ya ait bilgilerin özet gösterimi SOD'de tutulur ve issuer'ın sayısal imzasıyla asıllanır. KPr_{AA} ise IC'nin güvenli belleğinde tutulur.

Görünen (optik) MRZ'nin SOD'deki özeti alınmış MRZ ile asıllanması, IC'deki KPr_{AA} ve KPu_{AA} anahtarları kullanılarak yapılan challange-response ile birleştirilir. Böylece sorgulayıcı sistem SOD'nin gerçek bir MRTD'ye ait gerçek bir IC'den okunduğunu doğrulamış olur. Bu asıllama yönteminde IC'nin hesaplama yetenekleri kullanılır.

Aktif asıllama şu işlem adımları ile gerçekleştirilir:

- Gözle okunan MRZ verisi, Veri Grubu 1'deki (DG1) veriler ile karşılaştırılır. DG1 verilerinin asıllığı ve bütünlüğü zaten pasif asıllama ile kanıtlandığından MRZ'nin de asıllığı ve değişmemiş olduğu kanıtlanmış olur.

- Pasif asıllama DG15'in de asıllığını ve bütünlüğünü kanıtladığından KPu_{AA}'nın da asıllığı ve bütünlüğü kanıtlanmıştır.

- SOD'nin kopya olmadığından emin olmak için KPr_{AA} ve KPu_{AA} anahtar çifti kullanılarak şu şekilde bir protokol izlenir: IFD, bir rastgele sayı üreterek Internal Authenticate komutu ile karta yollar. Kart bunu KPr_{AA} anahtarı ile imzalar ve IFD'ye geri yollar. IFD KPu_{AA} anahtarı ile imzayı doğrular.

Temel Erişim Kontrolü (Basic Access Control)

IC'deki veriler yetkisiz şekilde okunabilir (skimming) ya da IC ile okuyucu arasındaki kriptosuz iletişim gizlice dinlenebilir. Skimming fiziksel olarak engellenebilse de eavesdropping engellenemez. Bu sebeple devletler BAC yöntemini kullanabilirler. Bu yöntemle skimming ve eavesdropping engellenir, MRtd sahibinin IC'deki kendisine ait bilgilerin güvenli olarak yetkili biri tarafından okunduğundan haberi olur.

Sorgulayıcı sistem, bir challange response protokolü yardımıyla IC'ye ait, MRZ üzerindeki bilgilerden türetilmiş K_{ENC} ve K_{MAC} (Document Basic Access Keys) anahtarlarını bildiğini kanıtlar. Bunun yanında sorgulayıcı sistem asıllandıktan sonra temassız IC, okuyucu ile arasında kriptolu haberleşmeyi güvenli mesajlaşma yoluyla zorunlu kılar.

BAC'ın desteklenmesi için, sorgulayıcı sistemde MRtd'den K_{ENC} ve K_{MAC}'in okunabilmesi için bir MRZ okuyucu ya da elle veri giriş ekipmanı bulunmak zorundadır.
BAC için sırasıyla şu basamaklar geçekleştirilir:

- Sorgulayıcı sistem "Doküman Numarası", "Doğum Tarihi" ve "Belgenin Son Kullanım Tarihi" alanlarını okur. Bu verinin SHA-1 özeti alınır ve özetin ilk 16 byte'ı Kseed olur. Kseed, c=0x 00 00 00 01 ile birletirilip SHA-1 özeti alınınca çıkan sonucun ilk 8 byte'ı K_{ENC}'e ait Ka, sonraki 8 byte'ı K_{ENC}'e ait Kb olur. Kseed, c=0x 00 00 00 02 ile birletirilip SHA-1 özeti alınınca çıkan sonucun ilk 8 byte'ı K_{MAC}'e ait Ka, sonraki 8 byte'ı K_{MAC}'e ait Kb olur.

- Terminal karta Get Challange komutu yollayarak 8 byte'lık bir rastgele sayı ister, kart yollar. Terminal 8 byte'lık bir rastgele sayı ve 16 bytelık keying material (K_IFD) yaratır ve bunları karttan gelen rastele sayı ile birleştirir. Elde edilen bu veriyi K_{ENC} ile şifreler (E_IFD) ve bu kriptogramın da K_{MAC} ile MAC'ini alır (M_IFD). E_IFD ve M_IFD'yi yanyana birleştirerek karta Mutual Authenticate komutu yollar. E_IFD'ye ait MAC'i (M_IFD) kontrol eder. E_IFD kriptogramını çözer. Kendisi tarafından yollanmış olan rastgele sayının doğru gelip gelmediğini kontrol eder. Kendisine ait Keying Material'ı (K_ICC) yaratır ve bunu kendi rastgele sayısı ve terminalin rastgele sayısı ile birleştirir. K_{ENC} ile bu veriyi kapatır (E_ICC). K_{MAC} ile mac'ini alır (M_ICC). E_ICC ve M_ICC'yi birleştirerek yanıt yollar. Terminal gelen yanıttaki E_ICC'ye ait M_ICC'yi kontrol eder. E_ICC'yi çözer. Kendi göndermiş olduğu rastgele sayının doğruluğunu kontrol eder.

- Asıllama tamamlandıktan sonra iletişim güvenli mesajlaşma ile devam eder. Güvenli mesajlaşma anahtarı K_IFD XOR K_ICC'nin Kseed olarak kullanılması sonucu ortaya çıkan KS_ENC ve KS_MAC anahtarları ile gerçekleştirilir.

Güvenli mesajlaşma mesaj yapısı şekildeki gibidir:

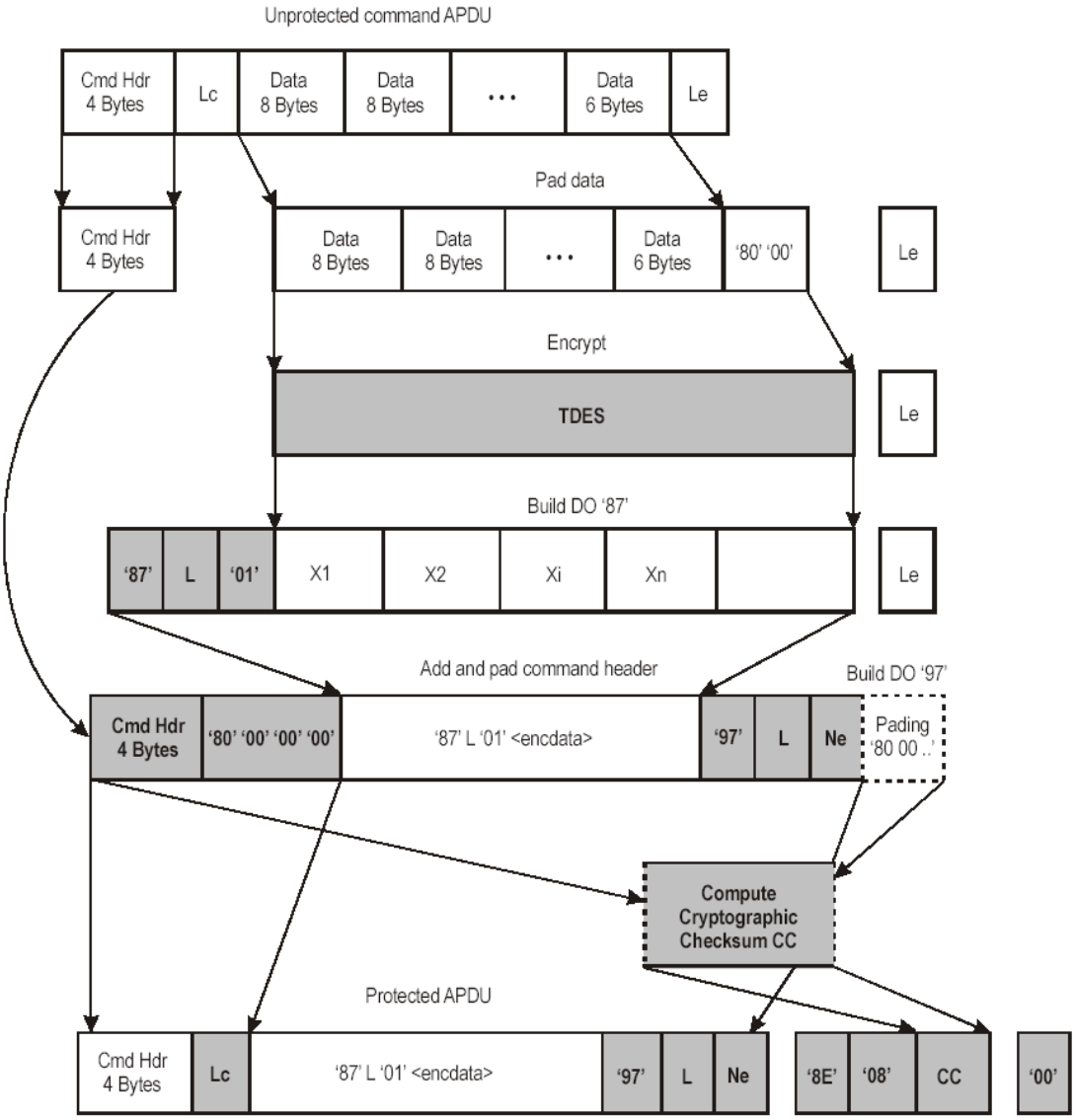


Figure A5-2. Computation of a SM command APDU for even INS Byte

Genişletilmiş Erişim Kontrolü (Extended Access Control)

IC içerisindeki biyometrik verilere (parmak izi, iris) erişimin sınırlanması için kullanılır. EAC, BAC'ye benzemektedir, ancak Document Basic Access Keys (K_{ENC} ve K_{MAC}) yerine Document Extended Access Keys kullanılır. Her bir IC'ye özgü Document Basic Access Keys ve kullanılacak EAC protokolü gerçekleştirime bağlıdır: MRZ'deki verilerden ve ulusal master key'den elde edilecek simetrik bir anahtar olabileceği gibi Card Verifiable Certificate ile ilgili asimetrik bir anahtar da olabilir. EAC'de IC'nin hesaplama kabiliyeti kullanılır.

Şifreleme

IC içerisindeki biyometrik verilere (parmak izi, iris) erişimin sınırlanması için kullanılan başka bir metoddur. IC içerisindeki biyometrik veriler bir anahtar ile şifrelenir. Sorgulayıcı sistemin de bu anahtara sahip olması gerekmektedir. Kripto algoritması, anahtarlar ve şifreleme protokolü gerçekleştirime bağlıdır.
MRtd'de kullanılan güvenlik mekanizmaları özetle tablodaki gibidir:

Method	Issuer	Insp. System	Benefits	Deficiencies
BASELINE SECURITY METHOD				
Passive Authentication (5.6.1)	**M**	**M**	Proves that the contents of the SO_D and the LDS are authentic and not changed.	Does not prevent an exact copy or IC substitution. Does not prevent unauthorized access. Does not prevent skimming.
ADVANCED SECURITY METHODS				
Comparison of conventional MRZ(OCR-B) and IC-based MRZ(LDS)	N/A	O	Proves that contactless IC's content and physical MRtd belong together.	Adds (minor) complexity. Does not prevent an exact copy of contactless IC and conventional document.
Active Authentication (5.6.2)	O	O	Prevents copying the SO_D and proves that it has been read from the authentic contactless IC. Proves that the contactless IC has not been substituted.	Adds complexity. Requires processor-ICs.
Basic Access Control (5.7)	O	O	Prevents skimming and misuse. Prevents eavesdropping on the communications between MRtd and inspection system (when used to set up encrypted session channel).	Does not prevent an exact copy or IC substitution (requires also copying of the conventional document). Adds complexity. Requires processor-ICs.
Extended Access Control (5.8.1)	O	O	Prevents unauthorized access to additional biometrics. Prevents skimming of additional biometrics.	Requires additional key management. Does not prevent an exact copy or IC substitution (requires also copying of the conventional document). Adds complexity. Requires processor-ICs.
Data Encryption (5.8.2)	O	O	Secures additional biometrics. Does not require processor-ICs.	Requires complex decryption key management. Does not prevent an exact copy or IC substitution. Adds complexity.

MRtd kartının içeriği aşağıdaki gibidir:

MF
|
|---------DF — LDS REQUIRED
|
|---------K_{ENC} OPTIONAL
|
|---------K_{MAC} OPTIONAL
|
|---------KPr_{AA} OPTIONAL
|
|---------EF — COM REQUIRED
|
|---------EF — SO_D REQUIRED
|
|---------EF — Datagroup_1 (MRZ) REQUIRED
|
|---------EF — Datagroup_2 (Encoded Face) REQUIRED
//
|---------EF — Datagroup_n OPTIONAL