ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ & ΔΙΚΤΥΩΝ

ΔΙΑΛΕΙΤΟΥΡΓΙΚΕΣ ΕΞΥΠΝΕΣ ΚΑΡΤΕΣ ΣΕ OPEN CARD FRAMEWORK

Ζιάκα Ευαγγελία

Βόλος
Οκτώβρης 2005

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ & ΔΙΚΤΥΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ:

ΔΙΑΛΕΙΤΟΥΡΓΙΚΕΣ ΕΞΥΠΝΕΣ ΚΑΡΤΕΣ ΣΕ OPEN CARD FRAMEWORK

Επιβλέποντες Καθηγητές :

Καθηγητής κ.Χούστης Ηλίας
Κ. Ηλιούδης Χρήστος

ΖΙΑΚΑ ΕΥΑΓΓΕΛΙΑ

ΒΟΛΟΣ
Οκτώβρης 2005

Περιεχόμενα

Πίνακας Εικόνων

Κεφάλαιο 1

Εισαγωγή

Το εισαγωγικό αυτό κεφάλαιο αποτελεί μια συνοπτική περιγραφή των περιεχομένων των κεφαλαίων που ακολουθούν και έχει σαν στόχο να κατατοπίσει τον αναγνώστη πάνω σε αυτά.

Σκοπός της διπλωματικής εργασίας είναι η μελέτη της τεχνολογίας των έξυπνων καρτών και η πειραματική υλοποίηση ενός συστήματος αυθεντικοποίησης με χρήση έξυπνων καρτών, ο οποίος θα επιτρέπει την πρόσβαση στους υπολογιστές του εργαστηρίου του τμήματος μόνο σε εξουσιοδοτημένους χρήστες.

Η διπλωματική εργασία είναι δομημένη συνολικά σε εφτά επιμέρους κεφάλαια. Το κεφάλαιο 1 αποτελεί την εισαγωγή και μια σύντομη περίληψη των υπολοίπων κεφαλαίων.

Στο δεύτερο κεφάλαιο ***Μέθοδοι Αυθεντικοποίησης*** αναλύεται ο όρος αυθεντικοποίηση και περιγράφονται τα βασικότερα μοντέλα αυθεντικοποίησης τα οποία διακρίνονται σε δύο κατηγορίες : στην αυθεντικοποίηση από στοιχεία που κατέχει ο χρήστης όπως είναι τα passwords, τα συνθηματικά μιας χρήσης, τα συστήματα πρόκλησης – απόκρισης, οι μαγνητικές κάρτες, οι έξυπνες κάρτες και στην αυθεντικοποίηση από χαρακτηριστικά του ίδιου του χρήστη όπου ανήκουν οι βιομετρικές τεχνικές αναγνώρισης όπως για παράδειγμα αναγνώριση με βάση το δακτυλικό αποτύπωμα, την ίριδα του ματιού, τη γεωμετρία παλάμης, αναγνώριση φωνής και αναγνώριση μορφολογίας του αμφιβληστροειδούς χιτώνα. Στη συνέχεια παρατίθενται τα πλεονεκτήματα των έξυπνων καρτών έναντι των άλλων μεθόδων τα βασικότερα από τα οποία είναι η ασφάλεια που παρέχουν σε επίπεδο υλικού, η αξιοπιστία, η πολύ-εφαρμογικότητα και η ευκολία στη χρήση.

Το τρίτο κεφάλαιο ***Τεχνολογία και Αρχιτεκτονική Έξυπνων Καρτών*** ασχολείται με την αρχιτεκτονική των έξυπνων καρτών και τα βασικά της συστατικά που είναι η κεντρική μονάδα επεξεργασίας, η μνήμη, η μονάδα εισόδου/ εξόδου και οι interface devices. Ιδιαίτερη έμφαση δίνεται στο λειτουργικό σύστημα των έξυπνων καρτών καθώς και στο σύστημα αρχείων που χρησιμοποιεί. Το τελευταίο αναφέρεται στο τρόπο με τον οποίο είναι οργανωμένα τα αρχεία και κατ' επέκταση τα δεδομένα που αποθηκεύονται στην κάρτα. Τα αρχεία λοιπόν μιας κάρτας έχουν τη δομή ενός ιεραρχικού δένδρου, ρίζα του οποίου είναι το Master File ενώ κάτω από αυτό βρίσκονται οι δύο άλλες κατηγορίες αρχείων που είναι τα Dedicated Files και Elementary Files τα οποία διακρίνονται σε δύο περαιτέρω κατηγορίες που είναι τα transparent και record files. Περιγράφεται επίσης ο τρόπος προσπέλασης αυτών των

αρχείων, οι εντολές που χρησιμοποιούνται για την ανάγνωση, εγγραφή και ενημέρωση τους καθώς και οι εντολές σύμφωνά με τις οποίες επιτυγχάνεται ένα ασφαλές σύστημα αρχείων στο οποίο η πρόσβαση και η προσπέλαση των αρχείων είναι απόλυτα ελεγχόμενη και επιτρέπεται μόνο σε εξουσιοδοτημένους χρήστες ενώ επιπλέον αναλύεται ο τρόπος με τον οποίο γίνεται η αυθεντικοποίηση και τα βήματα που ακολουθούνται κατά τη διάρκεια αυτής της διαδικασίας. Επιπλέον περιγράφονται τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται μεταξύ έξυπνης κάρτας και συσκευής ανάγνωσης τόσο στο transport layer όσο και στο application layer. Σε αυτό το υποκεφάλαιο περιγράφεται επίσης η δομή των πακέτων δεδομένων που ανταλλάσσονται μεταξύ κάρτας και αναγνώστη και τα οποία καλούνται TPDUs για τα transport protocols και APDUs για τα application protocols. Τέλος γίνεται μια σύντομη αναφορά στα πρότυπα των έξυπνων καρτών όπως αυτά έχουν καθοριστεί από το διεθνή οργανισμό προτύπων ISO καθώς και στα υπάρχοντα βιομηχανικά πρότυπα όπως : ***EMV, OPEN CARD FRAMEWORK, PESONAL COMPUTER/SMART CARD*** και ***JAVA CARD***.

Αντικείμενο του τέταρτου κεφαλαίου ***Open Card Framework*** είναι η περιγραφή του πλαισίου εργασίας Open Card Framework το οποίο παρέχει ένα σύνολο από κλάσεις και μηχανισμούς απαραίτητους για την ανάπτυξη εφαρμογών αλληλεπίδρασης δικτύων υπολογιστών, web browsers και άλλων πλατφόρμων οι οποίες τρέχουν java, με έξυπνες κάρτες. Σκοπός της ανάπτυξης ενός τέτοιου ανοικτού πλαισίου εργασίας είναι η μείωση της εξάρτησης που υπάρχει μεταξύ των οντοτήτων που εμπλέκονται στην ανάπτυξη μιας εφαρμογής έξυπνων καρτών δηλαδή μεταξύ εκδοτών, application service providers, card operating system providers και card terminal providers. Στα περιεχόμενα του κεφαλαίου περιλαμβάνονται η περιγραφή της αρχιτεκτονικής του OCF η οποία χωρίζεται σε δύο επίπεδα τα CardTerminal και το CardService καθώς και των βασικών τους συστατικών.

Στο πέμπτο κεφάλαιο ***Java Card Technology*** παρουσιάζεται η τεχνολογία Java Card η οποία παρέχει τα εργαλεία και τις προδιαγραφές για την ανάπτυξη Java Card εφαρμογών. Περιγράφεται η αρχιτεκτονική του συστήματος μιας Java Card και τα συστατικά της Java Card πλατφόρμας που είναι η Java Card Virtual Machine, το Java Card runtime Environment και το Java Card API καθώς και τα συστατικά μιας ολοκληρωμένης java card εφαρμογής την οποία συνθέτουν ένα Back End σύστημα, η host εφαρμογή, η συσκευή υποδοχής κάρτας και τα applets τα οποία τρέχουν στην κάρτα. Παρουσιάζονται επίσης οι τρόποι επικοινωνίας της java card με την host εφαρμογή. Επιπλέον περιγράφεται η Java Card Virtual Machine η οποία αποτελείται από δύο τμήματα τον converter ο οποίος παράγει αρχεία .cap ώστε αυτά να μπορούν να φορτωθούν στην κάρτα και τον interpreter. Εν συνεχεία περιγράφονται τα άλλα δύο συστατικά της Java Card πλατφόρμας που είναι το Java Card Runtime Environment αρμοδιότητες του οποίου είναι η διαχείριση των πόρων της κάρτας, οι δικτυακές επικοινωνίες, η εκτέλεση και ασφάλεια των applets και τα Java Card APIs τα οποία περιλαμβάνουν ένα σύνολο ειδικών κλάσεων που χρησιμοποιούνται για τον προγραμματισμό εφαρμογών έξυπνων καρτών. Τέλος το κεφάλαιο ασχολείται με τα Java Card Applets τη διαδικασία ανάπτυξης τους, τη διάρκεια ζωής τους και τους μηχανισμούς ασφάλειας τους.

Το έκτο κεφάλαιο ***Πειραματική υλοποίηση αυθεντικοποίησης με έξυπνες κάρτες*** περιγράφει την έξυπνη φοιτητική κάρτα, τα βασικά της χαρακτηριστικά και τα δεδομένα τα οποία πρέπει να περιλαμβάνει, παρουσιάζει τα πλεονεκτήματα ανάπτυξης εφαρμογών με τη χρήση του Java Card Framework τα βασικότερα από τα οποία είναι η εύκολη ανάπτυξη εφαρμογών, η ασφάλεια, η ανεξαρτησία από το hardware που χρησιμοποιείται, η δυνατότητα συνύπαρξης και διαχείρισης πολλαπλών εφαρμογών και η συμβατότητα με τα ήδη υπάρχοντα πρότυπα έξυπνων καρτών. Εν συνεχεία παρουσιάζεται το σενάριο υλοποίησης μιας ολοκληρωμένης εφαρμογής έξυπνων καρτών σύμφωνα με το οποίο θα παρέχονται υπηρεσίες στους φοιτητές με χρήση της φοιτητικής τους έξυπνης κάρτας. Οι υπηρεσίες αυτές είναι : αυθεντικοποίηση των χρηστών για απόκτηση πρόσβασης στα PC των εργαστηρίων, πρόσβαση στους φυσικούς χώρους του πανεπιστημίου με χρήση της έξυπνης κάρτας, πρόσβαση στις βάσεις δεδομένων της γραμματείας (interactive λειτουργίες με τις γραμματείες), πρόσβαση σε δίκτυα όπως για παράδειγμα στο internet και τέλος πρόσβαση στις βιβλιοθήκες του πανεπιστημίου. Επιπλέον περιγράφονται τα χαρακτηριστικά του περιβάλλοντος δηλαδή τα πληροφορικά συστήματα του τμήματος και στα οποία έχουν πρόσβαση οι χρήστες καθώς και ο τρόπος με τον οποίο γίνεται η αυθεντικοποίηση των χρηστών. Το κεφάλαιο τελειώνει με την παρουσίαση του λογισμικού και των εργαλείων που έχουν χρησιμοποιηθεί για την πειραματική υλοποίηση της αυθεντικοποίησης και την απεικόνιση της διαδικασίας της αυθεντικοποίησης.

Το τελευταίο κεφάλαιο ***Συμπεράσματα και Μελλοντικές Επεκτάσεις*** αποτελεί τον επίλογο της εργασίας, ενώ εκτός από τα συμπεράσματα που εξάγονται περιλαμβάνει και τον σχεδιασμό ενός Java Card Student Applet ως μια μελλοντική επέκταση της πειραματικής υλοποίησης αυθεντικοποίησης που έλαβε χώρα στο έκτο κεφάλαιο.

Κεφάλαιο 2

Μέθοδοι Αυθεντικοποίησης

Το παρόν κεφάλαιο επιχειρεί μια συνοπτική εισαγωγή στην έννοια της αυθεντικοποίησης και στην ανάγκη που υπάρχει για ανάπτυξη αξιόπιστων συστημάτων αυθεντικοποίησης των χρηστών κάθε πληροφοριακού συστήματος, έτσι ώστε οι μηχανισμοί ασφάλειας του να λειτουργούν σωστά. Περιγράφονται τα διάφορα μοντέλα και τεχνικές αυθεντικοποίησης που έχουν αναπτυχθεί, τα μειονεκτήματα και τα προβλήματα που αυτά παρουσιάζουν, η χρήση των έξυπνων καρτών ως μια από τις πιο αξιόπιστες μεθόδους αυθεντικοποίησης και τα πλεονεκτήματα που αυτή παρέχει έναντι των άλλων μεθόδων.

2.1 Αυθεντικοποίηση

Με τον όρο αυθεντικοποίηση καλούμε τη διαδικασία σύμφωνα με την οποία επιβεβαιώνεται ο ισχυρισμός της ταυτότητας ενός χρήστη. Αποτελεί ένα από τα βασικότερα χαρακτηριστικά κάθε υπολογιστικού συστήματος και αποσκοπεί στην αποτροπή της μη εξουσιοδοτημένης χρήσης των δεδομένων του συστήματος ενώ επιτρέπει στους χρήστες που έχουν αποδείξει τη γνησιότητα της ταυτότητας τους, να χρησιμοποιούν το σύστημα και να προσπελαύνουν τα δεδομένα που τους αφορούν πάντα σύμφωνα με ένα προκαθορισμένο τρόπο.
Η εποχή μας χαρακτηρίζεται από την ευρεία χρήση των νέων τεχνολογιών και υπηρεσιών που προσφέρουν οι κάθε είδους υπολογιστικές συσκευές. Ο άνθρωπος επιθυμεί να παράγει και να ανταλλάσσει πληροφορία οπουδήποτε και οποτεδήποτε γεγονός που επιβάλει την αντιπροσώπευση φυσικών οντοτήτων σε συστήματα ηλεκτρονικών συναλλαγών, με κάποιο μέσο όπως είναι ένα κλειδί ή ένας κωδικός ή μια έξυπνη κάρτα κ.τ.λ. . Η αυξανόμενη ανάγκη για μεταφορά και χρήση πληροφοριών μέσω συσκευών όπως είναι το κινητό τηλέφωνο, οι ηλεκτρονικές ατζέντες και οι διάφορες κάρτες εισάγουν ζητήματα ασφάλειας και προστασίας των ιδιωτικών δεδομένων του κατόχου. Η ανάπτυξη λοιπόν συστημάτων που θα επιτρέπουν την πρόσβαση σε εξουσιοδοτημένα μόνο άτομα είναι αναπόφευκτη ανάγκη.
Η ανάπτυξη τεχνικών και συστημάτων αυθεντικοποίησης των χρηστών πληροφοριακών συστημάτων, έχει ακριβώς σαν στόχο την αντιμετώπιση της απάτης, του ηλεκτρονικού εγκλήματος και της παράνομης πρόσβασης σε προσωπικές πληροφορίες. Στην επόμενη παράγραφο μελετώνται τα διάφορα μοντέλα και τεχνολογίες αυθεντικοποίησης.

2.2 Μοντέλα Αυθεντικοποίησης

Τα βασικότερα μοντέλα αυθεντικοποίησης χρήστη μπορούν να διαχωριστούν στις δύο παρακάτω κατηγορίες :

- Αυθεντικοποίηση από στοιχεία που έχει ο χρήστης
- Αυθεντικοποίηση από χαρακτηριστικά του ίδιου του χρήστη

2.2.1 Αυθεντικοποίηση από κάτι που έχει ο χρήστης

Σε αυτή την κατηγορία ανήκουν τα:

Συνθηματικά (passwords)/ Προσωπικοί αριθμοί αναγνώρισης:

Μέθοδος που χρησιμοποιείται για λογαριασμούς σε υπολογιστικά συστήματα αλλά και για τραπεζικές συναλλαγές Η χρήση τους είναι ο πιο διαδεδομένος και εύκολος τρόπος αυθεντικοποίησης των χρηστών χωρίς αυτό σε καμία περίπτωση να συνεπάγεται ότι είναι και ο πιο ασφαλής. Κάθε χρήστης προμηθεύεται ένα όνομα χρήστη (user name) και ένα συνθηματικό (password) το οποίο πρέπει να εισάγεται κάθε φορά που ο χρήστης θέλει να αποκτήσει πρόσβαση στο σύστημα. Το σύστημα αυθεντικοποίησης επιτρέπει την πρόσβαση στο σύστημα αφού πρώτα ελέγξει πως το συνθηματικό που εισήγαγε ο χρήστης ταιριάζει πράγματι με το αντίστοιχο που βρίσκεται ήδη αποθηκευμένο στη βάση δεδομένων του συστήματος.
Καθώς στα περισσότερα υπολογιστικά συστήματα δίνεται το δικαίωμα στους χρήστες να ορίσουν τον προσωπικό κωδικό τους προκύπτουν προβλήματα ασφάλειας, αφού είναι αρκετά εύκολο για έναν επίδοξο εισβολέα να μαντέψει και να σπάσει αυτό το συνθηματικό, είτε γνωρίζοντας προσωπικές πληροφορίες για τον κάθε χρήστη και δεδομένου ότι η επιλογή των συνθηματικών των νόμιμων χρηστών βασίζεται σε αυτές, είτε χρησιμοποιώντας ειδικά πακέτα λογισμικού που έχουν σχεδιασθεί ακριβώς για να εντοπίζουν κακώς επιλεγμένα συνθηματικά χρηστών. Τα πακέτα αυτά τυπικά υποστηρίζονται από τεράστια λεξιλόγια που περιέχουν συνηθισμένα συνθηματικά και η «απόδοση» τους είναι μεγάλη αν αναλογιστούμε ότι μέσα σε μια μέρα μπορούν να βρεθούν πάνω από το 50% των συνθηματικών των χρηστών. Κοινός παρονομαστής όλων αυτών είναι ότι οι κωδικοί αν δεν έχει γίνει σωστή επιλογή από το χρήστη κατά τον ορισμό τους μπορούν να αποκαλυφθούν πολύ εύκολα και να διαρρεύσουν έτσι προσωπικά δεδομένα με όλα τα αρνητικά επακόλουθα που αυτό συνεπάγεται. Επίσης ένα άλλο πρόβλημα που προκύπτει αφορά στην αποθήκευση των κωδικών σε κάποιον υπολογιστή. Αν λοιπόν τα συνθηματικά αποθηκεύονται σε μια μορφή χωρίς καθόλου προστασία ή χωρίς τη χρήση κρυπτογραφικών αλγορίθμων τότε τουλάχιστον ένα πρόσωπο, ο διαχειριστής του υπολογιστικού συστήματος γνωρίζει τον κωδικό και φυσικά μπορεί να τον χρησιμοποιήσει.

Συνθηματικά μια χρήσης:

Είναι μια παραλλαγή της παραπάνω μεθόδου. Στην περίπτωση αυτή ο χρήστης έχει στη διάθεση του, μια λίστα από κωδικούς τους οποίους μπορεί να χρησιμοποιεί μέχρι να εξαντληθεί η λίστα. Καθένας από τους κωδικούς μπορεί να χρησιμοποιείται για μια και μοναδική φορά.
Η μέθοδος αυτή υπερτερεί σε επίπεδο ασφάλειας, σε σύγκριση με την τεχνική των απλών συνθηματικών, καθώς το συνθηματικό του κάθε χρήστη αλλάζει τακτικά και κάθε πιθανός εισβολέας θα πρέπει να έχει στη διάθεση του ολόκληρη τη λίστα για να αποκτήσει μη νόμιμη πρόσβαση. Προβλήματα με αυτή τη μέθόδο μπορούν να προκύψουν αν η διανομή των λιστών στους νόμιμους κατόχους γίνει με μη ασφαλή και αναξιόπιστο τρόπο ή τέλος αν η λίστα αυτή κλαπεί.

Συστήματα πρόκλησης-απόκρισης:

Ένα σύστημα πρόκλησης - απόκρισης απαιτεί από τον χρήστη να κατέχει ένα μυστικό συνθηματικό P και να διαθέτει τα αναγκαία μέσα για τον υπολογισμό μιας μονόδρομης συνάρτησης f.
Όταν ο χρήστης ζητά πρόσβαση σε ένα σύστημα τότε πρέπει αρχικά να παρέχει το όνομα του (user name). Από την άλλη πλευρά ο υπολογιστής (host) αποκρίνεται στέλνοντας μια τυχαία πρόκληση R. Εν συνεχεία ο χρήστης αποκρίνεται στέλνοντας το αποτέλεσμα της μονόδρομης συνάρτησης που προκύπτει με το συνδυασμό των τιμών R και P, δηλαδή το αποτέλεσμα f(R,P). Το σύστημα με τη σειρά του εκτελεί και εκείνο τον ίδιο υπολογισμό και έτσι συγκρίνοντας τα αποτελέσματα αποφασίζει αν ο χρήστης θα γίνει αποδεκτός ή όχι.
Τα συνθηματικά των χρηστών αποθηκεύονται σε ένα φυσικά ασφαλές υποσύστημα ώστε να απαγορεύεται κάθε μη εξουσιοδοτημένη προσπέλαση του αρχείου των συνθηματικών. Ωστόσο η ασφάλεια της μεθόδου έγκειται στο κατά πόσο η μονόδρομη συνάρτηση f υποστηρίζει την ιδιότητα της μη γνωστοποίησης του συνθηματικού P(τουλάχιστον σε ένα εύλογο χρονικό διάστημα), ακόμα και αν οι τιμές τόσο του αποτελέσματος f(R,P) όσο και των τιμών R,f είναι γνωστές. Αυτό θα πρέπει να ισχύει ακόμα και στην περίπτωση όπου κάποιος υποκλοπέας έχει στην διάθεση του μια σειρά από τέτοια πιθανά συνθηματικά. Αν το σύνολο αυτών των πιθανών συνθηματικών είναι μικρό τότε το σύστημα πρόκλησης – απόκρισης μπορεί να θεωρηθεί πραγματικά ανασφαλές. Στο ίδιο συμπέρασμα θα καταλήξουμε ακόμα και αν σκεφτούμε ότι αν υπάρχουν πολύ μικρές πιθανότητες το P, που έχει στα χέρια ο υποκλοπέας, να είναι σωστό, αυτός μπορεί να δοκιμάσει όλα τα πιθανά συνθηματικά μέχρι να βρεθεί κάποιο που να του δώσει το σωστό αποτέλεσμα όταν σαν είσοδος της f δοθεί η υποκλεμμένη πρόκληση και το συνθηματικό που μάντεψε ο υποκλοπέας.

Συστήματα με βάση κουπόνια – Μαγνητικές κάρτες :

Η χρήση της είναι αρκετά διαδεδομένη σήμερα καθώς χρησιμοποιείται ευρέως για την αναγνώριση χρηστών σε αυτόματα μηχανήματα αναλήψεων ATM, σε σημεία πωλήσεων, σε καταστήματα και σε κτίρια για έλεγχο πρόσβασης.
Οι μαγνητικές κάρτες διαθέτουν μαγνητική ταινία στην οποία αποθηκεύονται οι διάφορες πληροφορίες (π.χ. αριθμός τραπεζικού λογαριασμού). Η κάρτα αυτή

χρησιμοποιείται πάντα σε συνδυασμό με έναν προσωπικό αναγνωριστικό αριθμό (PIN) ώστε να επιβεβαιώνεται η ταυτότητα του νόμιμου κατόχου.
Υπάρχουν δύο κατηγορίες συστημάτων επιβεβαίωσης του PIN:

- Off – line: Σε συστήματα off – line, το PIN αποθηκεύεται σε κρυπτογραφημένη μορφή στην κάρτα (είναι συνήθως το αποτέλεσμα της εφαρμογής μιας μονόδρομης κρυπτογράφησης πάνω στο PIN). Πριν την κρυπτογράφηση του, το PIN πρέπει να συνδυάζεται με μια πληροφορία που εξαρτάται από τον κάτοχο της κάρτας. Αυτός θα μπορούσε να είναι για παράδειγμα ο αριθμός ταυτότητας ώστε οι επιτιθέμενοι να δυσκολεύονται να συγκρίνουν λίστες κρυπτογραφημένων PINs.

- On – line: Σε συστήματα on – line τα PINs των χρηστών επιβεβαιώνονται κεντρικά και δεν χρειάζεται να γράφονται πάνω στην κάρτα.

Συστήματα με βάση κουπόνια – Έξυπνες κάρτες :

Οι έξυπνες κάρτες μοιάζουν εξωτερικά με τις κοινές πλαστικές κάρτες που αναφέραμε παραπάνω. Η ειδοποιός διαφορά είναι ότι οι έξυπνες κάρτες δεν διαθέτουν απλή μαγνητική ταινία αλλά ενσωματώνουν ένα μικροεπεξεργαστή ο οποίος μπορεί να χειριστεί τις πληροφορίες που βρίσκονται αποθηκευμένες στην μνήμη της κάρτας, να επεξεργάζεται τα δεδομένα δηλαδή να εκτελεί αριθμητικές ή και πιο σύνθετες πράξεις, έχει τη δυνατότητα επικοινωνίας μέσω μια θύρας εισόδου /εξόδου (I/O port) και για τον λόγο αυτό μπορεί να υλοποιεί αλγορίθμους κρυπτογράφησης και αυθεντικοποίησης αυξάνοντας έτσι την ασφάλεια που παρέχει η κάρτα.
Η πρώτη γενιά έξυπνων καρτών διέθετε απλούς επεξεργαστές των 8- bit και μνήμη των 8 Kbytes. Μερικές διέθεταν περιορισμένες ενσωματωμένες λειτουργίες κρυπτογράφησης. Η δεύτερη γενιά έξυπνων καρτών παρέχει πιο δυνατούς επεξεργαστές, περισσότερη μνήμη και μια ποικιλία κρυπτογραφικών λειτουργιών. Μερικά μάλιστα από τα τελευταία μοντέλα μπορούν να εκτελούν υπολογισμούς ψηφιακών υπογραφών σε κλάσματα του δευτερολέπτου.
Η διαδικασία της αυθεντικοποίησης, στην περίπτωση που το συνθηματικό του χρήστη είναι αποθηκευμένο στην κάρτα μπορούμε να πούμε ότι μοιάζει με το σχήμα αυθεντικοποίησης με το μοντέλο πρόκλησης – απόκρισης υπολογίζοντας μια επιλεγμένη μονόδρομη συνάρτηση και παράγοντας ένα μήνυμα αυθεντικοποίησης. Η διαδικασία ξεκινά από τη στιγμή που ο χρήστης θα εισάγει την κάρτα του στο τερματικό ανάγνωσης για να αποκτήσει πρόσβαση σε κάποιες υπηρεσίες ή δεδομένα. Ο κεντρικός εξυπηρετητής στον οποίο είναι μόνιμα εγκατεστημένη η μητρική έξυπνη κάρτα (δηλαδή εκεί όπου αποθηκεύεται ο μητρικός μυστικός κωδικός που εγγράφεται στην κάρτα κατά τη διαδικασία της προσωποποίησης της κάρτας) ζητά από την έξυπνη κάρτα του χρήστη τη δημιουργία και αποστολή ενός μηνύματος αυθεντικοποίησης. Για τη δημιουργία του μηνύματος αυθεντικοποίησης χρησιμοποιείται μεταξύ των άλλων, ένας τυχαίος αριθμός και ο σειριακός αριθμός της κάρτας του χρήστη. Το ίδιο μήνυμα αυθεντικοποίησης υπολογίζεται και από την μητρική έξυπνη κάρτα και τα δύο μηνύματα συγκρίνονται. Αν είναι τα ίδια τότε διασφαλίζεται το γεγονός ότι η κάρτα του χρήστη είναι γνήσια και του επιτρέπεται η πρόσβαση.

Μια βελτιωμένη εκδοχή του παραπάνω σχήματος θα απαιτούσε από το χρήστη να εισάγει τον αριθμό PIN μέσω του τερματικού εξοπλισμού ή της συσκευής ανάγνωσης της κάρτας) πριν η κάρτα εκτελέσει την λειτουργία της, για προστασία σε περίπτωση κλοπής.
Επιπλέον το επίπεδο ασφάλειας αυξάνεται ακόμη περισσότερο στην περίπτωση που η διαδικασία της αυθεντικοποίησης συνδυαστεί με την τεχνολογία των biometrics. Στην περίπτωση αυτή η επαλήθευση του ιδιοκτήτη της κάρτας γίνεται μέσω του δακτυλικού του αποτυπώματος ή μέσω ανάγνωσης της ίριδας του ματιού του. Ο χρήστης δηλαδή τοποθετεί το κατάλληλο μέρος του σώματος του στη συσκευή ανάγνωσης και αν τα δεδομένα που συλλέγονται από αυτή συμπίπτουν με τα δεδομένα της κάρτας τότε γίνεται και η πιστοποίηση του χρήστη. Έτσι ακόμη και σε περίπτωση κλοπής της κάρτας με την τεχνολογία των biometrics μπορεί να διαπιστωθεί αν ο πραγματικός ιδιοκτήτης της κάρτας είναι αυτός που την κατέχει την συγκεκριμένη στιγμή.
Η αξιοποίηση της τεχνολογίας των έξυπνων καρτών παρουσιάζει ολοένα και μεγαλύτερη δυναμική λόγω της προόδου στις τεχνολογίες κατασκευής ολοκληρωμένων κυκλωμάτων και λόγω της αύξησης περιστατικών απάτης σε συγκεκριμένους τομείς εφαρμογών, δεδομένου φυσικά του αυξημένου επιπέδου ασφαλείας που παρέχει η τεχνική αυθεντικοποίησης με έξυπνες κάρτες . Έτσι πληθώρα οργανισμών σε όλο τον κόσμο υιοθετούν τη συγκεκριμένη τεχνολογία σε διάφορες εφαρμογές όπως: ηλεκτρονικό πορτοφόλι, κινητά τηλέφωνα, κάρτα ασθενούς, έλεγχος πρόσβασης σε δίκτυα, έλεγχος φυσικής πρόσβασης σε εγκαταστάσεις, κ.τ.λ.

2.2.2 Αυθεντικοποίηση από χαρακτηριστικά του χρήστη

Κάποιες από τις πιο γνωστές βιομετρικές τεχνικές για αναγνώριση της ταυτότητας είναι:

- Δακτυλικό αποτύπωμα
- Αναγνώριση φωνής
- Αναγνώριση μορφολογίας του αμφιβληστροειδούς χιτώνα
- Ίριδα
- Γεωμετρία παλάμης

Δακτυλικό αποτύπωμα :

Ένας πρώτος τρόπος αυθεντικοποίησης με τη χρήση των βιομετρικών χαρακτηριστικών είναι η επιβεβαίωση μέσω του δακτυλικού αποτυπώματος του χρήστη. Είναι γνωστό ότι τα σχέδια των γραμμώσεων του δέρματος των δακτύλων μπορούν να χρησιμοποιηθούν για τον μονοσήμαντο προσδιορισμό των ανθρώπων. Εκτός από τις γραμμώσεις, κάποια άλλα κύρια χαρακτηριστικά των δακτύλων είναι τα σημεία εκκίνησης, διακλάδωσης και διασταύρωσης. Για να γίνει η αυθεντικοποίηση θα πρέπει να έχουν ήδη αποθηκευτεί δείγματα δακτυλικών αποτυπωμάτων έτσι ώστε να γίνεται κάθε φορά η σύγκριση. Τέτοια συστήματα βασίζονται στην αναγνώριση διαφόρων τύπων "μορφών" δακτυλικών αποτυπωμάτων, οι πιο γνωστές από τις οποίες είναι το τόξο, ο βρόχος και η σπείρα.

Η μέτρηση της δομής του δακτυλικού αποτυπώματος του χρήστη γίνεται με οπτικούς ή ηλεκτρικούς αισθητήρες. Κάθε δάκτυλο διαθέτει τουλάχιστον μια τέτοια βασική μορφή και κάποιες άλλες δευτερεύουσες μορφές 50 έως 200 σε κάθε ανθρώπινο δάκτυλο και μάλιστα ιδιαίτερα σημαντικές για την μονοσήμαντη αναγνώριση του κάθε ανθρώπου.
Στα βασικά πλεονεκτήματα της μεθόδου συγκαταλέγονται η μεγάλη ακρίβεια και το χαμηλό κόστος και μικρό μέγεθος των αισθητήρων. Ωστόσο η τεχνική αυτή μπορεί να επηρεαστεί από κάποιους περιβαλλοντικούς παράγοντες όπως είναι η σκόνη, ο ιδρώτας, η πίεση του δακτύλου και να δώσει λανθασμένα αποτελέσματα.

Αναγνώριση φωνής :

Στα συστήματα αυθεντικοποίησης με χρήση της ανθρώπινης φωνής, θα πρέπει αρχικά να αποτυπωθεί η φωνή του κάθε εξουσιοδοτημένου χρήστη έτσι ώστε να μπορεί να γίνει η σύγκριση. Έτσι θα πρέπει να καταγραφεί ένα σύνολο από προτάσεις ειπωμένες από το χρήστη, φυσικά μέσα σε ένα περιβάλλον απόλυτα ελεγχόμενο για την ανάπτυξη ενός όσο το δυνατόν αξιόπιστου συστήματος αναγνώρισης φωνής. Στη συνέχεια χρησιμοποιούνται τεχνικές ανάλυσης Fourier για την μέτρηση των χαρακτηριστικών του φάσματος συχνοτήτων του φωνητικού δείγματος.
Πρόκειται για συστήματα που είναι αρκετά απλά στους χρήστες, δεν χρειάζεται η απομνημόνευση κάποιου κωδικού, ωστόσο όμως παρουσιάζουν και αρκετά μειονεκτήματα. Έτσι απαιτείται αρκετή ώρα για να ειπωθεί και να επεξεργαστεί κατάλληλα το απαιτούμενο φωνητικό δείγμα ενώ η φωνή μπορεί να μεταβάλλεται καθώς εξαρτάται από τη φυσική κατάσταση του χρήστη, όπως συμβαίνει για παράδειγμα σε περιπτώσεις κρυολογήματος οπότε και υπάρχει μεγάλη πιθανότητα αποτυχίας του συστήματος. Ένα άλλο πρόβλημα είναι η αντιγραφή της φωνής του χρήστη με ένα απλό μαγνητόφωνο, ειδικά σε περίπτωση που το σύνολο των προτάσεων που απαιτεί το σύστημα να ειπωθούν από το χρήστη δεν είναι τυχαία επιλεγμένο.

Αναγνώριση μορφολογίας του αμφιβληστροειδούς χιτώνα :

Χρησιμοποιείται η μορφολογία των αγγείων αίματος στον αμφιβληστροειδή χιτώνα του ματιού, καθώς η μορφολογία αυτή αποτελεί ένα τέλειο χαρακτηριστικό για την ταυτότητα του κάθε ατόμου. Η αυθεντικοποίηση των χρηστών γίνεται με την ακόλουθη διαδικασία :

- ✓ Κοίταγμα του χρήστη μέσα από ένα προσοφθάλμιο
- ✓ Εστίαση σε ένα πλέγμα και
- ✓ Πίεση ενός πλήκτρου
- ✓ Η μηχανή σκανάρει το μάτι με μια ακτίνα υπέρυθρου φωτός και χαμηλής εντάσεως.

Επειδή το παραπάνω σύστημα είναι πολύ κουραστικό για τους χρήστες, οι οποίοι μάλιστα αντιδρούν σε τέτοιου είδους συστήματα από φόβο μήπως πάθουν κάτι τα μάτια τους, τείνει να αντικατασταθεί από συστήματα που δεν απαιτούν τη χρήση του προσοφθαλμίου αλλά έχουν τη δυνατότητα να διαβάζουν τη μορφολογία του αμφιβληστροειδούς χιτώνα από την απόσταση ενός μέτρου.

Ανάλυση Ίριδας

Η ίριδα αποτελεί εξαιρετικά πλούσιο σε πληροφορία μέρος του ματιού, που δημιουργείται μέχρι τον όγδοο μήνα της κύησης. Το κύριο χαρακτηριστικό της είναι η ακτινωτή δομή, ενώ υπάρχουν και άλλα βοηθητικά χαρακτηριστικά όπως είναι τα αυλάκια, τα στίγματα και η κορώνα. Η αναγνώριση γίνεται με τη χρήση μιας κάμερας η οποία παίρνει μιας υψηλής ευκρίνειας εικόνα της ίριδας του ματιού. Στα πλεονεκτήματα της μεθόδου περιλαμβάνονται τα:

- ✓ Μοναδικότητα της ίριδας ακόμα και ανάμεσα σε δίδυμα αδέλφια
- ✓ Η ίριδα είναι χαρακτηριστικό αμετάβλητο κατά τη διάρκεια της ζωής του ανθρώπου.
- ✓ Παρέχει υψηλή ακρίβεια

Το σημαντικότερο μειονέκτημα της μεθόδου είναι το μεγάλο της κόστος .

Γεωμετρία παλάμης

Χρησιμοποιείται ένας οπτικός αισθητήρας για να μετρηθούν τα χαρακτηριστικά της παλάμης του χρήστη.

Πλεονεκτήματα:

- ✓ Τα χαρακτηριστικά περιγράφονται πολύ περιεκτικά
- ✓ Γρήγορη επιβεβαίωση
- ✓ Υψηλή αποδοχή από τους χρήστες

Μειονεκτήματα:

- ✓ Η μέθοδος αυτή χρησιμοποιείται μόνο για επιβεβαίωση ταυτότητας
- ✓ Οι συσκευές ανάγνωσης της παλάμης είναι αρκετά ογκώδεις και υψηλού κόστους.

2.3 Πλεονεκτήματα Έξυπνων Καρτών

Η αξιοποίηση της τεχνολογίας των έξυπνων καρτών παρουσιάζει ολοένα και μεγαλύτερη δυναμική καθώς οι τεχνολογίες κατασκευής ολοκληρωμένων κυκλωμάτων εμφανίζουν αξιοσημείωτη πρόοδο γεγονός που σημαίνει ότι οι έξυπνες κάρτες μπορούν να γίνουν εξυπνότερες με τις βελτιώσεις του chip. Παράλληλα περιστατικά απάτης αυξάνονται συνεχώς με αποτέλεσμα να απαιτείται μια όσο το δυνατόν πιο αξιόπιστη και ασφαλής μέθοδος αυθεντικοποίησης. Οι έξυπνες κάρτες είναι μια τεχνολογία που μπορεί να δώσει μια αξιοπρεπή λύση σε αυτό το πρόβλημα αφού διαθέτουν μικροεπεξεργαστή, μπορούν να αποθηκεύουν και να επεξεργάζονται τα δεδομένα, διαθέτουν πολύ μεγάλη μνήμη και παρέχουν ένα υψηλό επίπεδο ασφάλειας. Στα σημαντικότερα πλεονεκτήματα των έξυπνων καρτών περιλαμβάνονται τα ακόλουθα:

Ασφάλεια σε επίπεδο υλικού

Ένας από τους στόχους σχεδίασης του ολοκληρωμένου μιας έξυπνης κάρτας είναι να παρέχει ασφάλεια στα δεδομένα που είναι αποθηκευμένα στο εσωτερικό της σε φυσικό επίπεδο. Έτσι ο επεξεργαστής και η μνήμη συνδυάζονται στο ίδιο κύκλωμα έτσι ώστε να είναι δύσκολη η υποκλοπή των σημάτων που ανταλλάσσονται μεταξύ των υπομονάδων. Ενσωματώνονται στρώματα προστασίας των ημιαγωγών με επικάλυψη ώστε να αποφευχθεί η περαιτέρω ανάλυση του ολοκληρωμένου. Οι γραμμές διεύθυνσης και τα κελιά της μνήμης διατάσσονται σε περίεργα μοτίβα για να μην είναι εύκολη η φυσική τους εξέταση. Επιπλέον μερικά ολοκληρωμένα έχουν την δυνατότητα να αντιλαμβάνονται αν έχει αφαιρεθεί κάποιο στρώμα πάνω από αυτά (όταν κάποιος προσπαθεί να το εξετάσει) και μπορούν να ανιχνεύουν ασυνήθιστες αλλαγές στην τάση τροφοδοσίας ή στους παλμούς του ρολογιού, διακόπτοντας σε κάθε περίπτωση τη λειτουργία τους και αποτρέποντας έτσι μια μη εξουσιοδοτημένη πρόσβαση.

Αξιόπιστη μέθοδος αυθεντικοποίησης

Για να γίνει η αυθεντικοποίηση του κατόχου της έξυπνης κάρτας θα πρέπει πρώτα να πληκτρολογηθεί το PIN, το οποίο εν συνεχεία συγκρίνεται με τον κωδικό που υπάρχει αποθηκευμένο στη μνήμη της κάρτας. Η συγκεκριμένη διαδικασία λαμβάνει χώρα στο εσωτερικό της κάρτας και συνεπώς με μεγαλύτερη ασφάλεια σε αντίθεση με άλλες τεχνολογίες που αναφέρθηκαν όπως για παράδειγμα οι μαγνητικές κάρτες όπου η σύγκριση γίνεται εξωτερικά αφού έχει ανακληθεί από τη μαγνητική κάρτα ο προσωπικός κωδικός του κατόχου.
Ακόμα και στην περίπτωση όπου ο κωδικός που ανακαλείται από την μαγνητική κάρτα είναι κρυπτογραφημένος, θα έπρεπε το τερματικό ανάγνωσης των καρτών να μπορεί να τον αποκρυπτογραφήσει, γεγονός βέβαια που δημιουργεί επιπρόσθετους κινδύνους καθώς ο αλγόριθμος κρυπτογράφησης μπορεί να αξιοποιηθεί για την ανεύρεση κωδικών κλεμμένων καρτών. Τέτοιους κινδύνους δεν διατρέχουν οι έξυπνες κάρτες δεδομένου ότι η διαδικασία εκτελείται εσωτερικά χωρίς να απαιτείται από το τερματικό ανάγνωσης να εκτελεί κάποιο αλγόριθμοι κρυπτογράφησης.
Ένα άλλο θετικό στοιχείο των έξυπνων καρτών είναι ο μεγάλος αποθηκευτικός χώρος που διαθέτουν ο οποίος επιτρέπει την υλοποίηση μηχανισμών αυθεντικοποίησης που στηρίζονται σε βιομετρικά χαρακτηριστικά του κατόχου της κάρτας δημιουργώντας έτσι ένα σύστημα αυθεντικοποίησης με εξαιρετικά υψηλό επίπεδο ασφάλειας. Έτσι αντί να χρησιμοποιείται ο κωδικός PIN μπορεί να αποθηκευτεί στην έξυπνη κάρτα το δακτυλικό αποτύπωμα ή μια εικόνα από την ίριδα του ματιού του κατόχου και η αναγνώριση να γίνει με τη χρήση των κατάλληλων συσκευών ανάγνωσης των βιομετρικών χαρακτηριστικών.

Πολλαπλές εφαρμογές

Οι έξυπνες κάρτες μπορούν να προσφέρουν στους χρήστες πολλαπλές εφαρμογές σε μια μόνο κάρτα εξασφαλίζοντας και την ανεξαρτησία μεταξύ των εφαρμογών. Μερικά παραδείγματα όπου διαπιστώνουμε την πολλαπλότητα χρήσης των έξυπνων καρτών είναι στον τομέα των τηλεπικοινωνιών όπου εκτός από τις βασικές

υπηρεσίες παρέχονται επιπλέον web – browsing , ενημέρωση και παροχή πληροφοριών σε διάφορα θέματα, εμπόριο κ.α.

Ευκολία στη χρήση

Πολλοί έχουν χαρακτηρίσει τις έξυπνες κάρτες ως «υπολογιστές πορτοφολιού» [3] και όχι άδικα. Κρίσιμες πληροφορίες μπορούν να μεταφέρονται ανά πάσα στιγμή οπουδήποτε και οποτεδήποτε με εύκολο και ασφαλή τρόπο. Επιπλέον οι πληροφορίες και οι μίνι- εφαρμογές που περιέχονται στην κάρτα μπορούν να αναβαθμιστούν ή να αλλαχτούν χωρίς να απαιτείται η αλλαγή της κάρτας.

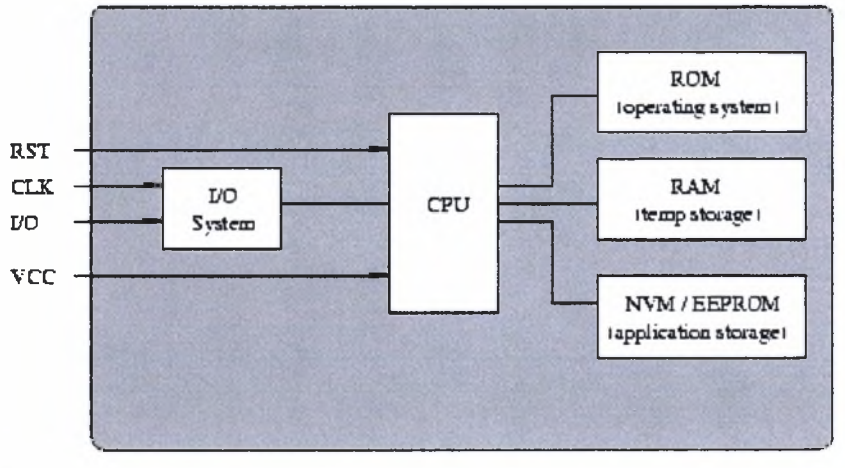
Κεφάλαιο 3

Τεχνολογία και Αρχιτεκτονική Έξυπνων καρτών

Στο κεφάλαιο Τεχνολογία και Αρχιτεκτονική Έξυπνων καρτών αναπτύσσονται τα ακόλουθα θέματα: Η αρχιτεκτονική των έξυπνων καρτών και τα συστατικά της δηλαδή η μονάδα επεξεργασίας, η μονάδα εισόδου / εξόδου και η μνήμη της, τα βήματα της διαδικασίας παραγωγής και έκδοσης της κάρτας. Αναλύεται το λειτουργικό σύστημα της κάρτας στο οποίο περιλαμβάνονται το σύστημα αρχείων της κάρτας, τα είδη των αρχείων που μπορούν να αποθηκεύονται σε αυτό, οι εντολές προσπέλασης τους και ο τρόπος με τον οποίο αυτά προστατεύονται από μη εξουσιοδοτημένη πρόσβαση. Περιγράφεται η διαδικασία αυθεντικοποίησης με την οποία διασφαλίζεται η γνησιότητα μιας έξυπνης κάρτας και ο τρόπος επικοινωνίας μεταξύ κάρτας και αναγνώστη με την χρήση πρωτοκόλλων επικοινωνίας τόσο σε επίπεδο σύνδεσης δεδομένων όσο και σε επίπεδο εφαρμογής. Τέλος γίνεται μια σύντομη αναφορά στα βασικότερα διεθνή και βιομηχανικά πρότυπα έξυπνων καρτών.

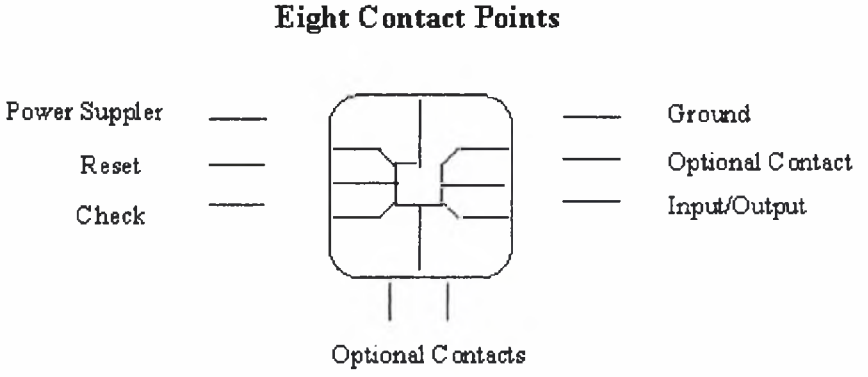
3.1 Αρχιτεκτονική Έξυπνων Καρτών

Τα βασικά συστατικά της αρχιτεκτονικής των έξυπνων καρτών όπως αυτά απεικονίζονται στην παρακάτω εικόνα είναι η κεντρική μονάδα επεξεργασίας (CPU), η μονάδα εισόδου / εξόδου και οι τρεις διαφορετικοί τύποι μνήμης (ROM, RAM, EEPROM).



3.1 Συστατικά του chip της έξυπνης κάρτας

Τα συστατικά αυτά τοποθετούνται όλα μαζί σε ένα chip ολοκληρωμένου κυκλώματος και όχι σε χωριστά chips συνδεδεμένα μεταξύ τους. Με τον τρόπο αυτό γίνεται δύσκολη η υποκλοπή των σημάτων που ανταλλάσσονται μεταξύ των υπομονάδων και έτσι εξασφαλίζεται η φυσική προστασία των δεδομένων που είναι αποθηκευμένα στην κάρτα.
Το ολοκληρωμένο κύκλωμα μιας έξυπνης κάρτας χρησιμοποιεί συνήθως 8 μεταλλικές επαφές από τις οποίες οι 2 μπορούν να παραλειφθούν από τους κατασκευαστές για λόγους κόστους και υφίστανται αποκλειστικά για μελλοντική χρήση.



3.2 Μεταλλικές επαφές Έξυπνης Κάρτας

V_{CC} (Supply voltage): Χρησιμοποιείται για την τροφοδοσία του επεξεργαστή της κάρτας με ενέργεια.
RST (Reset the microprocessor): Με την επαφή αυτή στέλνεται σήμα στον επεξεργαστή της κάρτας, ο οποίος θα αρχικοποιήσει τις εντολές επαναφοράς
CLK (Clock frequency): Απαραίτητη για τον καθορισμό της ταχύτητας λειτουργίας του επεξεργαστή αλλά και για το ρυθμό μεταφοράς των δεδομένων μεταξύ έξυπνης κάρτας και τερματικού.
GND (Ground)
V_{PP} (External programming voltage): Αποτελεί μια δεύτερη πηγή ενέργειας διαφορετική από την V_{CC}
I/O (Serial input/ output communications): Χρησιμοποιείται για την εγκατάσταση του καναλιού επικοινωνίας μεταξύ κάρτας και τερματικού.
RFU (Reserved for future use)

3.1.1 Κεντρική Μονάδα Επεξεργασίας έξυπνης Κάρτας

Η CPU του chip της έξυπνης κάρτας είναι ένας 8-bit επεξεργαστής με δυνατότητα εκτέλεσης 400,000 εντολών το δευτερόλεπτο. Το σύνολο των εντολών που χρησιμοποιείται αφορά στη διαχείριση της μνήμης και των καταχωρητών, στις λειτουργίες εισόδου εξόδου και στην διευθυνσιοδότηση. Ωστόσο οι κατασκευαστές των chip μπορούν να επεκτείνουν το σύνολο των εντολών προσθέτοντας επιπλέον εντολές . Τα τελευταίας τεχνολογίας chip έχουν μεγαλύτερη ταχύτητα και μπορούν να εκτελέσουν πάνω από 1 εκατομμύριο εντολές το δευτερόλεπτο. Ο χρόνος που

χρειάζεται μια έξυπνη κάρτα για την διενέργεια μιας συναλλαγής κυμαίνεται από 1 έως 3 δευτερόλεπτα. Στην περίπτωση όμως που χρειάζεται να εκτελεστεί κάποιος κρυπτογραφικός αλγόριθμος ο χρόνος αυτός είναι κατά πολύ μεγαλύτερος. Για παράδειγμα αν χρησιμοποιείται ο RSA με κλειδί των 1024 bits ο χρόνος εκτέλεσης είναι από 10 έως 20 δευτερόλεπτα. Για το λόγο αυτό στο chip μπορεί να περιλαμβάνεται και ένας co-processor για την επιτάχυνση των κρυπτογραφικών υπολογισμών. Αυτός είναι ένας δεύτερος επεξεργαστής με δυνατότητες γρήγορης εκτέλεσης πολύπλοκων αριθμητικών πράξεων μεταξύ ακεραίων, όπως για παράδειγμα η πράξη του πολλαπλασιασμού μεταξύ ακεραίων. Ωστόσο η προσθήκη ενός δεύτερου επεξεργαστή αυξάνει το μέγεθος του chip και φυσικά το κόστος του.

3.1.2 Μνήμη

Τα κύρια τμήματα μιας έξυπνης κάρτας είναι:

Μνήμη Εργασίας

Η μνήμη εργασίας (working memory-Random Access Memory) η οποία διατηρεί τα περιεχόμενα της μόνο κατά τη διάρκεια που η έξυπνη κάρτα τροφοδοτείται με ρεύμα και συνεπώς αξιοποιείται από τον μικροεπεξεργαστή για προσωρινή αποθήκευση δεδομένων. Το μέγεθος της κυμαίνεται από 256 bytes έως 1KByte

Μη διαγράψιμη μνήμη Rom

Η μη διαγράψιμη μνήμη ROM(Read Only Memory) δεν απαιτεί συνεχή τροφοδοσία για την διατήρηση των δεδομένων που έχουν αποθηκευθεί σε αυτή. Η συγκεκριμένη μνήμη χρησιμοποιείται για την αποθήκευση του λειτουργικού συστήματος της κάρτας μέσω του οποίου υποστηρίζονται οι λειτουργικές προδιαγραφές και οι μηχανισμοί ασφάλειας της κάρτας, όπως είναι η διαχείριση των μυστικών κλειδιών και κωδικών, εκτέλεση ρουτινών επικοινωνίας και η εκτέλεση κρυπτογραφικών αλγορίθμων. Το περιεχόμενο της δεν μπορεί να αλλάξει και οι πληροφορίες εγγράφονται από τον κατασκευαστή της έξυπνης κάρτας. Το μέγεθος της κυμαίνεται από 8 Kbytes έως 32Kbytes.

Μνήμη Εφαρμογών (EEPROM)

Η μνήμη εφαρμογών (EEPROM), αξιοποιείται για την εγγραφή, την ενημέρωση και διαγραφή των δεδομένων που είναι αποθηκευμένα στην κάρτα, καθ' όλη τη διάρκεια του κύκλου ζωής της. Συνήθως οργανώνεται σε λέξεις μήκους 32 bits και η συνολική της χωρητικότητα κυμαίνεται από 1 έως 64Kbytes.
Χωρίζεται σε ανεξάρτητα τμήματα το κάθε ένα από τα οποία αποθηκεύει συγκεκριμένες κατηγορίες δεδομένων, υλοποιώντας διαφορετικούς μηχανισμούς για τον έλεγχο πρόσβασης σε κάθε ένα από αυτά. Τα τμήματα στα οποία διαμερίζεται η μνήμη εφαρμογών είναι:

- ***Μυστική Περιοχή(Secrete Area)***: Η συγκεκριμένη περιοχή μνήμης μπορεί να εγγραφεί μόνο μια φορά, ενώ απαγορεύεται οποιαδήποτε προσπάθεια

προσπέλασης των δεδομένων. Τα δεδομένα που αποθηκεύονται στην περιοχή αυτή είναι τα διάφορα μυστικά κλειδιά και κωδικοί που αξιοποιούνται εσωτερικά από την κάρτα για την υλοποίηση των μηχανισμών ασφαλείας. Κάποιοι τύποι κλειδιών και κωδικών είναι:

- ***Κλειδί εργοστασίου(manufacture's key)***: Προστατεύει την έξυπνη κάρτα από τη στιγμή που ολοκληρώνεται η κατασκευή της μέχρι τη στιγμή που προσωποποιείται για κάποιον συγκεκριμένο χρήστη.
- ***Κύρια και δευτερεύοντα κλειδιά του εκδότη τη κάρτας(Primary Issuer Key PIK, and Co-Issuer key CIK)***: Εξασφαλίζουν την εμπιστευτικότητα και ακεραιότητα των δεδομένων κάποιας εφαρμογής από μη εξουσιοδοτημένες προσπάθειες προσπέλασης αυτών.
- ***Μυστικός προσωπικός κωδικός του κατόχου της κάρτας(Personal Identification Number-PIN)***: Περιορίζει τη χρήση της κάρτας από πρόσωπα εκτός του κατόχου της.
- ***Κλειδιά Κρυπτογράφησης(Encryption Keys)***: Τα κλειδιά αυτά χρησιμοποιούνται κατά την εκτέλεση των κρυπτογραφικών αλγόριθμων που υποστηρίζονται από την έξυπνη κάρτα. Αυτά μπορεί να είναι κλειδιά για την υποστήριξη συμμετρικής κρυπτογραφίας (π.χ. DES), ή να είναι το ιδιωτικό κλειδί του κατόχου της κάρτας στα πλαίσια υλοποίησης μηχανισμών ασύμμετρης κρυπτογραφίας (π.χ. RSA).

- ***Περιοχή Ιστορικού Πρόσβασης (Access Area)***: Στην περιοχή αυτή καταγράφονται αυτόματα όλες οι προσπάθειες πρόσβασης σε προστατευμένη πληροφορία ανεξαρτήτως αν ήταν επιτυχείς ή όχι. Σε περίπτωση επανειλημμένων προσπαθειών χρήσης λανθασμένων κωδικών (PINs) η κάρτα κλειδώνεται.

- ***Περιοχή Ελεύθερης Πρόσβασης (Public Area)*** : Όπως υποδηλώνει και το όνομα, η περιοχή αυτή αξιοποιείται για την αποθήκευση μη εμπιστευτικών δεδομένων και η προσπέλαση της δεν απαιτεί τη χρήση μυστικών κλειδιών ή κωδικών.

- ***Περιοχή Εργασίας (Work Area)***: Η συγκεκριμένη περιοχή είναι ίσως η πλέον σημαντική καθώς αξιοποιείται για την αποθήκευση δεδομένων από τις εφαρμογές. Ανάλογα με τα χαρακτηριστικά της εφαρμογής τα δεδομένα μπορεί να προστατεύονται σε ότι αφορά την ανάγνωση, ή/και την εγγραφή ή/και την διαγραφή τους μέσω διαφορετικών μυστικών κλειδιών.

3.1.3 Μονάδα Εισόδου/ Εξόδου

Η επικοινωνία με τον εξωτερικό κόσμο γίνεται μέσω μιας απλής θύρας εισόδου / εξόδου. Τα δεδομένα που μεταφέρονται ελέγχονται και προστατεύονται από τον επεξεργαστή. Αυτό μπορεί να γίνει με τη χρήση υψηλού επιπέδου πρωτοκόλλων επικοινωνίας (τα οποία αναλύονται παρακάτω) σύμφωνα με τα οποία ο επεξεργαστής φιλτράρει όλη την πληροφορία που ανταλλάσσεται μεταξύ των

συστατικών του chip. Αυτά τα πρωτόκολλα χρησιμοποιούνται επίσης για την αυθεντικοποίηση.

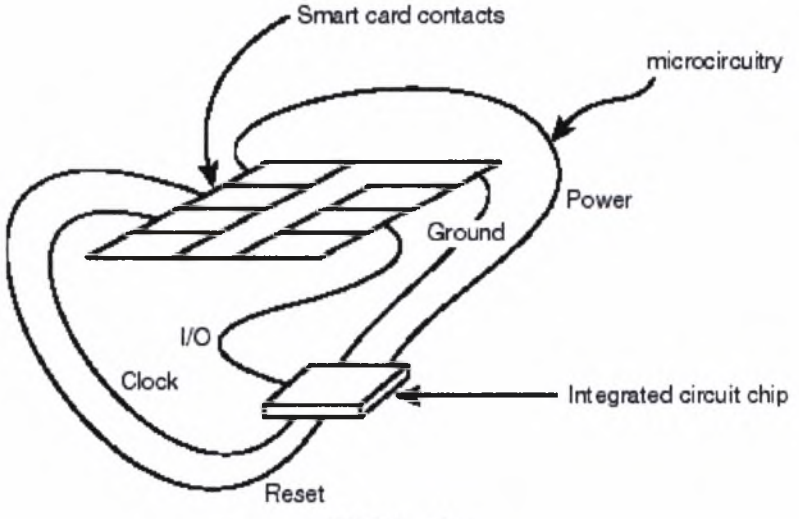
3.1.4 Interface Devices

Μια έξυπνη κάρτα δεν περιέχει κάποια ανεξάρτητη πηγή ενέργειας, η παρουσία της οποίας κρίνεται απαραίτητη για την τροφοδοσία του επεξεργαστή. Για το λόγο αυτό η έξυπνη κάρτα πρέπει να εισαχθεί σε μια συσκευή η οποία να παρέχει την απαιτούμενη ενέργεια. Αυτή καλείται συσκευή διεπαφής (interface device IFD) ή τερματικό ή αναγνώστης. Ο αναγνώστης είναι επίσης υπεύθυνος για τη δημιουργία ενός καναλιού επικοινωνίας μεταξύ του λειτουργικού συστήματος της κάρτας και του λογισμικού της εφαρμογής που μπορεί να τρέχει σε κάποιον υπολογιστή. Στην ουσία οι περισσότεροι αναγνώστες έξυπνων καρτών έχουν και τη δυνατότητα της εγγραφής ώστε να επιτρέπεται στην εφαρμογή (με την οποία γίνεται η επικοινωνία) τόσο η εγγραφή δεδομένων στην κάρτα όσο και η ανάγνωση.

3.2 Βήματα Διαδικασίας Κατασκευής της έξυπνης κάρτας

Τα βήματα που ακολουθούνται για την παραγωγή και έκδοση μιας έξυπνης κάρτας είναι τα ακόλουθα:

- Κατασκευή του ολοκληρωμένου κυκλώματος
- Ο δίσκος πυριτίου με τα ολοκληρωμένα κυκλώματα ελέγχεται και σημειώνονται αυτά που λειτουργούν σωστά. Στη συνέχεια ο δίσκος τεμαχίζεται σε ξεχωριστά ολοκληρωμένα το κάθε ένα από τα οποία συνδέεται με το μεταλλικό έλασμα, που περιέχει τις 8 επαφές, ώστε να δημιουργηθεί ένα άρθρωμα(module). Αυτές οι 8 μηχανικές επαφές χρησιμοποιούνται για να παρέχεται ενέργεια και το σήμα ρολογιού στην κάρτα καθώς και για τη μετάδοση δεδομένων.

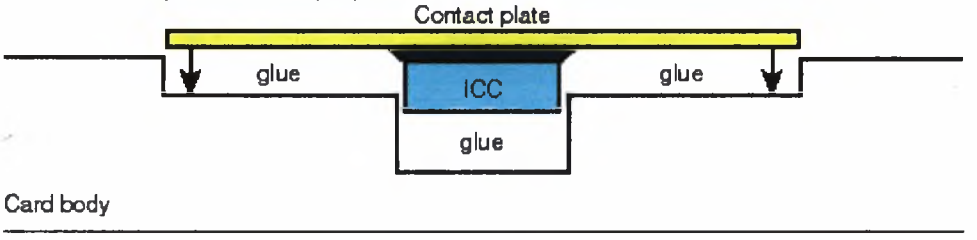


3.3 Βήμα 2ο

- Για την κατασκευή της πλαστικής κάρτας χρησιμοποιείται ως υλικό, η χημική ένωση πολυβινιλοχλωρίδιο (PVC). Τόσο τα χημικά χαρακτηριστικά όσο και οι

διαστάσεις της κάρτας και οι φυσικές αντοχές της καθορίζονται από διεθνή πρότυπα. Το υλικό της κάρτας που παράγεται αρχικά έχει τη μορφή ενός μεγάλου επίπεδου φύλλου και με το προκαθορισμένο πάχος. Στη φασή αυτή μπορεί να γίνει η εκτύπωση της κάρτας δηλαδή η τοποθέτηση κειμένων και γραφικών και στις δύο όψεις της. Στη συνέχεια αυτό το υλικό κόβεται, ώστε να παραχθούν πολλές ξεχωριστές κάρτες.

- Στο αμέσως επόμενο βήμα γίνεται η εισαγωγή του chip στο εσωτερικό της κάρτας. Δημιουργείται ένα βαθούλωμα ή τρύπα στην πλαστική κάρτα και σε αυτό εφαρμόζεται το άρθρωμα (module).



3.4 Εισαγωγή του chip στην κάρτα

- Πριν λάβει χώρα η αρχικοποίηση ή προσωποποίηση της κάρτας θα πρέπει να φορτωθούν στην μνήμη EEPROM της έξυπνης κάρτας όλα τα απαιτούμενα προγράμματα και τα αρχεία δεδομένων. Το αμέσως επόμενο βήμα είναι η διαδικασία προσωποποίησης.
- Στη φάση αυτή εισάγονται στην κάρτα πληροφορίες όπως ονόματα κατόχων ή αριθμοί λογαριασμών, όπως επίσης και ο προσωπικός κωδικός του κάθε κατόχου (PIN) τον οποίο μπορεί να χρησιμοποιεί στη συνέχεια για να αποδεικνύει την ταυτότητα του. Σε αυτό το βήμα μπορεί επίσης να γίνει η εκτύπωση στοιχείων όπως το όνομα του κατόχου ή η διεύθυνση του ή ακόμη και η φωτογραφία του πάνω στην κάρτα.

3.3 Λειτουργικό Σύστημα Έξυπνων Καρτών (Chip Operating System)

Κάθε έξυπνη κάρτα με ενσωματωμένο μικροεπεξεργαστή διαθέτει λειτουργικό σύστημα το οποίο ονομάζεται ***Λειτουργικό Σύστημα Κάρτας*** (***Card Operating System*** ή ***Chip Operating System***). Παρέχει τη δυνατότητα εκτέλεσης βασικών λειτουργιών όπως ασφαλή πρόσβαση και αποθήκευση δεδομένων στην κάρτα, πιστοποίηση ταυτότητας και εκτέλεση κρυπτογραφικών αλγορίθμων.

3.3.1 Κατηγορίες Λειτουργικών Συστημάτων

Το Chip Operating System της έξυπνης κάρτας είναι ουσιαστικά μια ακολουθία εντολών ενσωματωμένη μόνιμα στην μνήμη ROM της έξυπνης κάρτας. Οι εντολές του COS δεν εξαρτώνται από κάποια εφαρμογή, όπως συμβαίνει και στο DOS ή τα Windows, αλλά χρησιμοποιούνται από τις περισσότερες εφαρμογές. Υπάρχουν δύο βασικές κατηγορίες λειτουργικών συστημάτων COS :

I. ***COS γενικού σκοπού (General purpose COS):*** Σύμφωνα με την πρώτη αυτή προσέγγιση η κάρτα αντιμετωπίζεται ως μια ασφαλής συσκευή υπολογισμού και αποθήκευσης. Τα αρχεία και η άδεια πρόσβασης σε αυτά ορίζονται από τον εκδότη της κάρτας. Η μόνη πρόσβαση στην κάρτα γίνεται μέσω του λειτουργικού της συστήματος ενώ δεν γίνεται καμία τροποποίηση της δομής των αρχείων της κάρτας. Τα περιεχόμενα της κάρτας διαβάζονται ή ενημερώνονται σύμφωνα με τις άδειες που έχουν ορίσει οι εκδότες. Οι ενέργειες για τις οποίες είναι υπεύθυνο το λειτουργικό σύστημα είναι η πιστοποίηση της ταυτότητας του κατόχου και η κρυπτογράφηση που εκτελούνται μέσω των εντολών που στέλνονται στην κάρτα.

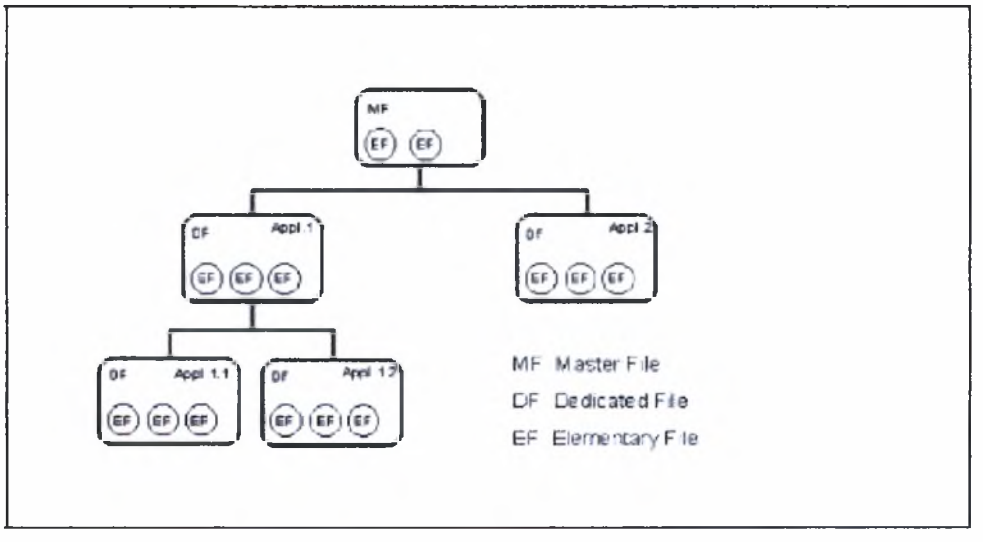
II. ***Dedicated COS:*** Οι εντολές του λειτουργικού συστήματος αυτής της κατηγορίας σχεδιάζονται για συγκεκριμένες εφαρμογές και μάλιστα μπορεί να περιλαμβάνουν και την ίδια την εφαρμογή. Σε αυτή την περίπτωση υπάρχει διαχειριστής μνήμης για τη φόρτωση της εφαρμογής και των αρχείων.

Οι βασικές λειτουργίες του COS οι οποίες είναι κοινές σε όλες τις έξυπνες κάρτες περιλαμβάνουν :

- Διαχείριση της επικοινωνίας και της ανταλλαγής δεδομένων μεταξύ έξυπνης κάρτας και εξωτερικού κόσμου
- Διαχείριση των αρχείων και δεδομένων που είναι αποθηκευμένα στη μνήμη
- Έλεγχος πρόσβασης σε δεδομένα και λειτουργίες
- Εκτέλεση κρυπτογραφικών αλγορίθμων
- Διαχείριση διακοπών και ανάκαμψη από τυχόν σφάλματα με δυνατότητα διατήρησης της συνέπειας των στοιχείων της κάρτας

3.3.2 Σύστημα Αρχείων

Τα περισσότερα λειτουργικά συστήματα έξυπνων καρτών χρησιμοποιούν ένα ιεραρχικό σύστημα αρχείων όπως αυτό καθορίζεται από το ISO 7816 smart card standard **[4]**.
Τα δεδομένα αποθηκεύονται στην μνήμη EEPROM και είναι οργανωμένα σε αρχεία με τρόπο πανομοιότυπο των προσωπικών υπολογιστών. Πιο συγκεκριμένα τα αρχεία είναι οργανωμένα σε μια ιεραρχική δομή δένδρου. Στην ρίζα του δένδρου βρίσκεται το λεγόμενο Master file (MF) το οποίο είναι υποχρεωτικό. Στο αμέσως επόμενο επίπεδο μπορεί να υπάρχουν είτε τα λεγόμενα dedicated Files (DFs) στα οποία αποθηκεύονται οι διάφορες εφαρμογές (στην ουσία αποτελούν τους καταλόγους όπου βρίσκονται οι εφαρμογές) , είτε τα elementary files (EFs) τα οποία χρησιμοποιούνται για την αποθήκευση των δεδομένων του κατόχου της κάρτας καθώς και πληροφοριών όπως είναι οι διάφοροι κωδικοί και τα κλειδιά.
Τα dedicated files τα οποία μπορούν να περιέχουν elementary files χρησιμοποιούνται για τον λογικό διαχωρισμό των δεδομένων ενώ παράλληλα δίνουν την δυνατότητα ορισμού διαφορετικών επιπέδων προστασίας σε σχέση με τα elementary files. Τα elementary files χωρίζονται σε δύο κατηγορίες, στα Internal EFs και στα External EFs. Τα πρώτα χρησιμοποιούνται από την κάρτα για λόγους διαχείρισης και ελέγχων πρόσβασης, ενώ η δεύτερη κατηγορία χρησιμοποιείται για την αποθήκευση δεδομένων προσπελάσιμων από τον εξωτερικό κόσμο.



3.5 Smart Card File System

3.3.3 Αναγνωριστικά αρχείων

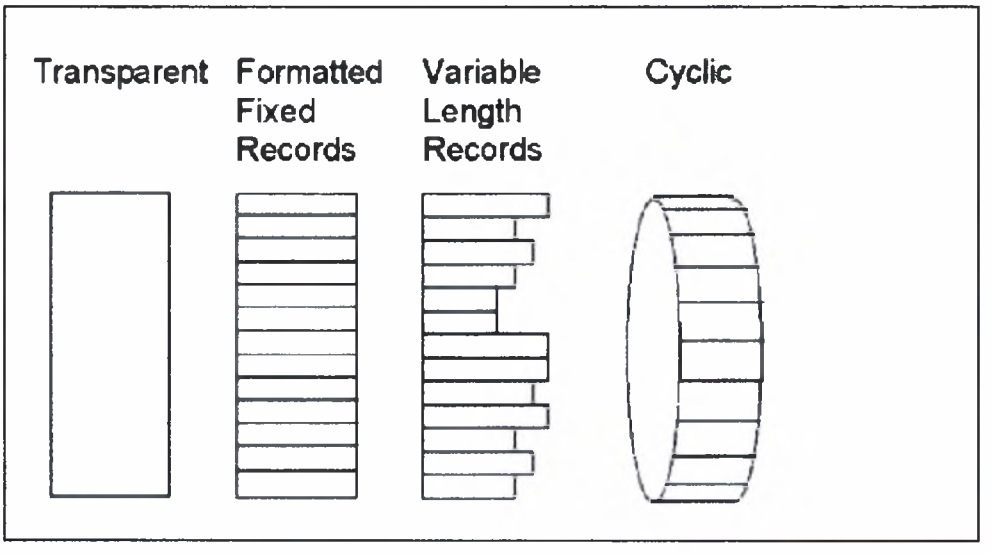
Σύμφωνα με τα ISO standards έχουν οριστεί διάφορα αναγνωριστικά αρχείων ώστε να μπορούν αυτά να προσπελαστούν. Κάθε αρχείο θα πρέπει να έχει τουλάχιστον ένα από τα ακόλουθα αναγνωριστικά:

- ***File Identifier***: Είναι ένας δεκαεξαδικός κωδικός μεγέθους των 2 bytes. Για παράδειγμα το αρχείο MF έχει το αναγνωριστικό (ID) 3F00 όπως έχει οριστεί από το ISO. Αντίστοιχα ένα EF θα μπορούσε να έχει το ID 2F01
- ***Path Name***: Είναι μια ακολουθία από αναγνωριστικά αρχείων (όπως για παράδειγμα 3F00A0010011) η οποία δείχνει το μονοπάτι από το αρχείο MF 3F00, μέσω του dedicated file A001, στο elementary αρχείο 0011.
- ***DF Name***: Μοναδικό κωδικοποιημένο όνομα μήκους από 1 έως 16 bytes για την προσπέλαση αρχείων DF. (Για παράδειγμα το Payment System Directory θα μπορούσε να έχει το όνομα "1.PAY.SYS.DDF01").

3.3.4 Elementary File Structure

Τα EF αρχεία όπως ορίζεται από το ISO διακρίνονται στους ακόλουθους τύπους αρχείων :

- ***Transparent***
- ***Record***
  - ***Linear Fixed***
  - ***Linear Variable***
  - ***Cyclic Fixed***



3.6 Smart Cards Elementary File Types

Transparent

Η πληροφορία που αποθηκεύεται στα transparent αρχεία είναι binary data ενώ δεν υπάρχει κάποια εσωτερική δομή. Αναφορικά με το μέγεθος των δεδομένων υπάρχει ένα θεωρητικό άνω όριο το οποίο είναι 64 Kbytes, μέγεθος που μπορεί να ισούται ή να ξεπερνά το μέγεθος την μνήμης EEPROM (κυμαίνεται από1 ή 64Kbytes) που μπορεί να παρέχεται από την σύγχρονη τεχνολογία. Πρακτικά όμως το μέγεθος των

δεδομένων είναι της τάξης των δεκάδων ή εκατοντάδων bytes έτσι ώστε ο χρόνος προσπέλασης των δεδομένων (ανάγνωση / εγγραφή) να είναι αποδεκτός.
Οι εντολές που χρησιμοποιούνται για τις διάφορες λειτουργίες του συγκεκριμένου τύπου αρχείων είναι : READ_BINARY, WRITE_BINARY και UPDATE_BINARY. Η προσπέλαση των δεδομένων γίνεται με τη χρήση ενός δείκτη offset που σχετίζεται με την αρχή του αρχείου.
Τυπικά χρησιμοποιούνται για την αποθήκευση δεδομένων όπως είναι το όνομα του κατόχου της κάρτας, ο αριθμός λογαριασμού του, προσωπικές πληροφορίες, κωδικούς και κλειδιά. Τα συγκεκριμένα αρχεία μπορούν επίσης να αποθηκεύουν επεκτάσεις κώδικα λειτουργικών συστημάτων και εκτελέσιμο κώδικα ο οποίος καλείται Application Specific Commands (ASC).

Linear Fixed

Αυτός ο τύπος αρχείων αποτελείται από εγγραφές το μέγεθος των οποίων προσαρμόζεται κατάλληλα, με την κάθε εγγραφή να αποτελείται από μια σειρά από bytes. Το μέγιστο μέγεθος μιας εγγραφής είναι 254 bytes όπως και το μέγιστο πλήθος εγγραφών, που είναι ίσο με 254. Κάθε εγγραφή έχει ένα αναγνωριστικό αριθμό και η αρίθμηση ξεκινά από το ένα. Η προσπέλαση των εγγραφών γίνεται είτε μέσω του αναγνωριστικού αριθμού είτε μέσω παραμέτρων όπως είναι FIRST, LAST, NEXT. Οι εντολές που χρησιμοποιούνται είναι της μορφής READ_RECORD, WRITE_RECORD, UPDATE_RECORD.
Στα Linear Fixed αρχεία αποθηκεύονται δεδομένα όπως λίστες με κλειδιά εφαρμογών.

Linear Variable

Στα αρχεία τύπου Linear Fixed μπορεί αρκετός χώρος να μένει ανεκμετάλλευτος αν οι εγγραφές είναι άνισου μεγέθους και ο χώρος που απαιτείται είναι μικρότερος του μέγιστου μεγέθους της εγγραφής. Για την αποφυγή της σπατάλης χώρου χρησιμοποιούνται τα Linear Variable αρχεία στα οποία η κάθε εγγραφή έχει ένα επιπρόσθετο byte για τον καθορισμό του μήκους της εγγραφής. Το μέγιστο μέγεθος της κάθε εγγραφής είναι 254 bytes και ο μέγιστος αριθμός εγγραφών είναι 254 εγγραφές, ακριβώς όπως συμβαίνει και στα αρχεία τύπου Linear Fixed.
Τα δεδομένα που αποθηκεύονται στα συγκεκριμένα αρχεία είναι λίστες από αριθμούς τηλεφώνων, και λίστες από εξουσιοδοτημένες εφαρμογές.

Cyclic Fixed

Η δομή τους είναι παρόμοια με αυτή των linear fixed αρχείων. Ένας δείκτης ο οποίος αυξάνεται αυτόματα δείχνει πάντα στην εγγραφή που προσπελάστηκε τελευταία. Αν ο δείκτης φτάσει στο όριο δηλαδή στο μέγιστο αριθμό των εγγραφών που μπορούν να υπάρχουν σε ένα αρχείο μηδενίζεται και ξεκινά από την αρχή. Οι εντολές που χρησιμοποιούνται για την προσπέλαση των εγγραφών είναι οι ίδιες που χρησιμοποιούνται και στα αρχεία που αναφέρθηκαν παραπάνω.
Τυπικά χρησιμοποιούνται για αποθήκευση πληροφοριών όπως για παράδειγμα μια λίστας με τις τελευταίες υπηρεσίες χρησιμοποιήθηκαν από τον κάτοχο της κάρτας.

3.3.5 Εντολές Προσπέλασης Αρχείων

Σύμφωνα με το πρότυπο ISO 7816-4 έχει οριστεί ένας αριθμός εντολών σύμφωνα με τις οποίες μπορεί να γίνει η επιλογή ενός αρχείου, η ανάγνωση και η εγγραφή σε αυτό. Αυτές οι εντολές περιγράφονται συνοπτικά παρακάτω **[8].**

Select File

Αυτή η εντολή δημιουργεί έναν λογικό δείκτη σε ένα συγκεκριμένο αρχείο που περιέχεται στο σύστημα αρχείων της έξυπνης κάρτας και απαιτείται από κάθε λειτουργία διαχείρισης αρχείων. Η πρόσβαση στο σύστημα αρχείων της κάρτας δεν είναι πολυνηματική, ωστόσο είναι πιθανό να οριστούν αρκετοί δείκτες αρχείων σε οποιαδήποτε χρονική στιγμή. Αυτό μπορεί να πραγματοποιηθεί με την εντολή Manage Channel η οποία δημιουργεί πολλαπλά λογικά κανάλια μεταξύ αναγνώστη και έξυπνης κάρτας. Αυτό επιτρέπει στον αναγνώστη να διαχειρίζεται διαφορετικά αρχεία την ίδια χρονική στιγμή.

Read Binary

Αυτή η εντολή χρησιμοποιείται από την εφαρμογή που τρέχει στον αναγνώστη, για την ανάκτηση ενός αρχείου EF το οποίο θα πρέπει να είναι τύπου transparent και όχι record διαφορετικά επιστρέφεται λάθος Η εντολή παίρνει δύο παραμέτρους, έναν δείκτη ο οποίος δείχνει στο πρώτο προς ανάγνωση byte και τον αριθμό των προς ανάγνωση bytes. Το αποτέλεσμα που επιστρέφεται στον αναγνώστη είναι τα bytes που διαβάστηκαν.

Write Binary

Χρησιμοποιείται από την πλευρά του αναγνώστη για την εισαγωγή δεδομένων σε κάποιο τμήμα ενός EF αρχείου της κάρτας. Έτσι μπορεί να γραφτεί στην κάρτα μια σειρά από bytes (τα αντίστοιχα bits παίρνουν την τιμή 1) όπως επίσης και να διαγραφεί μια ακολουθία bytes (τα αντίστοιχα bits παίρνουν την τιμή 0).

Update Binary

Καλείται από την πλευρά της εφαρμογής που τρέχει στον αναγνώστη και μοιάζει με την εντολή write binary, μόνο που σε αυτή την περίπτωση τα bytes που πρόκειται να αντικατασταθούν αρχικά διαγράφονται από το αρχείο και στη συνέχεια ακριβώς στην ίδια θέση του αρχείου εγγράφονται τα νέα bytes. Η εντολή παίρνει ως παραμέτρους έναν δείκτη offset που δείχνει στην αρχή του αρχείου και έναν μετρητή byte για τον συνολικό αριθμό των bytes που πρόκειται να γραφτούν.

Erase Binary

Χρησιμοποιείται για την διαγραφή bytes που περιέχονται σε ένα αρχείο EF. Η εντολή παίρνει ως παραμέτρους έναν δείκτη που καθορίζει ποιο τμήμα από bytes θα διαγραφεί και τον αριθμό των bytes που περιέχει το συγκεκριμένο τμήμα.

Read Record

Αυτή η εντολή χρησιμοποιείται για την ανάγνωση μιας ή περισσοτέρων εγγραφών (records) που περιέχονται σε ένα EF αρχείο την κάρτας. Εκτελείται από την πλευρά της εφαρμογής που τρέχει στον αναγνώστη και επιστρέφει σε αυτή το περιεχόμενο της εγγραφής ή των εγγραφών που έχουν διαβαστεί. Ανάλογα με τις παραμέτρους που θα περαστούν στην εντολή μπορεί είτε να διαβαστεί και να επιστραφεί μια εγγραφή που θα επιλεγεί, είτε όλες οι εγγραφές από την αρχή του αρχείου έως μια επιθυμητή εγγραφή, είτε όλες οι εγγραφές αρχίζοντας από μια επιθυμητή εγγραφή έως το τέλος του αρχείου.

Write Record

Για την προσθήκη μιας νέας εγγραφής σε ένα αρχείο της κάρτας καλείται η εντολή write record. Επιπλέον χρησιμοποιείται για την προσθήκη ή διαγραφή ενός συγκεκριμένου αριθμού bits σε μια εγγραφή του αρχείου, που θα καθοριστεί από την εφαρμογή. Οι μέθοδοι εντοπισμού και αναγνώρισης των εγγραφών, είναι : τα αναγνωριστικά first/next/previous/last record όπως επίσης και ο αναγνωριστικός αριθμός που ορίζεται για κάθε εγγραφή.

Append Record

Η συγκεκριμένη εντολή χρησιμοποιείται για την προσθήκη μια εγγραφής στο τέλος ενός αρχείου τύπου linear record είτε για την προσθήκη της πρώτης εγγραφής σε ένα αρχείο τύπου cyclic record.

Update Record

Όμοια με την εντολή update binary καλείται για την αντικατάσταση μιας εγγραφής. Το αποτέλεσμα της εντολής είναι η διαγραφή μιας συγκεκριμένης εγγραφής και η προσθήκη της νέας.

Get Data

Χρησιμοποιείται για την ανάγνωση και ανάκτηση δεδομένων που είναι αποθηκευμένα στα αρχεία της κάρτας. Ο ακριβής ορισμός και οι παράμετροι που παίρνει η εντολή ποικίλουν ανάλογα με την κάρτα και το είδος των δεδομένων που είναι αποθηκευμένα σε αυτή.

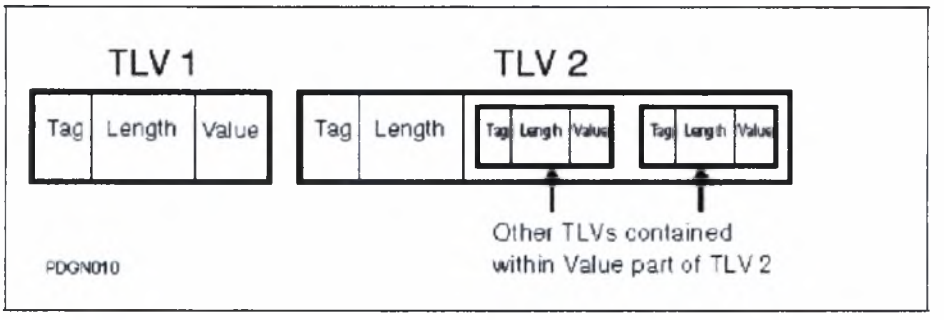
Put Data

Χρησιμοποιείται για την προσθήκη πληροφοριών στα δεδομένα που είναι αποθηκευμένα στα αρχεία της κάρτας.

Δομές Δεδομένων

Η δομή των δεδομένων που θα αποθηκευτούν στα αρχεία της κάρτας καθορίζεται από τις διάφορες εφαρμογές. Ωστόσο μια στατική δομή δεδομένων μπορεί να προκαλέσει προβλήματα, όπως στην περίπτωση που το μήκος του πεδίου των

δεδομένων μπορεί να είναι μικρότερο ή μεγαλύτερο από το μέγεθος που είχε καθοριστεί αρχικά. Αυτού του είδους προβλήματα μπορεί να προκαλέσουν δυσλειτουργίες και στις διάφορες εφαρμογές της κάρτας. Τη λύση σε αυτό το πρόβλημα έχει δώσει η δομή δεδομένων TLV (Tag Length Value) σύμφωνα με την οποία χρησιμοποιείται μια επικεφαλίδα (header or tag) η οποία περιγράφει το μήκος των δεδομένων και το περιεχόμενο τους.



3.7.TLV Data Structures

TLV Δομή Δεδομένων

Η δομή αυτή μπορεί να χρησιμοποιηθεί σε όλους του τύπους αρχείων που μελετήθηκαν. Τα δεδομένα προσπελαύνονται χρησιμοποιώντας την επικεφαλίδα και το ρυθμιζόμενο μήκος του πεδίου. Η δομή TLV χρησιμοποιείται και για τον ορισμό ακόμη πιο σύνθετων δομών όπως δομές TLV μέσα σε μια TLV.

3.3.6 Σύστημα Ασφάλειας Αρχείων

Η ασφάλεια αποτελεί το σημαντικότερο πλεονέκτημα της τεχνολογίας των έξυπνων καρτών. Για την επίτευξη ενός ασφαλούς συστήματος αρχείων θα πρέπει να καθορίζεται με ποιο τρόπο και ποιος μπορεί να έχει πρόσβαση στα δεδομένα που αποθηκεύονται στην έξυπνη κάρτα.
Οι διαφορετικοί τύποι αρχείων που περιγράψαμε παραπάνω μπορούν να προστατευθούν με την χρήση ιδιοτήτων πρόσβασης και συνθηκών πρόσβασης. Για παράδειγμα μια τέτοια ιδιότητα θα μπορούσε να είναι η *ανάγνωση* και μια συνθήκη η *πιστοποίηση του κατόχου της κάρτας*. Αυτό σημαίνει ότι για να επιτραπεί η ανάγνωση του αντίστοιχου αρχείου της κάρτας θα πρέπει πρώτα να γίνει η εισαγωγή του σωστού κωδικού από τον κάτοχο.
Οι βασικότεροι μηχανισμοί ασφάλειας όπως αυτοί ορίζονται από το ISO 7816-4 είναι :

- ***Αυθεντικοποίηση με password*** : Ο εξωτερικός κόσμος θα πρέπει πρώτα να αποδεικνύει ότι γνωρίζει τον απαιτούμενο κωδικό προτού του επιτραπεί η πρόσβαση.
- ***Αυθεντικοποίηση με κλειδί*** : Ο εξωτερικός κόσμος θα πρέπει να αποδεικνύει ότι γνωρίζει το κλειδί που είναι ήδη αποθηκευμένο στην κάρτα προτού του επιτραπεί η πρόσβαση.

- ***Αυθεντικότητα δεδομένων*** : Η κάρτα επισυνάπτει κάποιο κώδικα στα δεδομένα έτσι ώστε να πιστοποιείται στον εξωτερικό κόσμο η γνησιότητα της κάρτας.
- ***Κρυπτογράφηση των δεδομένων*** : Η κάρτα με τους κατάλληλους κρυπτογραφικούς αλγορίθμους, κρυπτογραφεί τα δεδομένα έτσι ώστε να εξασφαλίζεται η εμπιστευτικότητα των δεδομένων.

Αυτοί οι μηχανισμοί ασφάλειας υλοποιούνται από το λειτουργικό σύστημα της έξυπνης κάρτας με τη χρήση των ιδιοτήτων πρόσβασης των αρχείων και με τις συνθήκες πρόσβασης. Για κάθε τύπο αρχείου (MF,EF,DF) υπάρχουν επίπεδα ασφάλειας. Στα αρχεία τύπου MF και DF το επίπεδο ασφάλειας είναι συνήθως υψηλό που σημαίνει ότι η πρόσβαση παρέχεται μόνο στους εκδότες καρτών (card issuers) και στους παροχείς εφαρμογών (application providers). Για παράδειγμα ένας παροχέας εφαρμογών μπορεί να έχει το δικαίωμα να απενεργοποιήσει μια εφαρμογή μόνο εφόσον γνωρίζει τα ειδικά κλειδιά που βρίσκονται αποθηκευμένα στα αρχεία της κάρτας.
Η υλοποίηση ενός ασφαλούς συστήματος αρχείων μπορεί είτε να περιλαμβάνει απλά τους ISO βασικούς μηχανισμούς ασφάλειας είτε και επιπρόσθετους μηχανισμούς αυθεντικοποίησης με password και κρυπτογράφηση δεδομένων για την βελτίωση του επιπέδου ασφάλειας της κάρτας.
Οι ιδιότητες πρόσβασης των αρχείων στο IBM MFC Smart Card λειτουργικό σύστημα είναι οι ακόλουθες :

- (ALW) always
- (NEV) never
- (CHV1) κωδικός κατόχου (cardholder's password PIN)
- (CHV2) κωδικός διαχειριστή (administrator' s password PIN)
- (PRO) protected
- (AUT) authenticated
- (ENC) enciphered
- Συνδυασμοί όπως
    - (CHV1or CHV2 / PRO)
    - (CHV1or CHV2 / AUT)
    - (CHV1or CHV2 / ENC)

Οι παραπάνω ιδιότητες εξηγούνται στα ακόλουθα τμήματα :

Always (ALW)

Όπως δηλώνει και το όνομα αυτής της ιδιότητας, η πρόσβαση είναι ελεύθερη, δηλαδή η κάρτα επιτρέπει την ανάγνωση και ενημέρωση των δεδομένων της. Στοιχεία όπως είναι το όνομα του κατόχου ή το υπόλοιπο χρηματικού ποσού μπορούν να διαβαστούν ενώ η ενημέρωση αυτών των στοιχείων δεν είναι πάντα επιτρεπτή.

Never (NEV)

Αποτελεί το ύψιστο επίπεδο προστασίας αφού η πρόσβαση απαγορεύεται ακόμα και στον εκδότη της κάρτας. Για παράδειγμα η ανάγνωση αρχείων στα οποία βρίσκονται αποθηκευμένοι κωδικοί δεν επιτρέπεται ποτέ και στα αρχεία αυτά προσδίδεται η ιδιότητα "Never". Μόνο οι μηχανισμοί που εκτελούνται στο

εσωτερικό της κάρτας και ο αντίστοιχος κώδικας που τρέχει μπορεί να πιστοποιήσει τον κάτοχο της κάρτας. Επίσης στα αρχεία Master File και Dedicated Files εφαρμόζεται το ύψιστο επίπεδο προστασίας.

Πιστοποίηση κατόχου της κάρτας (CHV1/CHV2)

Το πλεονέκτημα των έξυπνων καρτών έναντι των καρτών με μαγνητική ταινία είναι ότι αποθηκεύουν τον προσωπικό κωδικό του κατόχου της κάρτας στο εσωτερικό τους με τέτοιο τρόπο ώστε αυτός να μην μπορεί να διαβαστεί από τον εξωτερικό κόσμο. Η σύγκριση του κωδικού τον οποίο γνωρίζει μόνο ο κάτοχος γίνεται στο εσωτερικό του chip.
Συνήθως μια κάρτα έχει δύο κωδικούς τον CHV1 και CHV2, από τους οποίους ο πρώτος αντιστοιχεί στον κάτοχο της κάρτας και ο δεύτερος στον διαχειριστή της ο οποίος μπορεί να διαγράψει έναν ξεχασμένο κωδικό ή να μπλοκάρει έναν CHV1 κωδικό. Τα δεδομένα τα οποία προστατεύονται με την ιδιότητα CHV επιτρέπουν την ανάγνωση ή εγγραφή στον εξωτερικό κόσμο μόνο με την εισαγωγή του σωστού κωδικού.

Protection (PRO)

Για τον έλεγχο της μη αλλοίωσης των μηνυμάτων που ανταλλάσσονται μεταξύ της έξυπνης κάρτας και της συσκευής ανάγνωσης, προσάπτεται στο μήνυμα ένας κωδικός αυθεντικότητας μηνύματος (Message Authentication Code – MAC) εφόσον βέβαια έχει οριστεί ως συνθήκη πρόσβασης του αρχείου η συνθήκη PRO. Ο κωδικός αυθεντικότητας μηνύματος δημιουργείται με τη χρήση μιας συνάρτησης η οποία παίρνει ως εισόδους ένα κλειδί, έναν τυχαίο αριθμό και το μήνυμα που ανταλλάσσεται.

Authentication (AUT)

Υπάρχουν δύο μέθοδοι αυθεντικοποίησης η εξωτερική και η εσωτερική.

- ***Εξωτερική αυθεντικοποίηση***: Η συσκευή ανάγνωσης έξυπνων καρτών πρέπει να αποδείξει στην έξυπνη κάρτα ότι είναι εξουσιοδοτημένη να προσπελάσει / ενημερώσει τα δεδομένα που σημαίνει ότι τόσο η συσκευή ανάγνωσης όσο και η έξυπνη κάρτα έχουν τα ίδια κλειδιά. Ο αναγνώστης αυθεντικοποιεί τον εαυτό του αιτώντας από την έξυπνη κάρτα έναν τυχαίο αριθμό. Στην συνέχεια δείχνει στην κάρτα ότι τον έχει κρυπτογραφήσει σωστά.
- ***Εσωτερική αυθεντικοποίηση***: Η έξυπνη κάρτα πρέπει να αποδείξει στον αναγνώστη ότι είναι εξουσιοδοτημένη να στείλει δεδομένα στον αναγνώστη. Η ίδια διαδικασία αυθεντικοποίησης εκτελείται και σε αυτή την περίπτωση. Η κάρτα για να αυθεντικοποιήσει τον εαυτό της στη συσκευή ζητά από αυτή έναν τυχαίο αριθμό και στέλνει σε αυτή το αποτέλεσμα της κρυπτογράφησης. Εάν ο αριθμός έχει κρυπτογραφηθεί σωστά σημαίνει ότι η κάρτα είναι γνήσια.

Encryption (ENC)

Αυτή η ιδιότητα είναι πανομοιότυπη της ιδιότητας PRO με τη μόνη διαφορά ότι το μήνυμα αυθεντικοποίησης είναι κρυπτογραφημένο.

Για τις διαδικασίες αυθεντικοποίησης και για την εξασφάλιση της εξουσιοδοτημένης και μόνο προσπέλασης των συστατικών του συστήματος αρχείων χρησιμοποιούνται οι ακόλουθες εντολές :

Verify

Η εντολή αυτή στέλνεται από την συσκευή ανάγνωσης στην κάρτα με στόχο να την πείσει ότι η πλευρά του αναγνώστη γνωρίζει τον κωδικό που βρίσκεται αποθηκευμένος σε κάποιο συγκεκριμένο αρχείο της κάρτας και έτσι να του επιτραπεί η πρόσβαση σε ευαίσθητες πληροφορίες. Εάν ο κωδικός που δοθεί είναι λανθασμένος η εντολή αποτυγχάνει και στον αναγνώστη επιστρέφεται μήνυμα λάθους.

Internal Authenticate

Η συγκεκριμένη εντολή επιτρέπει στην κάρτα να αυθεντικοποιήσει τον εαυτό της αποδεικνύοντας ότι κατέχει το μυστικό κλειδί που μοιράζεται με τον αναγνώστη. Ο αναγνώστης δημιουργεί έναν τυχαίο αριθμό τον οποίο κρυπτογραφεί με έναν αλγόριθμο που γνωρίζουν και τα δύο μέρη. Η κάρτα στη συνέχεια αποκρυπτογραφεί τον αριθμό με τη χρήση του μυστικού κλειδιού που κατέχει και επιστρέφει το αποτέλεσμα πίσω στον αναγνώστη. Τα αποτελέσματα συγκρίνονται και αν είναι τα ίδια η διαδικασία αυθεντικοποίησης ήταν επιτυχής.

External Authenticate

Χρησιμοποιείται σε συνδυασμό με την εντολή Get Challenge για να δοθεί η δυνατότητα στον αναγνώστη να αυθεντικοποιήσει τον εαυτό του στην κάρτα. Ο αναγνώστης λαμβάνει έναν τυχαίο αριθμό από την κάρτα και τον κρυπτογραφεί με το κοινό μυστικό κλειδί. Το αποτέλεσμα στέλνεται στην κάρτα με την εντολή External Authenticate. Ακολούθως η κάρτα αποκρυπτογραφεί τα δεδομένα που της στάλθηκαν και συγκρίνει το αποτέλεσμα με τον τυχαίο αριθμό (που δημιούργησε προηγουμένως με την εντολή Get Challenge) και αν αυτά ταυτίζονται τότε η διαδικασία της εξωτερικής αυθεντικοποίησης έχει ολοκληρωθεί επιτυχώς.

Get Challenge

Στέλνεται από τον αναγνώστη στην κάρτα ώστε να αποκτήσει ο αναγνώστης έναν τυχαίο αριθμό τον οποίο θα χρησιμοποιήσει στην εντολή External authenticate.

Manage Challenge

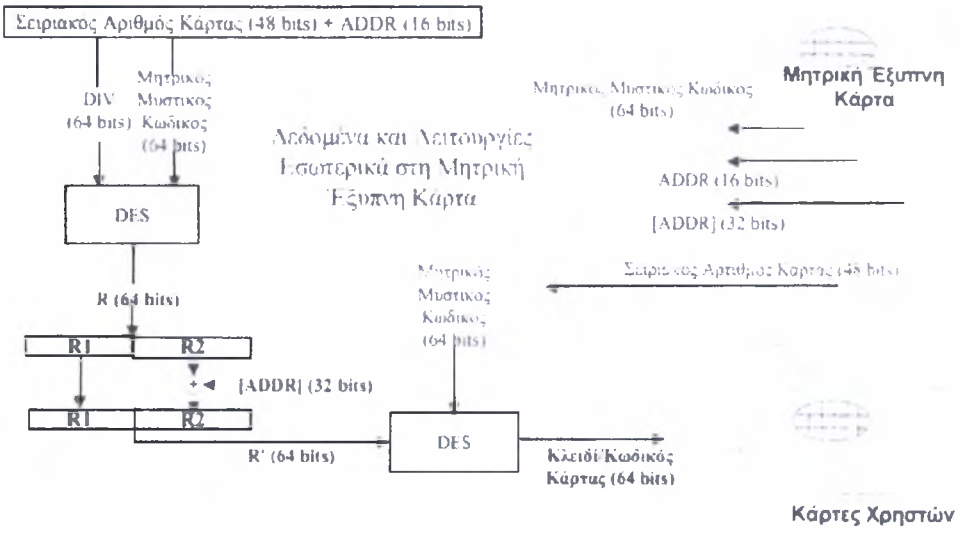
Καλείται από την πλευρά του αναγνώστη για το άνοιγμα και κλείσιμο των λογικών καναλιών επικοινωνίας μεταξύ αυτού και της κάρτας. Αρχικά η κάρτα ανοίγει το κανάλι επικοινωνίας ορίζοντας το πρωτόκολλο επιπέδου εφαρμογής που θα χρησιμοποιηθεί για την επικοινωνία με τον αναγνώστη. Αυτό το αρχικό κανάλι επικοινωνίας χρησιμοποιείται για το άνοιγμα ή κλείσιμο επιπλέον καναλιών επικοινωνίας μέσω της εντολής Manage Challenge.

Envelope

Χρησιμοποιείται για την ασφαλή ανταλλαγή μηνυμάτων όταν ως πρωτόκολλο επικοινωνίας ορίζεται το T=0. Έτσι τα APDUs (έτσι καλούνται τα μηνύματα που ανταλλάσσονται μεταξύ κάρτας και αναγνώστη στο επίπεδο εφαρμογής) που ανταλλάσσονται κρυπτογραφούνται πρώτα, συγχωνεύονται στο κατάλληλο τμήμα της εντολής Envelope και έπειτα εκτελείται η εντολή από τον APDU processor της κάρτας.

3.4 Αυθεντικοποίηση με έξυπνες κάρτες

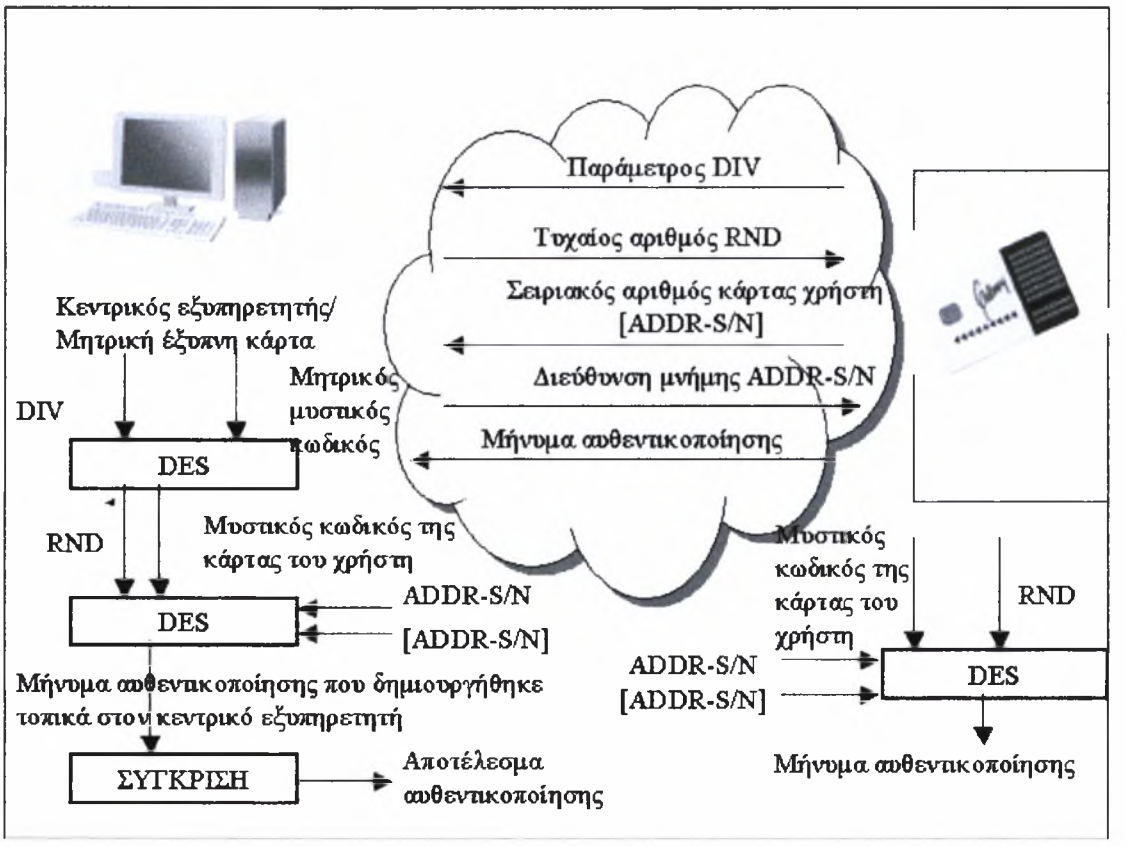
Η διαδικασία της αυθεντικοποίησης έχει σαν στόχο τη διασφάλιση της γνησιότητας μιας έξυπνης κάρτας, δηλαδή ότι πράγματι έχει εκδοθεί από εξουσιοδοτημένο φορέα στα πλαίσια κάποιας συγκεκριμένης εφαρμογής και κατά συνέπεια να επιτρέψει την πρόσβαση του κατόχου της σε συγκεκριμένες υπηρεσίες και δεδομένα.
Η υλοποίηση του μηχανισμού της αυθεντικοποίησης γίνεται είτε με τη χρήση ψηφιακών υπογραφών και ασύμμετρων κρυπτογραφικών αλγορίθμων, είτε αξιοποιώντας συμμετρικούς αλγόριθμους κρυπτογράφησης.
Οι έξυπνες κάρτες πριν διανεμηθούν στους κατόχους τους, πρέπει να προσωποποιηθούν (η προσωποποίηση αποτελεί ένα από τα βήματα παραγωγής και έκδοσης μιας έξυπνης κάρτας). Κατά τη διάρκεια της συγκεκριμένης φάσης καθορίζονται και εγγράφονται στην κάρτα οι απαραίτητοι μυστικοί κωδικοί και κλειδιά. Με στόχο την προστασία της εμπιστευτικότητας των κλειδιών και κωδικών κάθε κάρτας, η δημιουργία τους βασίζεται στη διαφοροποίηση ενός μητρικού μυστικού κωδικού (master secret code) που βρίσκεται αποθηκευμένο στη γνωστή ως μητρική έξυπνη κάρτα (security module).
Όπως απεικονίζεται στο σχήμα που ακολουθεί (σχήμα 8), η μητρική έξυπνη κάρτα εκτελεί εσωτερικά τον κρυπτογραφικό αλγόριθμο DES (αν υποθέσουμε ότι αυτός υποστηρίζεται από το συγκεκριμένο τύπο έξυπνης κάρτας) ο οποίος παίρνει ως παραμέτρους εισόδου : το μητρικό μυστικό κωδικό που όπως έχει προαναφερθεί βρίσκεται αποθηκευμένος μέσα στην κάρτα, μια προκαθορισμένη διεύθυνση μνήμης της μητρικής έξυπνης κάρτας (ADDR), τα περιεχόμενα της συγκεκριμένης θέσης μνήμης [ADDR]και το σειριακό αριθμό της κάρτας που πρόκειται να προσωποποιηθεί **[14].**



3.8 Διαδικασία προσωποποίησης Έξυπνων Καρτών

Το αποτέλεσμα του κρυπτογραφικού αλγορίθμου αποτελεί το μυστικό κλειδί / κωδικό που τελικά θα αποθηκευτεί στην έξυπνη κάρτα του χρήστη. Το γεγονός ότι όλες οι έξυπνες κάρτες έχουν διαφορετικό σειριακό αριθμό, διασφαλίζει τη διαφοροποίηση των παραγόμενων κλειδιών / κωδικών για τις κάρτες των χρηστών. Σημαντικό γεγονός επίσης αποτελεί ότι το κλειδί / κωδικός που τελικά καταχωρήθηκε στην κάρτα του χρήστη δεν είναι γνωστό σε κανένα και ο μόνος τρόπος για να αναπαραχθεί προϋποθέτει την χρήση της μητρικής έξυπνης κάρτας και την επανάληψη της ίδιας ακριβώς διαδικασίας που ακολουθήθηκε για την αρχική παραγωγή του.

Έχουμε ήδη αναφέρει ότι για να αποκτήσει κάποιος χρήστης- κάτοχος κάρτας πρόσβαση σε υπηρεσίες ή δεδομένα θα πρέπει να προηγηθεί η αυθεντικοποίηση του. Η διαδικασία αυτή υλοποιείται στο ακόλουθο διάγραμμα (σχήμα 9). Όταν κάποιος χρήστης εισάγει την κάρτα σε ένα τερματικό, ο κεντρικός εξυπηρετητής στον οποίο είναι μόνιμα εγκαταστημένη η μητρική έξυπνη κάρτα, ζητά από την έξυπνη κάρτα του χρήστη τη δημιουργία και την αποστολή ενός μηνύματος αυθεντικοποίησης (authentication message). Για τη δημιουργία αυτού του μηνύματος χρησιμοποιούνται μεταξύ των άλλων ένας τυχαίος αριθμός και ο σειριακός αριθμός της έξυπνης κάρτας του χρήστη. Το ίδιο μήνυμα αυθεντικοποίησης υπολογίζεται και από την μητρική έξυπνη κάρτα, ανεξάρτητα από την κάρτα του χρήστη, και τα δύο μηνύματα συγκρίνονται. Αν αυτά ταυτίζονται τότε διασφαλίζεται το γεγονός ότι η κάρτα του χρήστη είναι γνήσια.



3.9 Διαδικασία Αυθεντικοποίησης

Τα βήματα που ακολουθούνται κατά τη διαδικασία της αυθεντικοποίησης αναλύονται παρακάτω:

Δημιουργία μηνύματος αυθεντικοποίησης από την κάρτα του χρήστη

Ο κεντρικός εξυπηρετητής δημιουργεί και αποστέλλει στην έξυπνη κάρτα του χρήστη ένα τυχαίο αριθμό (RND), μαζί με τη διεύθυνση της θέσης μνήμης που είναι αποθηκευμένος ο σειριακός αριθμός της έξυπνης κάρτας (ADDR-S/N).
Η έξυπνη κάρτα του χρήστη εκτελεί τον κρυπτογραφικό αλγόριθμο DES με παραμέτρους εισόδου τον τυχαίο αριθμό (RND), το μυστικό κωδικό της κάρτας, τη διεύθυνση της θέσης μνήμης που είναι αποθηκευμένος ο σειριακός αριθμός της (ADDR-S/N) και τον ίδιο τον σειριακό αριθμό της ([ADDR-S/N]). Στην πράξη ο αλγόριθμος DES εκτελείται δύο φορές όπως συμβαίνει στη διαδικασία προσωποποίησης της κάρτας. Η διαφορά είναι ότι δεν χρησιμοποιείται ο σειριακός αριθμός της κάρτας αλλά ένας τυχαίος αριθμός και η διεύθυνση (ADDR-S/N) αναφέρεται σε θέση μνήμης της έξυπνης κάρτας του χρήστη και όχι της μητρικής κάρτας.
Το αποτέλεσμα της παραπάνω διαδικασίας είναι η δημιουργία του "μηνύματος αυθεντικοποίησης" το οποίο και αποστέλλεται από την έξυπνη κάρτα του χρήστη στον κεντρικό εξυπηρετητή.

Αναπαραγωγή Μυστικού κωδικού της Κάρτας του χρήστη από τη Μητρική Έξυπνη κάρτα

Η έξυπνη κάρτα δημιουργεί και αποστέλλει στον κεντρικό εξυπηρετητή την παράμετρο DIV (περιλαμβάνει τον σειριακό αριθμό της κάρτας +(ADDR-S/N)).
Η μητρική έξυπνη κάρτα παράγει το μυστικό κωδικό της κάρτας του χρήστη, ακολουθώντας την ίδια ακριβώς διαδικασία που περιγράψαμε κατά τη διάρκεια της προσωποποίησης της έξυπνης κάρτας.

Δημιουργία Μηνύματος Αυθεντικοποίησης από την Μητρική Έξυπνη Κάρτα

Η έξυπνη κάρτα αποστέλλει στον κεντρικό εξυπηρετητή τα περιεχόμενα ([ADDR-S/N]) της θέσης μνήμης (ADDR-S/N), που είναι ο σειριακός αριθμός της κάρτας.
Η μητρική έξυπνη κάρτα αξιοποιεί τον μυστικό κωδικό της κάρτας του χρήστη, τον οποίο αναπαρήγαγε, για να υπολογίσει το "μήνυμα αυθεντικοποίησης" με τον ίδιο τρόπο που περιγράφηκε στο πρώτο βήμα (Δημιουργία μηνύματος αυθεντικοποίησης από την κάρτα του χρήστη)

Σύγκριση των Μηνυμάτων Αυθεντικοποίησης

Στην τελική αυτή φάση της αυθεντικοποίησης η μητρική έξυπνη κάρτα συγκρίνει το "μήνυμα αυθεντικοποίησης" που υπολογίστηκε τοπικά με αυτό που έστειλε η έξυπνη κάρτα του χρήστη. Αν τα δύο αυτά μηνύματα ταυτίζονται τότε αποδεικνύεται η γνησιότητα της κάρτας του χρήστη.

Το σημαντικό πλεονέκτημα της παραπάνω διαδικασίας είναι ότι κατά τη διάρκεια εκτέλεσης των βημάτων κανένα μυστικό κλειδί / κωδικός δεν μεταφέρεται μέσω του δικτύου από ή προς την έξυπνη κάρτα του χρήστη. Μόνο το "μήνυμα αυθεντικοποίησης" που υπολογίζεται από την κάρτα του χρήστη αποστέλλεται μέσω του δικτύου στον κεντρικό εξυπηρετητή, που όμως λόγω της χρήσης του τυχαίου αριθμού είναι κάθε φορά διαφορετικό.

3.5 Πρωτόκολλα Επικοινωνίας

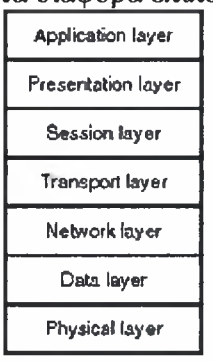
Μια τυπική έξυπνη κάρτα έχει την δυνατότητα να μεταφέρει και να λαμβάνει 115,200 bits το δευτερόλεπτο. Ωστόσο ο ρυθμός μεταφοράς των δεδομένων περιορίζεται στα 9,6000 bps αφού εκ των πραγμάτων, συνήθως μεταφέρεται μικρή ποσότητα δεδομένων και εκείνο που έχει πρωταρχική σημασία είναι η αξιοπιστία του καναλιού επικοινωνίας και όχι η ταχύτητα του.
Το πρώτο βήμα που πρέπει να γίνει είναι ο μηχανισμός answer- to- reset ATR (περιγράφεται παρακάτω) με τον οποίο εγκαθίσταται το βασικό μονοπάτι επικοινωνίας μεταξύ του αναγνώστη και της έξυπνης κάρτας **[6].** Το μονοπάτι αυτό είναι ένα φυσικό κανάλι half-duplex που σημαίνει ότι είτε η κάρτα μπορεί να "μιλά" και ο αναγνώστης να "ακούει" είτε το αντίστροφο. Συνεπώς το τερματικό και η κάρτα πρέπει να είναι συγχρονισμένα και να συμφωνούν στο ποιος έχει προτεραιότητα να στείλει δεδομένα. Αν την ίδια χρονική στιγμή αποφασίσουν και οι δύο πλευρές να στείλουν δεδομένα τότε αυτά θα χαθούν, ενώ δημιουργείται

κατάσταση αδιεξόδου στην περίπτωση που η μια πλευρά περιμένει την άλλη και καμία δεν αποφασίζει να στείλει δεδομένα. Τα δεδομένα που μεταφέρονται ή λαμβάνονται από την κάρτα αποθηκεύονται προσωρινά σε ένα buffer (είναι μια αρκετά περιορισμένη σε μέγεθος μνήμη RAM). Η επικοινωνία γίνεται με την ανταλλαγή μηνυμάτων σε κάθε ένα από τα οποία μεταφέρεται ένα πακέτο δεδομένων μεγέθους 10 έως 100 bytes. Η μορφή αυτών των μηνυμάτων καθορίζεται από διεθνή πρότυπα (ISO, GEN standards) ενώ τα επικρατέστερα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται είναι τα T=0 και T=1 .
Στη συνέχεια τα πρωτόκολλα του επιπέδου εφαρμογής επικοινωνούν με τι εφαρμογές της κάρτας και του αναγνώστη χρησιμοποιώντας το φυσικό επίπεδο διασύνδεσης. Το ISO/IEC 7816-4 standard έχει ορίσει δύο τέτοια πρωτόκολλα επιπέδου εφαρμογής : Το ένα έχει σαν στόχο την εξασφάλιση της συνέπειας του συστήματος αρχείων τόσο κατά την αποθήκευση όσο και κατά την ανάκτηση πληροφοριών στο/από το εσωτερικό της κάρτας. Η εφαρμογή διεπαφής (application programming interface) που συνιστά αυτό το πρωτόκολλο περικλείει ένα σύνολο κλήσεων εντολών μέσω των οποίων επιτυγχάνεται η διαχείριση των αρχείων, δηλαδή η επιλογή ενός αρχείου με βάση το όνομα του, η εγγραφή / ανάγνωση ενός αρχείου κ.τ.λ.. Το δεύτερο πρωτόκολλο έχει την ίδια μορφή με το παραπάνω με τη διαφορά ότι είναι αρμόδιο για τις υπηρεσίες ασφάλειας της κάρτας. Έτσι μέσω αυτού επιτρέπεται στη συσκευή ανάγνωσης και στην έξυπνη κάρτα να αυθεντικοποιούν την ταυτότητα τους (ο ένας στον άλλο) και επιπλέον παρέχονται όλοι οι απαραίτητοι μηχανισμοί που εξασφαλίζουν την εμπιστευτικότητα των δεδομένων που ανταλλάσσονται.
Για την υποστήριξη των παραπάνω πρωτοκόλλων του επιπέδου εφαρμογής έχει οριστεί από ISO/IEC 7816-4 μία δομή μηνύματος μέσω της οποίας ανταλλάσσονται μεταξύ της συσκευής ανάγνωσης και της έξυπνης κάρτας, οι κλήσεις συναρτήσεων μαζί με τις σχετικές παραμέτρους. Αυτή η δομή του μηνύματος χαρακτηρίζεται από τις λεγόμενες APDUs (application protocol data units) μονάδες δεδομένων οι οποίες μεταβιβάζονται μεταξύ αναγνώστη και κάρτας μέσω των πρωτοκόλλων του επιπέδου διασύνδεσης (δηλαδή είτε μέσω του T=0 πρωτοκόλλου είτε μέσω του T=1).

3.5.1 Πρωτόκολλα επιπέδου σύνδεσης Δεδομένων - Link Level Protocols

Μιλώντας για πρωτόκολλα επικοινωνίας θα πρέπει αρχικά να αναφερθούμε στο μοντέλο αναφοράς OSI, το οποίο προτάθηκε το 1970 από το Διεθνή Οργανισμό τυποποίησης (International Organization for Standardization, ISO) και αποτελεί ένα πρότυπο αναφοράς για τη διασύνδεση ανοικτών συστημάτων (open systems interconnection reference model-OSI model). Το μοντέλο αναφοράς OSI είναι μια διαστρωματωμένη αρχιτεκτονική αποτελούμενη από 7 επίπεδα τα οποία απεικονίζονται παρακάτω και παρέχει έναν αποτελεσματικό τρόπο επικοινωνίας μεταξύ απομακρυσμένων εφαρμογών. Στο πρότυπο OSI υπάρχει αυστηρός διαχωρισμός μεταξύ των επιπέδων κάθε ένα από τα οποία παρέχει και μια υπηρεσία η οποία χρησιμοποιείται από το ακριβώς ανώτερο επίπεδο. Τα περισσότερα δημοφιλή δίκτυα διαθέτουν μια αρχιτεκτονική παρόμοια με αυτή του

μοντέλου OSI. Έτσι και στην περίπτωση των πρωτοκόλλων T=0 και T=1, το δεύτερο αντιστοιχεί στο επίπεδο σύνδεσης δεδομένων (data link layer) του OSI ενώ το πρώτο συνδυάζει στοιχεία από τα διάφορα επίπεδα.

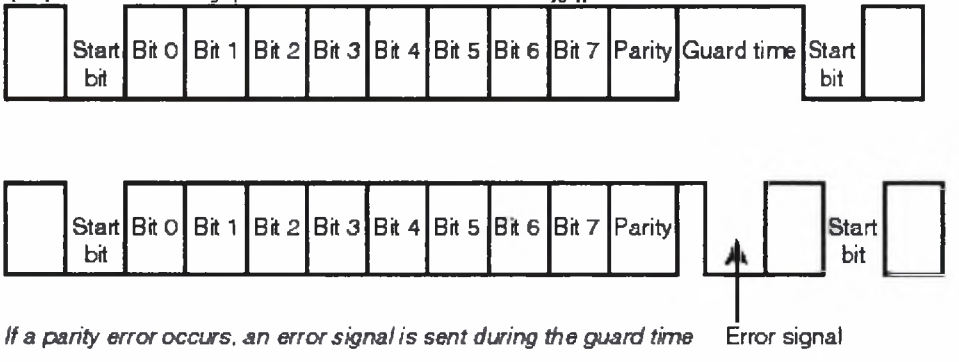


3.10 Μοντέλο αναφοράς OSI

Αναφορικά με την επικοινωνία της συσκευής ανάγνωσης και της έξυπνης κάρτας η διαδικασία έχει ως ακολούθως : Από την πλευρά του αναγνώστη στέλνεται μια εντολή συμπεριλαμβάνοντας και κάποια δεδομένα που χρειάζονται για την εκτέλεση της από την κάρτα. Στη συνέχεια η κάρτα εκτελεί την εντολή και στέλνει το αποτέλεσμα πίσω στον αναγνώστη. Οι δομές των δεδομένων που ανταλλάσσονται αναφέρονται ως transmission protocol data units (TPDUs). Αυτές οι δομές δεδομένων διαφέρουν ανάλογα με το πρωτόκολλο που χρησιμοποιείται (T=0 ή T=1).

3.5.2 Το πρωτόκολλο T=0

Πρόκειται για byte-oriented πρωτόκολλο, που σημαίνει ότι η μονάδα πληροφορίας που μεταφέρεται διαμέσου του καναλιού επικοινωνίας είναι το byte και επιπλέον η διαχείριση λαθών γίνεται ανά ένα byte κάθε φορά. Ο εντοπισμός λαθών γίνεται κοιτάζοντας το bit ισότητας (parity bit) σε κάθε byte που μεταφέρεται μεταξύ έξυπνης κάρτας και αναγνώστη. Κάθε byte πληροφορίας που μεταφέρεται απαιτεί τη χρήση 10 bits όπως φαίνεται στο ακόλουθο σχήμα:



3.11 πρωτόκολλο T=0

Το bit ισότητας τίθεται ίσο με το 0 ή 1 έτσι ώστε ο συνολικός αριθμός των bits που είναι ίσα με τη μονάδα (ανά χαρακτήρα κάθε φορά) να είναι ζυγός αριθμός. Η συσκευή ανάγνωσης κοιτάζει τις τιμές των bits που μεταφέρονται προτού δει την

τιμή του bit ισότητας και αποφασίζει ποια θα πρέπει να είναι η σωστή τιμή του. Αν η τιμή αυτή δεν είναι η αναμενόμενη τότε σημαίνει ότι υπήρξε λάθος κατά τη μεταφορά του συγκεκριμένου byte και κατά συνέπεια εκτελείται μια διαδικασία ανάκαμψης του λάθους. Η διαδικασία ανάκαμψης προκαλείται από την πλευρά του παραλήπτη, ο οποίος ανιχνεύοντας το λάθος στέλνει ένα σήμα στον αποστολέα με το οποίο του ζητά να επαναλάβει τη μεταφορά του byte στο οποίο εντοπίσθηκε το λάθος. Αυτό γίνεται προκαλώντας καθυστέρηση στην γραμμή εισόδου / εξόδου, οπότε ο αποστολέας περιμένει τόσο χρόνο όσο απαιτείται για την αποστολή 2 bytes και αν αυτός ο χρόνος περάσει χωρίς να λάβει κάποια απάντηση από τον παραλήπτη, καταλαβαίνει ότι υπήρξε λάθος και ξαναστέλνει το byte. Έτσι εάν η ανίχνευση λαθών και οι μηχανισμοί ανάκαμψης συμβαίνουν σπάνια σημαίνει ότι το κανάλι έχει "τέλεια συμπεριφορά" ενώ στην αντίθετη περίπτωση μπορεί να οδηγήσει σε απώλεια συγχρονισμού μεταξύ αποστολέα και παραλήπτη. Σε αυτή την περίπτωση η κάρτα είναι προγραμματισμένη να μεταβαίνει σε κατάσταση απραξίας και να μην αποκρίνεται στις εντολές που στέλνει ο αναγνώστης. Εάν ο αναγνώστης αντιληφθεί πρώτος αυτή την κατάσταση στέλνει σήμα στην κάρτα με το οποίο εξαναγκάζει το πρωτόκολλο επικοινωνίας να ξεκινήσει από την αρχή.

Έχει ήδη αναφερθεί ότι τα δεδομένα που ανταλλάσσονται μεταξύ κάρτας και αναγνώστη καλούνται TPDUs και συγκεκριμένα στο πρωτόκολλο T=0 αποτελούνται από δύο διαφορετικές δομές :

- *Command*, η οποία στέλνεται από τον αναγνώστη στην κάρτα και
- *Response*, η οποία στέλνεται από την κάρτα στον ανγνώστη.

Η εντολή (command) που στέλνεται από τον αναγνώστη στην κάρτα περιλαμβάνει πέντε πεδία μήκους ενός byte το καθένα, και είναι τα ακόλουθα:

- ***CLA***: χρησιμοποιείται για την εγκατάσταση ενός συνόλου εντολών και συνήθως αναφέρεται ως class designation, περιγράφει ουσιαστικά σε ποια κατηγορία ανήκει το σύνολο των εντολών.
- ***INS***: προσδιορίζει μια συγκεκριμένη εντολή η οποία ανήκει στο σύνολο των εντολών που έχει καθοριστεί από το πεδίο CLA. Συνήθως αναφέρεται ως instruction designation.
- ***P1***: ορίζει τον τρόπο διευθυνσιοδότησης που χρησιμοποιείται από την [CLA, INS] εντολή.
- ***P2***: επίσης ορίζει τον τρόπο διευθυνσιοδότησης που χρησιμοποιείται από την [CLA, INS] εντολή.
- ***P3***: ορίζει τον αριθμό των bytes των δεδομένων που μεταφέρονται είτε στην κάρτα είτε από την κάρτα αφού αυτή εκτελέσει την [CLA, INS] εντολή.

Για κάθε εντολή TPDU που στέλνεται από τον αναγνώστη στην κάρτα, η κάρτα αποστέλλει μια απάντηση (response) TPDU η οποία αποτελείται από τρία υποχρεωτικά πεδία και ένα προαιρετικό, κάθε ένα από τα οποία έχει μήκος του ενός byte. Καθένα από τα πεδία περιγράφεται ακολούθως :

- ***ACK***: υποδηλώνει ότι η κάρτα έχει λάβει την [CLA, INS] εντολή.
- ***NULL***: χρησιμοποιείται για τον έλεγχο ροής του καναλιού εισόδου / εξόδου. Με το πεδίο αυτό η κάρτα σηματοδοτεί στον αναγνώστη ότι επεξεργάζεται

ακόμη την εντολή που στάλθηκε και άρα θα πρέπει να περιμένει πριν στείλει την επόμενη εντολή.

- ***SW1***: δείχνει σε τι κατάσταση βρίσκεται η απάντηση που στέλνει η κάρτα για την τρέχουσα εντολή.
- ***SW2***: είναι προαιρετικό και αν περιλαμβάνεται στην απάντηση που στέλνει η κάρτα επίσης δείχνει την κατάσταση της απάντησης που στέλνει η κάρτα για την τρέχουσα εντολή.

3.5.3 Το πρωτόκολλο T=1

Πρόκειται για ένα πρωτόκολλο τύπου block – oriented που σημαίνει ότι τα δεδομένα που μεταφέρονται μεταξύ έξυπνης κάρτας και αναγνώστη είναι οργανωμένα σε blocks. Ένα block μπορεί να περιέχει μια APDU μιας συγκεκριμένης εφαρμογής. Σημαντικό πλεονέκτημα αυτής της υλοποίησης είναι ότι παρέχει τέλεια επικοινωνία μεταξύ του επιπέδου σύνδεσης δεδομένων και του επιπέδου εφαρμογής. Ωστόσο το πρωτόκολλο T=1 μειονεκτεί στην περίπτωση ύπαρξης λάθους στο block που μεταφέρεται αφού η διαδικασία ανίχνευσης και διόρθωσης του είναι αρκετά σύνθετη σε σχέση με τις αντίστοιχες διαδικασίες του πρωτοκόλλου T=0. Για την ανίχνευση και την ανάκαμψη του λάθους χρησιμοποιείται ο αλγόριθμος CRC και ορίζεται από το ISO 3309 standard. Έτσι μόλις ανιχνευθεί ένα λάθος στο block που μεταφέρεται, ο αποστολέας στέλνει σήμα στον αποστολέα ώστε αυτός να επαναλάβει την μεταφορά του ίδιου block.
Το T=1 πρωτόκολλο χρησιμοποιεί τρεις διαφορετικούς τύπους block οι οποίοι έχουν την ίδια δομή, επιτελώντας όμως διαφορετικούς σκοπούς ο καθένας.

- ***Information block:*** χρησιμοποιείται για τη μεταφορά πληροφορίας μεταξύ του λογισμικού της εφαρμογής που τρέχει στην έξυπνη κάρτα και του λογισμικού της εφαρμογής του αναγνώστη.
- ***Receieve Ready block:*** χρησιμοποιείται για την αποστολή θετικών ή αρνητικών επιβεβαιώσεων. Μια θετική επιβεβαίωση σημαίνει ότι το block στάλθηκε σωστά ενώ η αρνητική σημαίνει ότι το block που στάλθηκε περιείχε κάποιο λάθος.
- ***Supervisory block:*** χρησιμοποιείται για τη μεταφορά πληροφοριών ελέγχου μεταξύ κάρτας και συσκευής ανάγνωσης.

Τα συστατικά του πρωτοκόλλου T=1 παρουσιάζονται στον ακόλουθο πίνακα και εξηγούνται στη συνέχεια.

Prologue Field			Information Field	Epilogue Field
Node Address	Protocol Control Byte	Length	APDU	Error Detection
NAD	PCB	LEN	Data Length	LRC/CRC
1 Byte	1 Byte	1 Byte	0 to 254 Bytes	1 or 2 Bytes

3.12 Συστατικά του T=1 πρωτοκόλλου

Κάθε block αποτελείται από τρία πεδία :

- ***Prologue field :*** περιέχεται υποχρεωτικά σε ένα block, έχει μήκος τριών bytes και αποτελείται από τα ακόλουθα τρία στοιχεία μήκους 1 byte το καθένα:
  - ***NAD:*** Node address
  - ***PCB:*** Protocol ControlByte
  - ***LEN:*** Length
- ***Information field:*** συμπεριλαμβάνεται προαιρετικά σε ένα block και έχει μήκος τουλάχιστον 254 bytes
- ***Epilogue field:*** η ύπαρξη του είναι υποχρεωτική και το μήκος κυμαίνεται από ένα έως δύο bytes.

Το στοιχείο NAD χρησιμοποιείται για να προσδιορίσει τις διευθύνσεις τόσο της πηγής όσο και του προορισμού του block. Η ύπαρξη του είναι ιδιαίτερα σημαντική στην περίπτωση όπου χρειάζεται να υποστηριχθούν πολλαπλές λογικές συνδέσεις μεταξύ κάρτας και αναγνώστη και περιέχει δύο υπο-επίπεδα:

- ***SAD:*** δείχνει την διεύθυνση της πηγής και αποτελείται από τα τρία χαμηλότερης προτεραιότητας bits, του byte του στοιχείου NAD
- ***DAD:*** δείχνει τη διεύθυνση του προορισμού και αποτελείται από το πέμπτο έως το έβδομο bit του byte του στοιχείου NAD.

Σε περιπτώσεις όπου δεν χρησιμοποιούνται πολλαπλά λογικά κανάλια τα bits του πεδίου NAD τίθενται όλα ίσα με το μηδέν. Τα υπόλοιπα δύο bits του πεδίου αυτού χρησιμοποιούνται για τον έλεγχο της V_{PP} (EEPROM programming power).

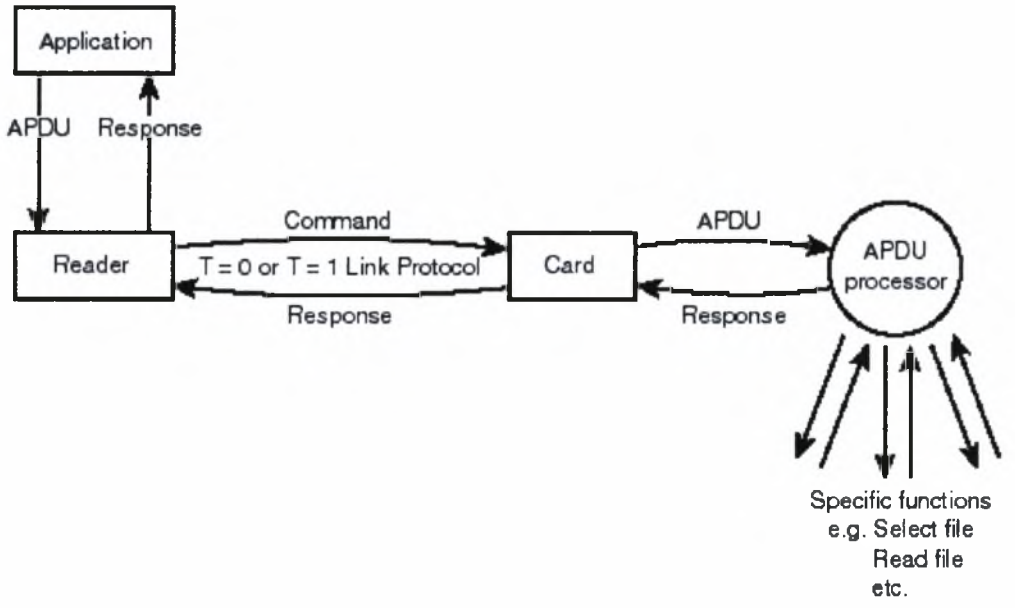
Το στοιχείο PCB χρησιμοποιείται για να δηλώσει τον τύπο του block που μεταφέρεται, δηλαδή αν είναι information, receive ready ή supervisory block. Τα δύο υψηλότερης προτεραιότητας bits του byte του PCB πεδίου χρησιμοποιούνται για να δηλώσουν τους παραπάνω διαφορετικούς τύπους block:

- Αν το υψηλότερης προτεραιότητας bit είναι ίσο με το μηδέν σημαίνει ότι το block είναι τύπου **information**.
- Αν και τα δύο υψηλότερης προτεραιότητας bits είναι ίσα με τη μονάδα σημαίνει ότι το block είναι τύπου **supervisory**.

- Τέλος αν το bit υψηλότερης προτεραιότητας ισούται με τη μονάδα και το αμέσως επόμενο ισούται με το μηδέν τότε το block ανήκει στην κατηγορία των **receive ready** blocks.

3.5.4 Πρωτόκολλα επιπέδου Εφαρμογής- Application Level Protocols

Χρησιμοποιούνται για να ορίσουν τις συναρτήσεις με τι οποίες οι συσκευές ανάγνωσης μπορούν να διαχειριστούν το σύστημα αρχείων της κάρτας και να προσπελαύνουν τα αρχεία της, καθώς επίσης και για τον ορισμό συναρτήσεων οι οποίες παρέχουν ελεγχόμενη πρόσβαση στα αρχεία έτσι ώστε να πληρούνται όλες οι προδιαγραφές ασφάλειας. Το πρωτόκολλο εφαρμογής βασίζεται σε μια δομή block η οποία καλείται APDU. Αυτές οι μονάδες δεδομένων APDUs ανταλλάσσονται μεταξύ κάρτας και συσκευής ανάγνωσης χρησιμοποιώντας είτε το T=1 είτε το T=0 πρωτόκολλο σύνδεσης δεδομένων. Η μετάφραση των APDUs γίνεται από ένα συστατικό του λογισμικού της έξυπνης κάρτας το οποίο στη συνέχεια εκτελεί την συγκεκριμένη εντολή. Η διαδικασία αυτή υλοποιείται στο ακόλουθο σχήμα:



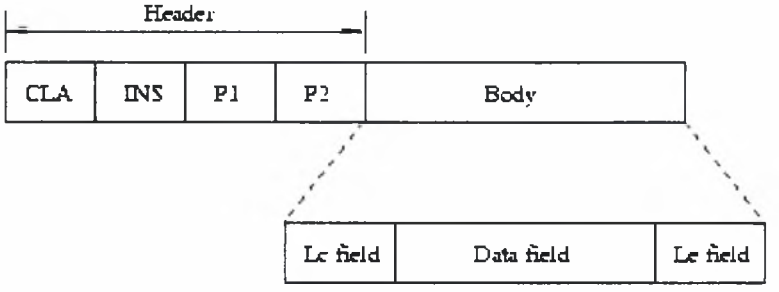
3.13 Αρχιτεκτονική επικοινωνίας σε επίπεδο εφαρμογών

Τα APDUs έχουν δομή μηνυμάτων όπως αυτή ορίζεται από το ISO-7816-4 standard και περιέχουν είτε μια εντολή, ίσως και δεδομένα, που στέλνει ο αναγνώστης στην κάρτα , είτε την απάντηση που στέλνει η κάρτα στον αναγνώστη ως απόκριση της εντολής που εκτέλεσε.

3.5.4.1 Δομή των μηνυμάτων APDU- ISO 7816-4 APDU

Τα μηνύματα που χρησιμοποιούνται για να υποστηρίξουν τα πρωτόκολλα εφαρμογής διαχωρίζονται σε δύο κατηγορίες : στα *command APDU* που είναι οι

εντολές που στέλνει ο αναγνώστης στην κάρτα και στα *response APDU* που είναι η απάντηση που στέλνει η κάρτα πίσω στον αναγνώστη.
Η command APDU αποτελείται από δύο βασικά πεδία την επικεφαλίδα και το κυρίως σώμα. Καθένα από αυτά τα πεδία περιέχουν όπως φαίνεται παρακάτω και κάποια υπό-πεδία.

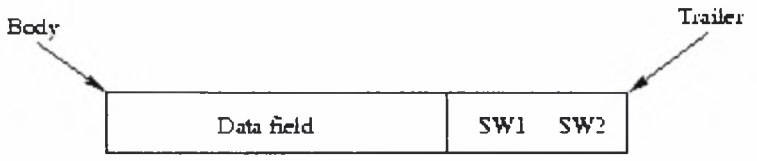


3.14 Δομή της command APDU

Όπως παρατηρούμε η επικεφαλίδα περιλαμβάνει τα πεδία CLA, INS, P1, P2. Όμοια με το πρωτόκολλο T=0 που μελετήθηκε παραπάνω, Τα πεδία CLA, INS χρησιμοποιούνται, το πρώτο για τον ορισμό της τάξης στην οποία ανήκει η εφαρμογή και του συνόλου των εντολών της συγκεκριμένης τάξης, ενώ το δεύτερο προσδιορίζει μια συγκεκριμένη εντολή η οποία ανήκει στο σύνολο των εντολών που έχει καθοριστεί από το πεδίο CLA. Τα πεδία P1, P2 χρησιμοποιούνται για τον περαιτέρω προσδιορισμό μιας [CLA,INS] εντολής, παρέχοντας έτσι όλες τις απαραίτητες διευκρινίσεις για την εκτέλεση της. Το πεδίο body του APDU μηνύματος έχει μεταβλητό μέγεθος και δομή και χρησιμοποιείται είτε για τη μεταφορά πληροφορίας στον APDU επεξεργαστή είτε για τη μεταφορά πληροφορίας από την κάρτα στον αναγνώστη. Τα υπό-πεδία του πεδίου body είναι : το **Lc field** το οποίο ορίζει τον αριθμό των bytes της εντολής που θα σταλούν στην κάρτα και περιέχει επίσης το μήκος του πεδίου data field, το **Data field** που περιλαμβάνει δεδομένα τα οποία πρέπει να μεταφερθούν στην κάρτα με στόχο να επιτρέψουν στον APDU processor της κάρτας να εκτελέσει την εντολή του APDU μηνύματος και τέλος το **Le field** το οποίο ορίζει τον αριθμό των bytes της απάντησης, που θα επιστρέψει η κάρτα στον αναγνώστη. Το πεδίο body του μηνύματος APDU μπορεί να έχει μια από τις ακόλουθες τέσσερις μορφές :

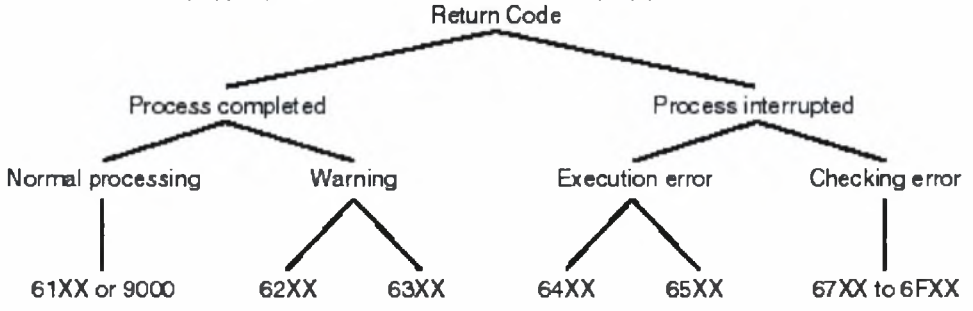
- ***Περίπτωση 1η*** :Δεν μεταφέρονται δεδομένα από ή προς την κάρτα και έτσι το APDU μήνυμα περιλαμβάνει μόνο την επικεφαλίδα (δηλαδή το πεδίο header).
- ***Περίπτωση 2η*** :Δεν μεταφέρονται δεδομένα προς την κάρτα, με τη διαφορά όμως ότι η κάρτα μπορεί να επιστρέψει δεδομένα. Έτσι στο πεδίο body περιέχεται μόνο το υπό-πεδίο *Le field.*
- ***Περίπτωση 3η*** :Δεδομένα μεταφέρονται στην κάρτα αλλά δεν επιστρέφονται από αυτή. Έτσι στο πεδίο body περιέχονται τα υπό-πεδία *Lc field και data field.*
- ***Περίπτωση 4η*** : Στο πεδίο body περιέχονται και τα τρία υπό-πεδία *Lc field, Le field και data field* αφού δεδομένα μεταφέρονται τόσο από τον αναγνώστη όσο και από την κάρτα.

Η response APDU είναι δομικά πολύ πιο απλή και όπως δείχνει η επόμενη εικόνα αποτελείται από δύο πεδία : το body και το trailer.



3. 15 Δομή της response APDU

Το πεδίο body μπορεί να είναι κενό ή να περιέχει το πεδίο data field, ανάλογα με την τρέχουσα εντολή (της οποίας αποτελεί απάντηση) και ανάλογα με την επιτυχή ή μη επιτυχή εκτέλεση της από τον APDU processor της κάρτας. Στην περίπτωση που περιλαμβάνεται το πεδίο Data Field το μήκος του καθορίζεται από το Le field της σχετιζόμενης εντολής. Αναφορικά με το πεδίο trailer αυτό μπορεί να περιλαμβάνει τα πεδία SW1, SW2 τα οποία περιέχουν πληροφορία σχετικά με την κατάσταση στην οποία βρίσκεται η εκτέλεση της εντολής. Αυτό γίνεται με την επιστροφή κάποιων κωδικών από την κάρτα στον αναγνώστη, οι οποίοι σύμφωνα με το ISO 7816-4 standard περιγράφονται στο ακόλουθο διάγραμμα:



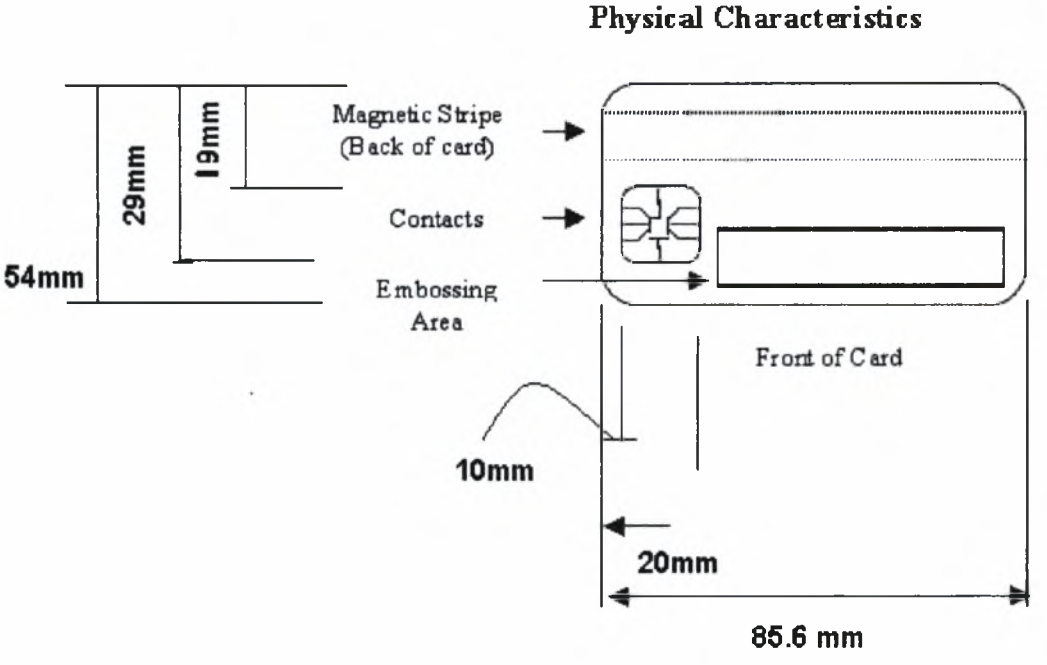
3.16 Κώδικας που επιστρέφεται

Χρησιμοποιούνται δύο bytes, το ένα για την περιγραφή της κατάστασης (δηλαδη Normal processing, Warning, Execution error, Checking error) και το άλλο για την επιστροφή ενός συγκεκριμένο κωδικού (δηλαδή ένα από τα φύλλα του παραπάνω δένδρου).

3.6 Πρότυπα Έξυπνων Καρτών – Contact Card Standards (ISO/IEC 7816)

ISO/IEC 7816-1:1998 Physical Characteristics of IC cards, physical dimensions

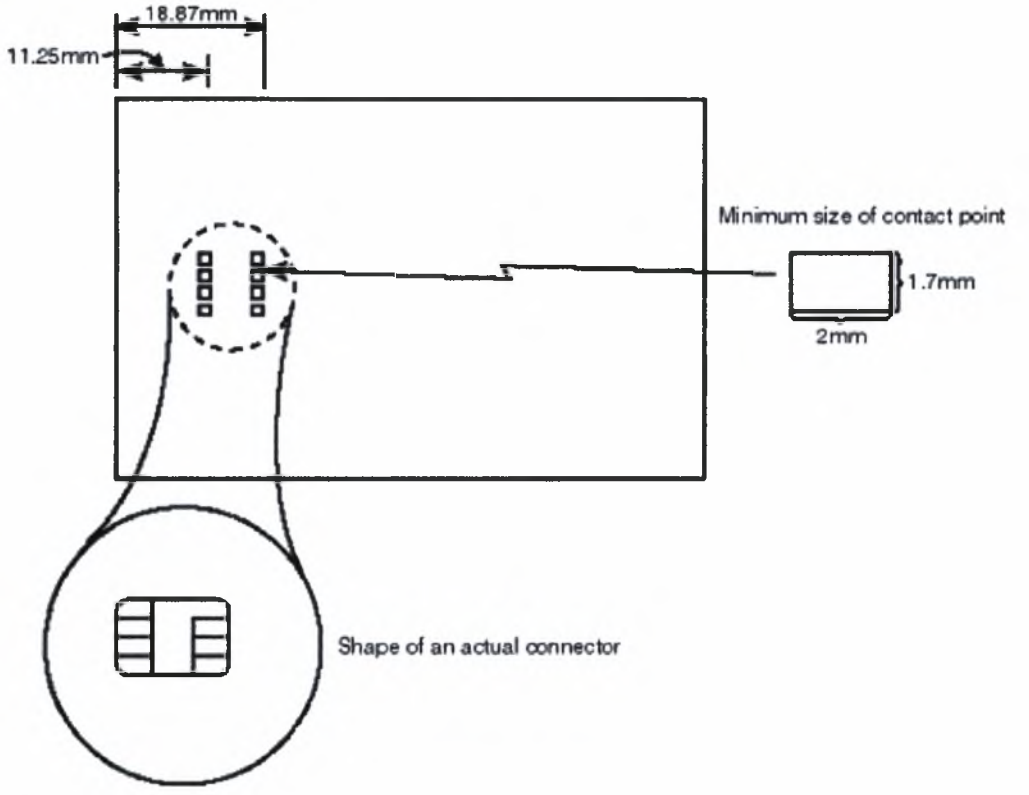
Καθορίζει τις φυσικές διαστάσεις της κάρτας (πλάτος, μήκος και πάχος), τα οποία είναι ίδια με αυτά μιας κανονικής πιστωτικής κάρτας. Επιπρόσθετα χαρακτηριστικά που αναφέρονται στο ISO/IEC 7816 –1 σχετίζονται με ακτίνες X, σημεία επαφής της επιφάνειας, μηχανική ισχύς, ηλεκτρομαγνητικά χαρακτηριστικά και στατικό ηλεκτρισμό [7].



3.17 Φυσικές Διαστάσεις

ISO/IEC 7816-2:1999 Position of Module and Contacts on IC cards

Ορίζει τις διαστάσεις και την περιοχή στην οποία τοποθετούνται οι επαφές. Υπάρχουν 8 συνολικά επαφές οι οποίες τοποθετούνται στο μπροστινό μέρος της κάρτας και αναφέρονται ως C1 έως και C8. Το πρότυπο βέβαια δεν ορίζει πως οι επαφές πρέπει να τοποθετούνται στο μπροστινό μέρος της κάρτας, μπορούν να τοποθετηθούν και στο πίσω μέρος όπου βρίσκεται και η μαγνητική ταινία αρκεί να μην την τέμνουν. Κάποιες από αυτές συνδέονται με το chip που βρίσκεται ενσωματωμένο στην κάρτα ενώ κάποιες άλλες υπάρχουν απλά για μελλοντική χρήση. Η τοποθεσία, το μέγεθος και το σχήμα των επαφών υλοποιούνται στο ακόλουθο σχήμα. Ωστόσο οι πιο πρόσφατες έξυπνες κάρτες υιοθετούν ένα διαφορετικό πρότυπο σχετικά με τη θέση των επαφών σύμφωνα με το οποίο οι επαφές τοποθετούνται στο επάνω – αριστερά μέρος την μπροστινή πλευράς της κάρτας. Αυτές οι κάρτες κυκλοφόρησαν πρωταρχικά στην Ευρώπη και ήταν κυρίως πιστωτικές και χρεωστικές κάρτες.

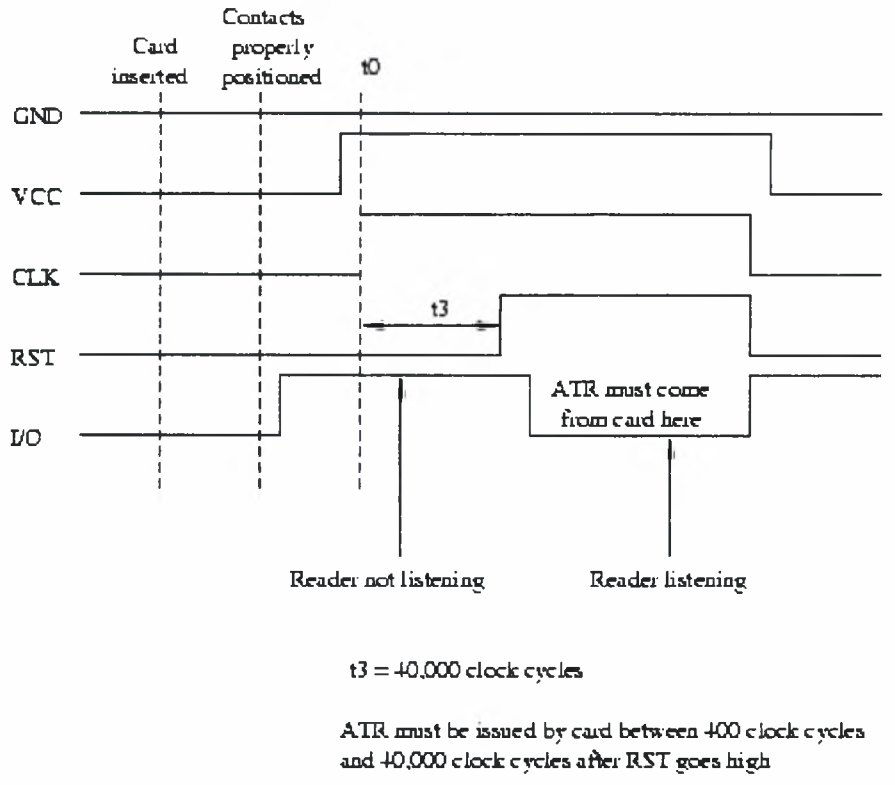


3.18 Τοποθεσία, μέγεθος και σχήμα επαφών

ISO/IEC 7816-3:1997 Electrical specifications and communication protocols

Ελέγχει τα χαρακτηριστικά των ηλεκτρικών σημάτων και των πρωτοκόλλων μεταφοράς καθώς και την τυποποίηση της κάρτας «answer to reset». Περιγράφει επίσης τη σχέση έξυπνης κάρτας και τερματικού που είναι μια σχέση master (τερματικό)-slave (έξυπνη κάρτα).Η επικοινωνία ξεκινά από τον αναγνώστη ο οποίος στέλνει σήμα στην έξυπνη κάρτα διαμέσου των επαφών που ορίστηκαν στο αμέσως προηγούμενο πρότυπο, και συνεχίζεται με την απόκριση της κάρτας. Το κανάλι επικοινωνίας δεν είναι πολυνηματικό που σημαίνει ότι από τη στιγμή που ο αναγνώστης στείλει μια εντολή στην έξυπνη κάρτα αυτός μπλοκάρεται μέχρι να παραλάβει την απάντηση από την κάρτα.

Όταν η κάρτα εισάγεται στον αναγνώστη καμία από τις επαφές τις δεν τροφοδοτείται με ρεύμα αρχικά. Το chip της κάρτας θα μπορούσε να καταστραφεί στην περίπτωση που οι λάθος επαφές τροφοδοτηθούν με ρεύμα και αυτή η κατάσταση μπορεί εύκολα να προκύψει αν η κάρτα εισαχθεί σε ήδη τροφοδοτούμενα με ρεύμα, σημεία επαφής. Έτσι οι επαφές παραμένουν χωρίς ρεύμα μέχρι να αποφασιστεί ότι αυτές έχουν τοποθετηθεί στα σωστά σημεία επαφής. Όταν ο αναγνώστης διαπιστώσει ότι η κάρτα είναι σωστά τοποθετημένη στα σημεία επαφής τότε παρέχεται σε αυτή ρεύμα. Η διαδικασία περιγράφεται στο παρακάτω χρονικό διάγραμμα.



3.19 Χρονικό διάγραμμα έξυπνης κάρτας

Οι επαφές αρχικά περνούν από μια κατάσταση η οποία λέγεται "idle state" και χαρακτηρίζεται από την τιμή τάσης της επαφής V_{cc}, η οποία στην idle state έχει τιμή ίση με 5 V.Στη συνέχεια ένα σήμα επαναφοράς στέλνεται από την κάρτα μέσω της επαφής RST και αυτό σηματοδοτεί ότι η κάρτα πρέπει να ξεκινήσει τη διαδικασία της αρχικοποίησης. Οι λειτουργίες αρχικοποίησης μπορεί να είναι διαφορετικές για κάθε κάρτα ωστόσο αυτές πάντα πρέπει να έχουν ως αποτέλεσμα την αποστολή της λεγόμενης απάντησης επαναφοράς (answer to reset (ATR)). Το ATR είναι μια ακολουθία χαρακτήρων και επιστρέφεται από την κάρτα, ως ένδειξη επιτυχούς ενεργοποίησης της. Το μέγιστο μήκος του είναι 33 bytes και η αποστολή πρέπει να γίνει σε συγκεκριμένο χρόνο. Δηλαδή το πρώτο byte πρέπει να ληφθεί από τον αναγνώστη σε διάρκεια 40,000 κύκλων ρολογιού και ο ρυθμός μεταφοράς των δεδομένων είναι 1 byte το δευτερόλεπτο.

ISO/IEC 7816-4:1995 Command set for microprocessor cards

Ελέγχει:

- τις εντολές μεταξύ διαφορετικών βιομηχανιών,
- τις προδιαγραφές του APDU,
- τα ιστορικά χαρακτηριστικά «Answer to Reset»,
- τις δομές αρχείων,
- τις μεθόδους πρόσβασης,
- και την ασφαλή αποστολή μηνυμάτων

ISO/IEC 7816-5:1994 Application identification

Καταχωρητικό σύστημα το οποίο κατέχει στοιχεία για αναγνώριση εφαρμογών και επιτρέπει στα διάφορα τερματικά να διαλέξουν μια εφαρμογή από την κάρτα.

ISO/IEC 7816-6:1996 Inter-industry data elements

Ορίζει τα στοιχεία δεδομένων που προορίζονται για ανταλλαγή.

ISO/IEC 7816-7:1999 Inter-industry commands for Structured Card Query Language (SCQL)

Ορίζει την SCQL και τις εντολές οι οποίες επιτρέπουν την διαχείριση σχεσιακών βάσεων δεδομένων στην έξυπνη κάρτα.

ISO/IEC DIS 7816-8 Security related inter-industry commands

Ορίζει θέματα ασφάλειας για εντολές που χρησιμοποιούνται μεταξύ βιομηχανιών.

ISO/IEC DIS 7816-9 Additional inter-industry commands and security attributes

Ορίζει επιπρόσθετα θέματα ασφάλειας για εντολές και χαρακτηριστικά που χρησιμοποιούνται μεταξύ βιομηχανιών.

ISO/IEC DIS 7816-10 Electronic signals and answer to reset for synchronous Cards

Ορίζει τα ηλεκτρονικά σήματα και την «answer to reset» για έξυπνες κάρτες που χρησιμοποιούν σύγχρονο τρόπο μετάδοσης δεδομένων.

ISO 10373 Card Testing

Αναφέρεται στις μεθόδους ελέγχου που χρησιμοποιούνται για τον έλεγχο των έξυπνων καρτών, όπως λύγισμα της κάρτας σε οριζόντιο και κάθετο άξονα, έλεγχος σε διαφορετικές θερμοκρασίες και υγρασία κ.τ.λ.

Πρωτόκολλα Ανταλλαγής Μηνυμάτων – Message Exchange Protocols

ISO 8583 Financial transaction card originated messages - interchange message specification

Πρότυπα μηνυμάτων που προέρχονται από κάρτες οικονομικών συναλλαγών – προσδιορισμός και προδιαγραφές ανταλλαγής μηνυμάτων

ISO 9992 Financial transaction cards - messages between the integrated circuit card and the card-accepting device

Πρότυπα καρτών για οικονομικές συναλλαγές – μηνύματα μεταξύ ολοκληρωμένου κυκλώματος καρτών και του αναγνώστη της κάρτας.

Πρότυπα σχετικά με θέματα Ασφάλειας – Security Related Standards

ISO 9564 Banking – PIN management and security

Πρότυπα για τραπεζικές λειτουργίες, διαχείριση και ασφάλεια του προσωπικού κωδικού ασφαλείας.

ISO 9796 IT security techniques – digital signatures

Πρότυπα για τεχνικές ασφάλειας – ψηφιακές υπογραφές

ISO 9797 IT security techniques – data integrity mechanism

Πρότυπα για τεχνικές ασφάλειας – μηχανισμός που ελέγχει την ακρίβεια των καταχωρημένων στοιχείων χρησιμοποιώντας μια κρυπτογραφική λειτουργία.

ISO 9798 IT security techniques – entity authentication

Πρότυπα για τεχνικές ασφάλειας – ταυτοποίηση οντότητας

ISO 10202 Financial transaction cards – security architecture using IC Cards

Πρότυπα καρτών για οικονομικές συναλλαγές

ISO 11568 Banking - key management (retail)

Πρότυπα για τραπεζικές λειτουργίες

3.7 Βιομηχανικά Πρότυπα Έξυπνων Καρτών

3.7.1 EMV

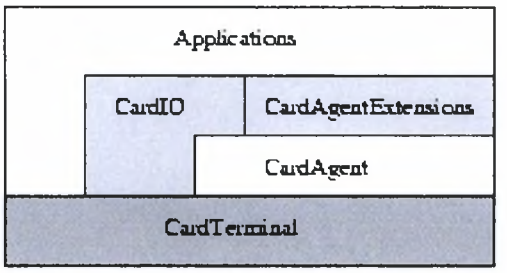
Τρεις τραπεζικοί οργανισμοί Europay International, MasterCard International, και Visa International το 1996 σχημάτισαν μια ομάδα εργασίας από κοινού με σκοπό τη δημιουργία ενός συνόλου τεχνικών προδιαγραφών για την χρησιμοποίηση των έξυπνων καρτών σε εφαρμογές ηλεκτρονικού εμπορίου και ασφαλών πληρωμών. Οι προδιαγραφές EMV περιλαμβάνουν τρία τμήματα :

- Έξυπνη κάρτα
- Τερματικά
- Αλληλεπίδραση μεταξύ εφαρμογών και κάρτας

Ένα μοναδικό χαρακτηριστικό του EMV standard είναι ότι παρέχει προδιαγραφές και για πολύ-εφαρμογικές (multiapplication) κάρτες, δηλαδή έξυπνές κάρτες οι οποίες περιέχουν πάνω από μια εφαρμογή. Αυτό βέβαια απαιτεί μια πολύ καλά ορισμένη μέθοδο ανταλλαγής δεδομένων ώστε να προστατεύονται προσωπικά δεδομένα και η κάθε εφαρμογή να έχει πρόσβαση μόνο σε πληροφορίες χρήσιμες για τη σωστή λειτουργία της. Δηλαδή ο κάτοχος της κάρτας μπορεί να ορίσει ένα σύνολο προσωπικών δεδομένων (όπως όνομα, διεύθυνση, αριθμός τηλεφώνου κ.λ.π) που θα είναι ορατό από όλες τις εφαρμογές ενώ τα υπόλοιπα δεδομένα θα είναι προσβάσιμα μόνο από την εφαρμογή για την οποία είναι απαραίτητα.

3.7.2 OpenCard Framework (OCF)

Αναπτύχθηκε από τις IBM, Netscape, NCI και τη Sun Microsystems **[13].** Πρόκειται για ένα ανοικτό πρότυπο το οποίο δεν περιορίζει τη χρήση των έξυπνων καρτών ενώ αντιθέτως παρέχει διαλειτουργικότητα μεταξύ εφαρμογών έξυπνων καρτών και δικτύων υπολογιστών, laptops κ.λ.π.. Το πλαίσιο εργασίας Open Card Framework έχει αναπτύξει μια αρχιτεκτονική και ένα σύνολο από προγραμματιστικά περιβάλλοντα διεπαφής (APIs) τα οποία δίνουν τη δυνατότητα εγγραφής εφαρμογών γενικού σκοπού με χρήση της γλώσσας προγραμματισμού JAVA, και οι οποίες μπορούν να επικοινωνούν με οποιαδήποτε συσκευή ανάγνωσης ανεξαρτήτου κατασκευαστή με την προϋπόθεση βέβαια να τηρούνται όλες οι προδιαγραφές που ορίζονται. Τα συστατικά που συντελούν την αρχιτεκτονική του Open Card Framework φαίνονται στην ακόλουθη εικόνα:

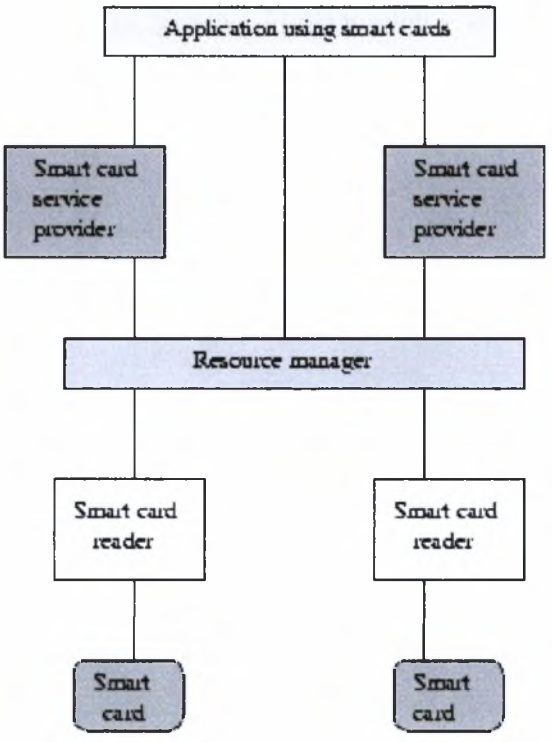


3.20 Open Card Framework αρχιτεκτονική

Με το συστατικό CardTerminal λύνεται το πρόβλημα εξάρτησης των έξυπνων καρτών από τις συσκευές ανάγνωσης. Το Open Card Framework με το συστατικό CardAgent μπορεί να διαχειρίζεται ένα μεγάλο εύρος λειτουργικών συστημάτων έξυπνων καρτών και να υποστηρίζει τις εντολές αυτών. Το CardIO παρέχει την πρόσβαση στο σύστημα αρχείων της κάρτας (ανάγνωση/ εγγραφή αρχείων κ.λ.π.). Τέλος το CardAgentextension χρησιμοποιείται για την υποστήριξη ειδικών εντολών και υπηρεσιών (όπως για παράδειγμα η κρυπτογραφία) και εξαρτάται από τη λειτουργικότητα της κάθε κάρτας.

3.7.3 Personal Computer/Smart Card (PC/SC)

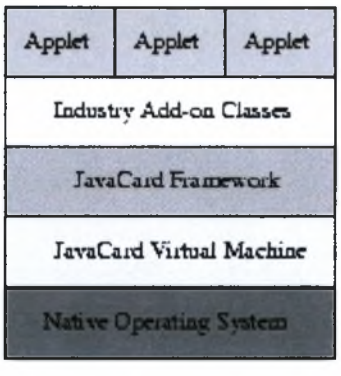
Η Microsoft και άλλες εταιρίες στο χώρο των προσωπικών υπολογιστών και των έξυπνων καρτών (συμπεριλαμβανομένων των Schlumberger, Bull CP8 Transac, Hewlett-Packard, Siemens Nixdorf Information Systems and IBM) θεμελίωσαν το PC/SC WorkGroup το οποίο τελικά πήρε την ονομασία PC/SC και εξασφαλίζει την επικοινωνία των έξυπνων καρτών με προσωπικούς υπολογιστές (Win32-based platforms). Το πρότυπο αυτό δημοσιεύτηκε το 1996 και ονομάζεται Interoperability Specification for ICCs and Personal Computer Systems. Η αρχιτεκτονική PC/SC η οποία υλοποιείται παρακάτω, έχει σχεδιαστεί με τέτοιο τρόπο ώστε να δίνεται η δυνατότητα στους κατασκευαστές έξυπνων καρτών και συσκευών ανάγνωσης να αναπτύσσουν τα προϊόντα τους ανεξάρτητα. Επιπρόσθετα οι προγραμματιστές εφαρμογών έξυπνων καρτών μπορούν να αναπτύξουν εφαρμογές οι οποίες δεν θα εξαρτώνται από ένα συγκεκριμένο τύπο τερματικού έξυπνης κάρτας. Η αρχιτεκτονική αυτή ορίζει μια διεπαφή μεταξύ έξυπνων καρτών και ενός resource manager. Έτσι από την πλευρά της εφαρμογής όλοι οι αναγνώστες συμπεριφέρονται με τον ίδιο τρόπο. Το περιβάλλον διεπαφής (API) που βλέπει η εφαρμογή παρέχεται από τον smart card service provider (SSP). Αυτή μπορεί να εξαρτάται από την κάρτα ή να είναι γενικού σκοπού. Στην πρώτη περίπτωση παρέχεται από τον κατασκευαστή μαζί με την έξυπνη κάρτα.



3.21 PC/SC αρχιτεκτονική

3.7.4 JavaCard

Η Java Card είναι μια έξυπνη κάρτα η οποία έχει τη δυνατότητα να τρέχει προγράμματα γραμμένα σε γλώσσα προγραμματισμού Java **[10].** Για την ανάπτυξη αυτού του προτύπου συνεργάστηκαν δύο μεγάλες εταιρίες οι : Schlumberger και GemPlus. Το πακέτο προδιαγραφών JavaCard 2.0 περιλαμβάνει λεπτομερείς πληροφορίες για τη δημιουργία εικονικής μηχανής JavaCard(virtual machine) και περιβαλλόντων διεπαφής για έξυπνές κάρτες (APIs). Οι ελάχιστες απαιτήσεις συστήματος είναι 16 kilobytes μνήμης ROM (read-only memory), 8 kilobytes EEPROM, και 256 bytes μνήμης RAM (random access memory). Το ακόλουθο σχήμα υλοποιεί τη αρχιτεκτονική της Java Card.



3.22 Αρχιτεκτονική JavaCard

Παρατηρούμε ότι η Java Card VM χτίζεται πάνω σε ένα συγκεκριμένο ολοκληρωμένο κύκλωμα (IC) και ήδη υπάρχων λειτουργικό σύστημα, επικαλύπτοντας την τεχνολογία του κατασκευαστή με τη χρήση μιας κοινής γλώσσας και ενός συστήματος διεπαφής. Το πλαίσιο εργασίας JavaCard Framework περιέχει ένα σύνολο εντολών API κλάσεων για την ανάπτυξη Java Card εφαρμογών ενώ επίσης παρέχει υπηρεσίες συστήματος για αυτές τις εφαρμογές. Δίνεται επίσης η δυνατότητα σε έναν κατασκευαστή να προσθέσει καινούργιες βιβλιοθήκες για την υποστήριξη επιπλέον υπηρεσιών ή για τη βελτίωση του συστήματος ασφάλειας της κάρτας – εφαρμογής. Οι εφαρμογές Java Card καλούνται applets και πολλά applets μπορούν να ανήκουν σε μια κάρτα. Κάθε τέτοιο applet έχει ένα μοναδικό αναγνωριστικό το οποίο ονομάζεται *AID* (application identifier), και ορίζεται από το ISO 7816 standard.
Τέλος είναι σημαντικό να τονιστεί ότι οι έξυπνες κάρτες δεν είναι προσωπικοί υπολογιστές αφού έχουν περιορισμένη μνήμη και υπολογιστική ικανότητα και άρα σε καμία περίπτωση δεν πρέπει να γίνεται σύγκριση του πακέτου προδιαγραφών Java Card 2.0 και της έκδοσης του JDK και των δυνατοτήτων του.

Κεφάλαιο 4

Open Card Framework

Το Open Card Framework αποτελεί το βασικό αντικείμενο αυτού του κεφαλαίου. Πρόκειται για ένα πλαίσιο εργασίας το οποίο παρέχει ένα σύνολο από κλάσεις και μηχανισμούς που χρησιμοποιούνται για την ανάπτυξη εφαρμογών αλληλεπίδρασης με τις έξυπνες κάρτες. Είναι ανεξάρτητο από το λειτουργικό σύστημα της κάρτας αφού έχει υλοποιηθεί με τη γλώσσα προγραμματισμού Java. Βασικός του στόχος είναι να αυξήσει το βαθμό διαλειτουργικότητας της έξυπνης κάρτας ώστε αυτή να μπορεί να χρησιμοποιηθεί από δίκτυα υπολογιστών, web browsers και από οποιεσδήποτε άλλες πλατφόρμες οι οποίες τρέχουν java και χρειάζεται να αλληλεπιδρούν με έξυπνες κάρτες. Στο κεφάλαιο αυτό αναπτύσσονται οι λόγοι εκείνοι που οδήγησαν στην ανάπτυξη ενός τέτοιου περιβάλλοντος εργασίας, τα πλεονεκτήματα που παρέχει, η αρχιτεκτονική του και τα βασικά συστατικά της ενώ η τελευταία παράγραφος περιγράφει πως μπορούμε να προγραμματίσουμε κάποιες βασικές λειτουργίες με το open card framework.

4.1 Οντότητες που εμπλέκονται στην ανάπτυξη μιας εφαρμογής έξυπνων καρτών

Για την ανάπτυξη εφαρμογών έξυπνων καρτών είναι απαραίτητη η συνεργασία των παρακάτω οντοτήτων καθένας από τους οποίους διαδραματίζει διαφορετικό ρόλο:

- ***Εκδότης της κάρτας (Card Issuer)***: Ο οποίος είναι υπεύθυνος για την έκδοση των καρτών προς τους πελάτες και είναι αυτός που αποφασίζει ποιες και που θα αποθηκευτούν οι μόνιμες εφαρμογές της κάρτας.

- ***Application Service Provider***: Είναι η οντότητα που προγραμματίζει-αναπτύσσει τις διάφορες εφαρμογές που θα συμπεριλαμβάνονται στην έξυπνη κάρτα.

- ***Παροχείς Λειτουργικών Συστημάτων Έξυπνων Καρτών (Card Operating System Providers)***: Υπάρχει ένας μεγάλος αριθμός από ανταγωνιστικές εταιρίες οι οποίες παρέχουν διαφορετικά λειτουργικά συστήματα έξυπνων καρτών και κατά συνέπεια απαιτείται η εγγραφή διαφορετικού κώδικα για τον προγραμματισμό των εντολών με τις οποίες θα επικοινωνούν οι κάρτες με τις συσκευές ανάγνωσης (εντολές command – response).

- ***Παροχείς συσκευών Ανάγνωσης(Card Terminal Providers)***: Παρέχουν τους αναγνώστες έξυπνων καρτών (ή συσκευές υποδοχής ή τερματικά) τα οποία μπορεί να είναι είτε μια πολύ απλή μονάδα υποδοχής καρτών είτε και πιο

σύνθετη όπως οι συσκευές εισόδου βιομετρικών χαρακτηριστικών. Γεγονός ωστόσο είναι ότι οι διάφοροι αυτοί προμηθευτές δεν έχουν συμφωνήσει σε ένα κοινό πρότυπο επικοινωνίας, με αποτέλεσμα την ύπαρξη πολλών διαφορετικών πρωτοκόλλων επικοινωνίας μεταξύ κάρτας και τερματικού.

Μεταξύ αυτών των οντοτήτων αρχικά και πριν την εμφάνιση ενός ανοιχτού πλαισίου εργασίας υπήρχε μεγάλος βαθμός εξάρτησης. Δηλαδή ο εκδότης και ο application provider έπρεπε να ανήκουν στον ίδιο οργανισμό, να προέρχονται από τον ίδιο προμηθευτή. Ωστόσο στην περίπτωση που είτε ένας application provider θέλει να αναπτύξει τις εφαρμογές του για διαφορετικές κάρτες διαφόρων εκδοτών, είτε ένας εκδότης θέλει να συμπεριλάβει στην κάρτα εφαρμογές διαφόρων application providers είναι σαφές ότι δημιουργείται πρόβλημα.
Επιπλέον δεν δίνεται η δυνατότητα σε έναν εκδότη να χρησιμοποιήσει την ίδια εφαρμογή σε διαφορετικές κάρτες με διαφορετικά λειτουργικά συστήματα, ενώ η πρόσβαση σε αυτές τις εφαρμογές είναι περιορισμένη αφού απαιτούνται συσκευές ανάγνωσης καρτών από έναν συγκεκριμένο προμηθευτή.
Λύση στα παραπάνω προβλήματα δίνει η δημιουργία του Open Card Framework που αποτελεί ένα ανοιχτό πλαίσιο εργασίας, παρέχοντας μια ανοιχτή αρχιτεκτονική και ένα σύνολο από application program interfaces (APIs) με στόχο να μειώσει το βαθμό εξάρτησης των διαφόρων οντοτήτων που εμπλέκονται στην ανάπτυξη εφαρμογών έξυπνων καρτών και να αυξήσει το βαθμό διαλειτουργικότητας τους έτσι ώστε να μπορούν να αναπτύσσουν τα προϊόντα τους ανεξάρτητα και αυτά φυσικά να μπορούν να λειτουργούν για οποιαδήποτε κάρτα και οποιοδήποτε τερματικό.

4.2 Πλεονεκτήματα

Το Open Card Framework συμβάλει στη δημιουργία ενός περιβάλλοντος συνεργασίας, όπου υπολογιστές, έξυπνες κάρτες, κατασκευαστές έξυπνων καρτών και συσκευών ανάγνωσης καθώς και προγραμματιστές εφαρμογών μπορούν να αλληλεπιδρούν εποικοδομητικά εξυπηρετώντας τα δικά τους συμφέροντα. Παρέχει πλεονεκτήματα σε όλες τις οντότητες που εμπλέκονται σε μια εφαρμογή έξυπνων καρτών.

- ***Smart Card Solutions Providers:*** Μπορούν να δημιουργούν τα προϊόντα τους χωρίς να ανησυχούν για τις πλατφόρμες, τις συσκευές ανάγνωσης (τερματικά) ή ακόμα και για τα ιδιαίτερα χαρακτηριστικά της κάθε κάρτας. Μπορούν να επικεντρώνουν την προσοχή τους στην ανάπτυξη εφαρμογών – λύσεων γνωρίζοντας με βεβαιότητα ότι θα μπορούν να εφαρμοστούν σε πλατφόρμες που υποστηρίζουν τη Java και να αλληλεπιδρούν με υλικό εξοπλισμό οποιουδήποτε κατασκευαστή.

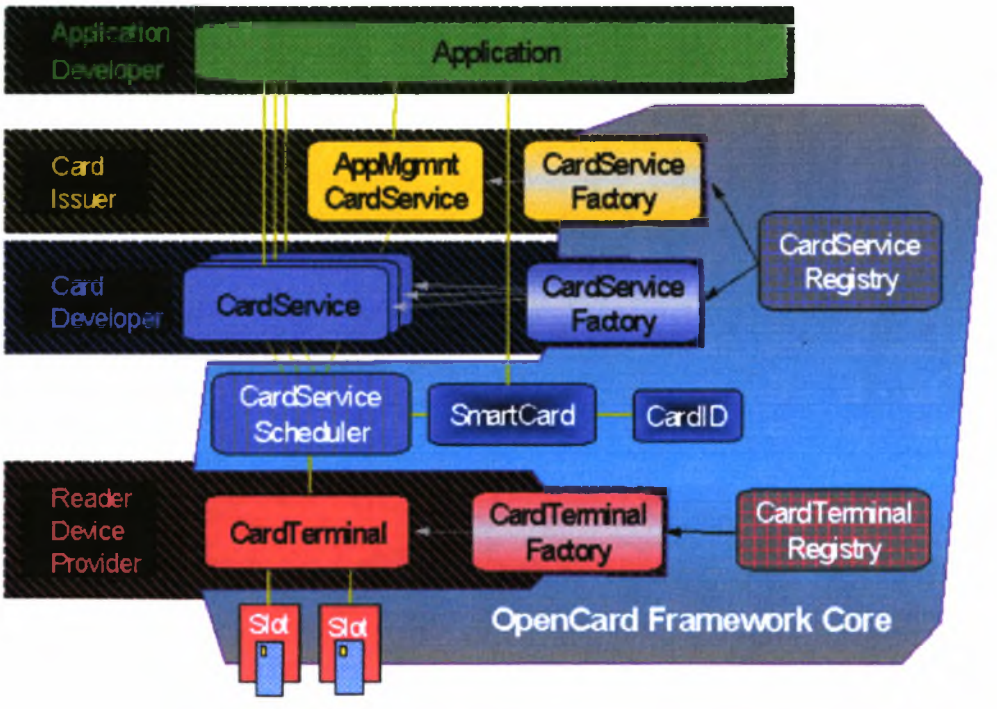
- ***Κατασκευαστές Καρτών και Τερματικών:*** Θα πρέπει πλέον να ανταγωνίζονται κυρίως σε θέματα λειτουργικότητας αφού η εξάρτηση από έναν μόνο πωλητή μειώνεται.

- ***Εκδότες Καρτών:*** έχουν τη δυνατότητα να επιλέξουν από ένα ευρύ σύνολο εφαρμογών, τις εφαρμογές που θα παρέχονται από την κάρτα, ενώ μπορούν να συνεργάζονται με άλλους εκδότες έτσι ώστε συγκεκριμένες εφαρμογές να γίνονται διαθέσιμες σε διαφορετικές κάρτες.

- ***Κάτοχοι / Χρήστες έξυπνων καρτών:*** Οι χρήστες σίγουρα είναι οι "μεγάλοι νικητές" αφού το Open Card Framework επεκτείνει το εύρος των πιθανών χρήσεων μιας έξυπνης κάρτας, ενώ παράλληλα βελτιώνει την ασφάλεια και την εμπιστευτικότητα που οφείλει να παρέχει μια έξυπνη κάρτα.

4.3 Αρχιτεκτονική Open Card Framework

Όπως φαίνεται στην παρακάτω εικόνα το πλαίσιο εργασίας Open Card μπορεί να διαχωριστεί σε δύο βασικά μέρη: στο επίπεδο CardTerminal και στο επίπεδο CardService **[20].**



4.1 Συστατικά Αρχιτεκτονικής Open Card Framework

Το επίπεδο ***CardTerminal*** περιλαμβάνει όλες τις κλάσεις και διεπαφές οι οποίες επιτρέπουν στον application provider να αποκτήσει πρόσβαση στο τερματικό και στις σχισμές του τερματικού(slots). Με τη χρήση αυτών των κλάσεων μπορούμε για παράδειγμα να διαπιστώσουμε εάν έχει γίνει εισαγωγή της κάρτας στην συσκευή ανάγνωσης.
Το επίπεδο ***CardService*** ορίζει την ομώνυμη κλάση: το open card framework αναπαριστά τις διάφορες λειτουργίες της έξυπνης κάρτας ως υπηρεσίες κάρτας (card services). Έτσι κάθε τέτοια υπηρεσία ορίζει ένα συγκεκριμένο API υψηλού

επιπέδου το οποίο επιτρέπει την πρόσβαση σε μια συγκεκριμένη λειτουργία της κάρτας. Για παράδειγμα η υπηρεσία file access card service παρέχει πρόσβαση στο σύστημα αρχείων της έξυπνης κάρτας.

Τα συστατικά από τα οποία αποτελείται η αρχιτεκτονική του OpenCardFramework περιγράφονται παρακάτω:

- ***CardTerminal***: Είναι η κλάση που αναπαριστά το φυσικό τερματικό στο οποίο εισάγεται η κάρτα. Ένα τερματικό πρέπει να διαθέτει μια τουλάχιστον σχισμή στην οποία θα εισαχθεί η κάρτα εκτός και αν πρόκειται για κάρτα τύπου contactless οπότε στην περίπτωση αυτή ο τρόπος εισαγωγής διαφέρει. Ανάλογα με τα χαρακτηριστικά του κάθε τερματικού το open card framework παρέχει και τις κατάλληλες μεθόδους και κλάσεις για τον προγραμματισμό του.

- ***Slot***: Αναπαριστά κάθε διαφορετική σχισμή ενός τερματικού και διαθέτουν δύο βασικά πεδία το slotID που είναι το αναγνωριστικό της σχισμής και το terminal που είναι το τερματικό στο οποίο ανήκει η σχισμή με το συγκεκριμένο slotID.

- ***CardTerminalFactory***: Είναι η κλάση με την οποία δημιουργούνται αντικείμενα CardTerminals ενός συγκεκριμένου τύπου. Με τον τρόπο αυτό ο κατασκευαστής τερματικών μπορεί να δημιουργήσει την δική του έκδοση ενός CardTerminalFactory ώστε να δημιουργούνται τερματικά σύμφωνα με τις προδιαγραφές και χαρακτηριστικά που αυτός θέτει.

- ***CardTerminalRegistry***: Το αντικείμενο αυτό χρησιμοποιείται για να καταγράφει τα αντικείμενα του τύπου CardTerminal ενώ παρέχονται και οι κατάλληλες μέθοδοι για να γνωρίζουμε τον ακριβή αριθμό των εγκαταστημένων τερματικών.

- ***CardServiceScheduler***: Διαχειρίζεται τα λογικά κανάλια μιας έξυπνης κάρτας που βρίσκεται στο τερματικό. Για κάθε έξυπνη κάρτα που είναι γνωστή στο σύστημα και χρησιμοποιείται από κάποια εφαρμογή υπάρχει ένα αντικείμενο CardServiceScheduler το οποίο χειρίζεται τη φυσική πρόσβαση στην έξυπνη κάρτα. Αυτό επιτυγχάνεται δεσμεύοντας κανάλια επικοινωνίας με εκείνες τις CardServices που θέλουν να αποκτήσουν πρόσβαση στην κάρτα.

- ***SmartCard***: Αποτελεί το βασικό αντικείμενο μέσω του οποίου μια εφαρμογή μπορεί να αποκτήσει πρόσβαση στο open card framework. Όλες οι υπηρεσίες CardServices προσπελαύνονται μέσω των μεθόδων του αντικείμενου SmartCard το οποίο συνδέεται πάντα με ένα αντικείμενο CardServiceScheduler από το οποίο και ελέγχεται.

- ***CardID***: Αναπαριστά το ATR της έξυπνης κάρτας δηλαδή τους χαρακτήρες που επιστρέφει η κάρτα ως ένδειξη επιτυχούς ενεργοποίησης της και έναρξης της επικοινωνίας της με την συσκευή ανάγνωσης. Ένα ATR είναι διαφορετικό για κάθε κάρτα και αποτελεί το αναγνωριστικό της. Επιπλέον περιλαμβάνει κάποιους χαρακτήρες οι οποίοι καλούνται historical characters οι οποίοι ορίζονται από τον εκδότη της κάρτας και καθορίζουν τον τύπο της όπως επίσης και τις εφαρμογές που αυτή υποστηρίζει.

- ***CardService***: Με το αντικείμενο αυτό παρέχονται στην εφαρμογή οι λειτουργίες τις έξυπνης κάρτας οι οποίες μπορεί να είναι για παράδειγμα το σύστημα αρχείων που χρησιμοποιεί η κάρτα ή η βάση δεδομένων της και εξαρτάται πάντα από το λειτουργικό της σύστημα. Για την επικοινωνία με την έξυπνη κάρτα το αντικείμενο CardService πρέπει να ζητήσει και να δεσμεύσει από τον CardServiceScheduler ένα κανάλι επικοινωνίας δηλαδή ένα αντικείμενο που καλείται CardChannel.

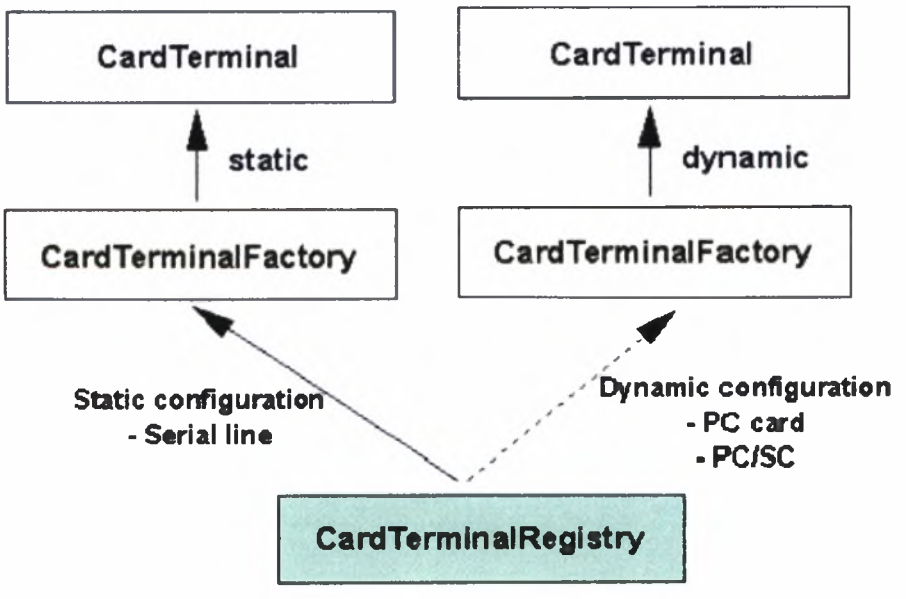
- ***CardServiceFactory***: Είναι το αντικείμενο που αρχικοποιεί τις υπηρεσίες CardServices μιας συγκεκριμένης έξυπνης κάρτας, ενώ αυτό αρχικοποιείται από το CardServiceRegistry.

- ***CardServiceRegistry***: Χρησιμοποιείται για να διατηρεί όλα τα αντικείμενα CardServiceFactory. Όταν γίνει αίτηση για μια CardService το αντικείμενο CardServiceRegistry ελέγχει όλα τα καταχωρημένα αντικείμενα CardServiceFactory ώστε να βρεθεί το κατάλληλο, αυτό που μπορεί δηλαδή να δημιουργήσει την συγκεκριμένη υπηρεσία. Στο σύστημα υπάρχει μόνο ένα στιγμιότυπο του συγκεκριμένου αντικειμένου και μπορεί να προσπελαστεί με τη μέθοδο CardServiceRegistry.getRegistry ().

4.3.1 Το επίπεδο CardTerminal

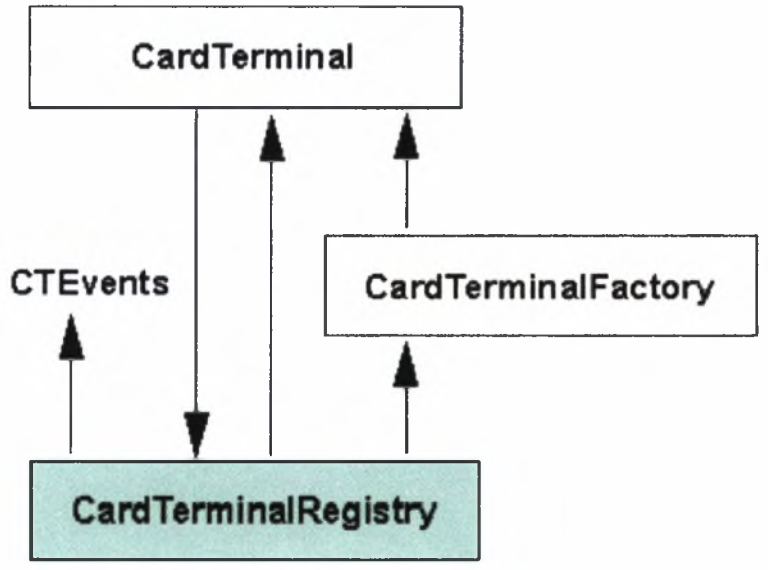
Έχουμε ήδη αναφέρει ότι στην αγορά κυκλοφορούν διάφοροι τύποι συσκευών ανάγνωσης έξυπνων καρτών κάθε ένας από τους οποίους μπορεί να παρέχει και διαφορετικές λειτουργίες. Οι απλές συσκευές ανάγνωσης υποστηρίζουν τη βασική λειτουργία εισόδου – εξόδου της έξυπνης κάρτας μέσω μιας μοναδικής σχισμής (slot) στην οποία εισάγεται η κάρτα. Περισσότερο πολύπλοκοι αναγνώστες καρτών μπορούν να διαθέτουν πολλαπλές σχισμές ή να περιλαμβάνουν την εισαγωγή κωδικού για την αυθεντικοποίηση του κατόχου της κάρτας μέσω πληκτρολογίου. Οι αναγνώστες μπορούν να συνδεθούν σε διαφορετικές θύρες εισόδου / εξόδου (σειριακές θύρες ή PC Card διαύλους) και συνοδεύονται από το κατάλληλο λογισμικό- οδηγό για ένα συγκεκριμένο λειτουργικό σύστημα. Ουσιαστικά το open card framework κρύβει τις λεπτομέρειες των διαφόρων τύπων συσκευών ανάγνωσης δημιουργώντας ένα πιο αφαιρετικό μοντέλο τερματικού.
Οι κλάσεις του επιπέδου CardTerminal εξυπηρετούν έναν διπλό σκοπό: υλοποιούν τη λειτουργία της πρόσβασης σε φυσικά τερματικά καρτών καθώς και στις έξυπνες κάρτες που έχουν εισαχθεί σε αυτά. Αυτή η λειτουργία ενθυλακώνετε στην CardTerminal class με τα αντίστοιχα αντικείμενα slots και στην CardID class.
Οι φυσικές συσκευές ανάγνωσης έξυπνων καρτών αναπαρίστανται στο open card framework ως στιγμιότυπα της κλάσης CardTerminal, ενώ η διαδικασία answer-to-reset αναπαρίσταται μέσω της κλάσης CardID. Η διαμόρφωση των τερματικών καρτών μπορεί να είναι είτε στατική είτε δυναμική. Στην περίπτωση που είναι στατική, ονομάζεται "εκ των προτέρων (priori)" και η διαμόρφωση του τερματικού πρέπει να λάβει χώρα κατά την εκκίνηση του συστήματος. Στην περίπτωση της δυναμικής διαμόρφωσης, επιπρόσθετοι αναγνώστες μπορούν να προστεθούν κατά τη διάρκεια της εκτέλεσης της εφαρμογής .

Η κλάση CardTerminal είναι μια αφηρημένη υπερκλάση από την οποία μπορούν να προκύψουν συγκεκριμένοι τύποι τερματικών με διαφορετικά χαρακτηριστικά ο καθένας. Κάθε αντικείμενο αυτής της κλάσης έχει το χαρακτηριστικό "slots(σχισμές/ εγκοπές)" ανάλογα με τις φυσικές σχισμές της κάρτας του συγκεκριμένου τερματικού. Η πρόσβαση στην κάρτα η οποία εισάγεται σε μια σχισμή του τερματικού γίνεται μέσω ενός exclusive gate αντικειμένου το οποίο καλείται SlotChannel. Η κλάση CardTerminal πρέπει να εξασφαλίζει ότι για κάθε χρονική στιγμή ο μέγιστος αριθμός των αντικειμένων SlotChannel ανά slot, πρέπει να είναι ίσος με τη μονάδα, έτσι ώστε από τη στιγμή που το αντικείμενο SlotChannel δεσμευτεί κανένα άλλο αντικείμενο να μην μπορεί να αποκτήσει πρόσβαση στην κάρτα μέχρι το αντικείμενο SlotChannel να αποδεσμευτεί.
Οι μέθοδοι που παρέχονται από την κλάση CardTerminal χρησιμοποιούνται για να ελέγχουν αν μια κάρτα βρίσκεται στο τερματικό ανάγνωσης και αυτό γίνεται με την δέσμευση ενός αντικειμένου SlotChannel και με την αποστολή / λήψη APDUs. Για τερματικά που προσφέρουν επιπρόσθετες λειτουργίες όπως διάλογους, PIN pad, ή ανάγνωση δακτυλικών αποτυπωμάτων κ.λ.π. το open card framework παρέχει επιπρόσθετες λειτουργίες οι οποίες υλοποιούνται από την κλάση CardTerminal.
Στο επίπεδο που εξετάζουμε παρέχονται επίσης οι απαραίτητοι μηχανισμοί για την πρόσθεση ή αφαίρεση τερματικών ανάγνωσης καρτών. Οι δύο αυτές λειτουργίες υλοποιούνται από την κλάση CardTerminalFactory και το αντικείμενο CardTerminalRegistry. Κάθε κατασκευαστής τερματικών ανάγνωσης καρτών που υποστηρίζουν το open card framework παρέχει την κλάση CardTerminalFactory η οποία «γνωρίζει» για μια οικογένεια τερματικών ανάγνωσης και για τις σχετικές κλάσεις CardTerminal. Το μοναδικό για το σύστημα αντικείμενο CardTerminalRegistry μπορεί να γνωρίζει πόσες συσκευές ανάγνωσης έχουν εγκατασταθεί και δημιουργεί ένα αντίστοιχο μονοπάτι με αυτά τα τερματικά. Αυτό υλοποιείται στο ακόλουθο σχήμα:



4.2 Το αντικείμενο CardTerminal Registry

Επιπλέον το παραπάνω αντικείμενο διαθέτει μεθόδους για την εγγραφή(καταγραφή) και διαγραφή των αντικειμένων τύπου CardTerminal όπως επίσης και για την αρίθμηση όλων των συσκευών ανάγνωσης.
Η κλάση CardTerminal **[23]** δημιουργεί γεγονότα όταν μια κάρτα εισάγεται ή εξάγεται από ένα τερματικό, με τα οποία ενημερώνει τα υπόλοιπα συστατικά του framework. Αυτά τα γεγονότα περνούν από το αντικείμενο CardTerminalRegistry στον EventGenerator και η διαδικασία αυτή φαίνεται στο παρακάτω διάγραμμα:



4.3 Δημιουργία Γεγονότων και Πέρασμα από τον CardTerminalRegistry στον EventGenerator

4.3.2 Το επίπεδο CardService

Η χρησιμότητα του συγκεκριμένου επιπέδου είναι να παρέχει στο OpenCard Framework έναν τρόπο διαχείρισης των ποικίλων λειτουργικών συστημάτων έξυπνων καρτών καθώς και των συναρτήσεων και λειτουργιών που αυτά παρέχουν.
Οι υπηρεσίες καρτών (Card Services) είναι ο όρος που χρησιμοποιεί το open card framework για να καθιστά διαθέσιμες τις λειτουργίες των έξυπνων καρτών στους προγραμματιστές εφαρμογών και γίνεται με τη χρήση της κλάσης CardService.
Αυτή τη στιγμή το open card framework ορίζει μόνο κάποιες από τις πιο σημαντικές λειτουργίες των έξυπνων καρτών όπως είναι η υπηρεσία FileAccessCardService για την πρόσβαση στα αρχεία της και SignatureCardService για τη δημιουργία ψηφιακών υπογραφών. Ωστόσο στα μελλοντικά σχέδια του OpenCard Consortium είναι η ανάπτυξη υπηρεσιών για όλες τις λειτουργίες που μπορεί να προσφέρει μια έξυπνη κάρτα. Στις πιο σημαντικές υπηρεσίες καρτών περιλαμβάνονται οι : FileAccessCardService, SignatureCardService, AppletAccessCardService και AppletManagerCardService οι οποίες περιγράφονται παρακάτω.

FileAccessCardService

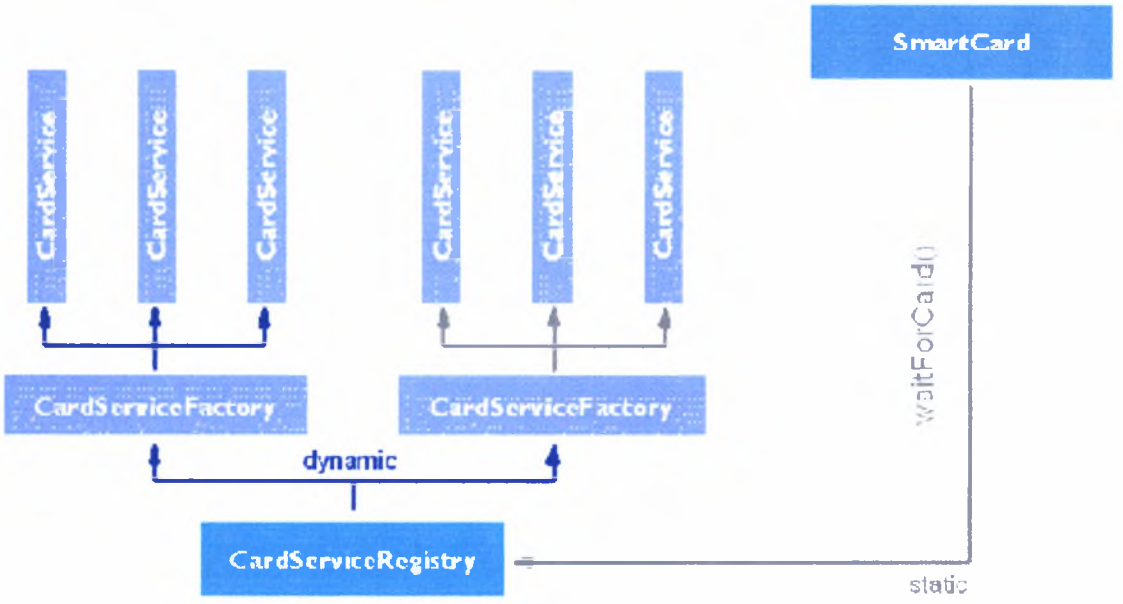
Παρέχει ένα ολοκληρωμένο σύνολο από διεπαφές και αφηρημένες κλάσεις οι οποίες υλοποιούν τις διάφορες συναρτήσεις με τις οποίες γίνεται η διαχείριση του συστήματος αρχείων μιας έξυπνης κάρτας όπως αυτές ορίζονται από το πρότυπο ISO. Όπως συμβαίνει και με τα υπόλοιπα τμήματα του open card πλαισίου εργασίας, οι κλάσεις αυτές και οι διεπαφές έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να ταιριάζουν απόλυτα με το μοντέλο της Java και με τον τρόπο που αυτή διαχειρίζεται τα αρχεία και εκτελεί τις λειτουργίες εισόδου – εξόδου.

SignatureCardService

Παρέχει μεθόδους για τη δημιουργία και πιστοποίηση ψηφιακών υπογραφών οι οποίες βασίζονται σε κρυπτογραφικούς αλγορίθμους δημοσίου κλειδιού όπως είναι ο RSA και ο DSA.
Γνωρίζουμε ότι οι ηλεκτρονικές υπογραφές βασίζονται σε κρυπτογραφικούς μηχανισμούς, οι οποίοι χρησιμοποιούν κλειδιά κρυπτογράφησης. Ένα σύνηθες τέτοιο σχήμα ηλεκτρονικής υπογραφής δεδομένων χρησιμοποιεί τον αλγόριθμο δημοσίου κλειδιού RSA ο οποίος ουσιαστικά περιλαμβάνει ένα ζεύγος κλειδιών : το δημόσιο και το ιδιωτικό κλειδί. Ο βασικός ρόλος της έξυπνης κάρας είναι να εξασφαλίζει την ασφαλή αποθήκευση αυτών των κλειδιών (κυρίως του ιδιωτικού κλειδιού) και από την άλλη να εκτελεί τη διαδικασία των ηλεκτρονικών υπογραφών.
Με την διεπαφή SignatureCardService μπορεί να γίνει η δημιουργία και η πιστοποίηση ψηφιακών υπογραφών με τη χρήση αλγορίθμων δημοσίου κλειδιού καθώς επίσης και η εισαγωγή, η πιστοποίηση και η εξαγωγή των κλειδιών με την χρήση των διεπαφών KeyImportCardService και KeyGenerationCardService.

AppletAccessCardService και AppletManagerCardService

Μια έξυπνη κάρτα γνωρίζουμε ότι μπορεί να παρέχει πολλές εφαρμογές και αυτό γίνεται μέσω ενός συνόλου applets τα οποία τρέχουν στην κάρτα. Για την διαχείριση αυτών των applets απαιτούνται κάποιες ενέργειες που περιλαμβάνουν την εγκατάσταση νέων applets ή την προσωρινή αφαίρεση applets μόνιμα εγκατεστημένων στην κάρτα, για μια συγκεκριμένη χρονική περίοδο, ή την επαναφορά τους ώστε αυτά να ξαναγίνουν διαθέσιμα. Τέλος η κάρτα θα πρέπει να έχει τη δυνατότητα να γνωρίζει τα διαθέσιμα applets της κάρτας και να παρουσιάζει στον κάτοχο της, το κατάλληλο, σε σχέση πάντα με την υπηρεσία (ενέργεια) που αυτός έχει επιλέξει. Αρμοδιότητα των AppletAccessCardService και AppletManagerCardService είναι όλες οι παραπάνω λειτουργίες. Η πρώτη παρουσιάζει μια λίστα όλων των εφαρμογών ενώ η δεύτερη ορίζει ένα υψηλού επιπέδου API μέσω του οποίου οι εφαρμογές μπορούν να εγκαθιστούν ή να αφαιρούν τα applets τη κάρτας.

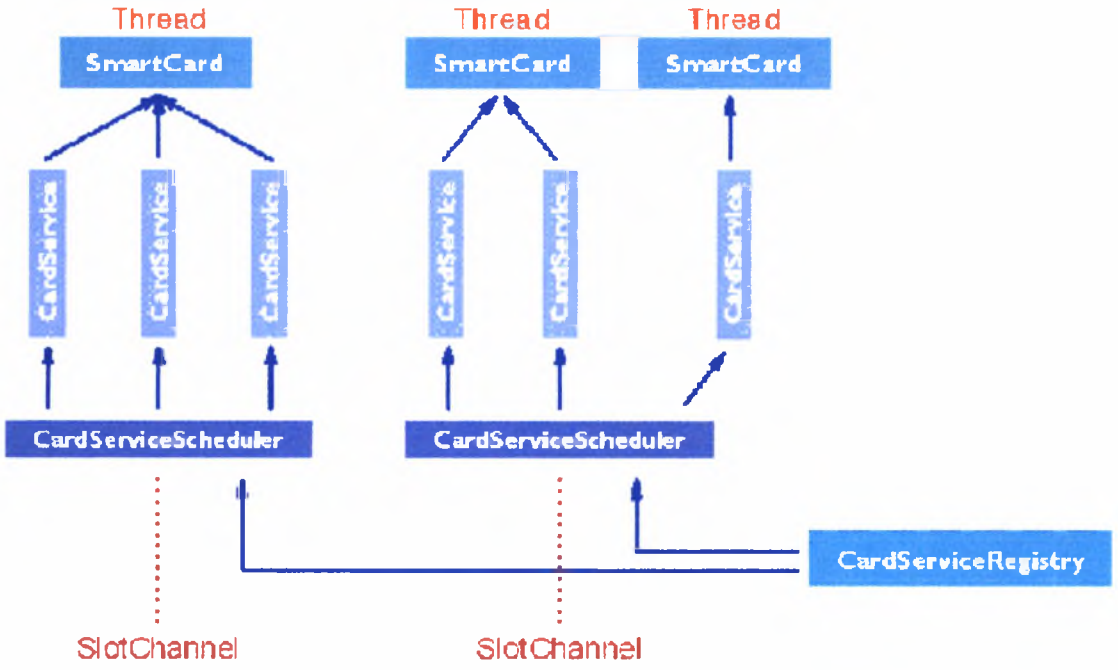


4.4 Κλάσεις του συστατικού CardService

Η κλάση CardService περιέχει όλη την πληροφορία σχετικά με το λειτουργικό σύστημα που υποστηρίζει η κάρτα. Μια CardService μπορεί να χρησιμοποιεί κάποιες άλλες CardServices όπως για παράδειγμα αν θέλει να αποκτήσει πρόσβαση στο σύστημα αρχείων, αυτό γίνεται μέσω της διεπαφής FileSystemService η οποία παρέχει μεθόδους για τη δημιουργία, διαγραφή, ακύρωση (καθιστά δηλαδή ένα αρχείο μη προσβάσιμο) και αποκατάσταση (αποτελεί την αντίστροφη ενέργεια της ακύρωσης) ενός αρχείου.

Η κλάση CardServiceFactory είναι αυτή που γνωρίζει και μπορεί να δημιουργεί αντικείμενα που ανήκουν στην κλάση CardService. Όταν γίνει αίτηση για μια συγκεκριμένη CardService από το open Card Framework, καλείται η CardServiceRegistry η οποία διατηρεί όλες τις καταχωρήσεις των CardServiceFactory, και ελέγχει ποια είναι η κατάλληλη CardServiceFactory που μπορεί να δημιουργήσει την αιτούμενη CardService. Έτσι εάν κάποιο από τα αντικείμενα CardServiceFactory που είναι εγκατεστημένα μπορεί να παράγει την επιθυμητή CardService, το αντικείμενο CardServiceRegistry αφού αποκτήσει την CardService την στέλνει πίσω στην εφαρμογή.

Ενώ στο σύστημα μπορούν να υπάρχουν πολλά αντικείμενα τύπου CardService και CardServiceFactory, μπορεί να υπάρχει μόνο ένα αντικείμενο τύπου CardServiceRegistry το οποίο προσπελαύνεται με τη μέθοδο CardServiceRegistry.getRegistry ().



4.5 Ταυτόχρονη πρόσβαση σε μια έξυπνη κάρτα

Η επικοινωνία μεταξύ αντικείμένων τύπου CardSrevice και CardTerminal γίνεται μέσω του CardServiceScheduler. Το αντικείμενο αυτό παρέχει ταυτόχρονη πρόσβαση σε μια έξυπνη κάρτα και αντιπροσωπεύεται από ένα αντικείμενο SlotChannel. Τα αντικείμενα της κλάσης SlotChannel λειτουργούν ως μια πύλη ώστε να παρέχουν πρόσβαση στην έξυπνη κάρτα και να αλληλεπιδρούν με αυτή ενώ επιπλέον χρησιμοποιούνται για την αποστολή και λήψη APDUs.
Για να επιτρέπεται σε περισσότερα από ένα αντικείμενα του τύπου CardService να αποκτήσουν πρόσβαση στην κάρτα που βρίσκεται στο τερματικό, κρίνεται απαραίτητη η παρουσία του CardServiceScheduler. Έτσι όταν ένα αντικείμενο CardService θέλει να αποκτήσει πρόσβαση στην κάρτα πρέπει αρχικά να δεσμεύσει ένα αντικείμενο CardChannel, (το οποίο είναι ένα κανάλι επικοινωνίας και χρησιμοποιείται τόσο για την ανταλλαγή APDUs όσο και για την πρόσβαση σε άλλους πόρους που σχετίζονται με την κάρτα όπως για παράδειγμα το τερματικό στο οποίο αυτή έχει εισαχθεί) και αφού ολοκληρώσει την εργασία του να το αποδεσμεύσει.

4.3.3 Το συστατικό CardManagement

Το συστατικό αυτό από τη σκοπιά του Open Card Framework είναι απλά μια ακόμη CardService και δεν διαφέρει σε τίποτα από τις άλλες CardServices. Ωστόσο ο διαχωρισμός αυτός υφίσταται γιατί η υπηρεσία που παρέχει θεωρείται ιδιαίτερα κρίσιμη για την επιτυχία του Open Card Framework.
Ένας προγραμματιστής εφαρμογών προτού ξεκινήσει να αναπτύσσει την εφαρμογή θα πρέπει να λάβει υπόψη του διάφορες προδιαγραφές και περιορισμούς που υποβάλλει ο εκδότης της κάρτας όπως για παράδειγμα την φυσική τοποθεσία των εφαρμογών στην κάρτα και των δεδομένων που αυτή θα χρησιμοποιεί καθώς και άλλες πληροφορίες όπως για παράδειγμα μια λίστα όλων των εφαρμογών που θα περιλαμβάνει η κάρτα σε όλη τη διάρκεια της ζωής της.

Το συστατικό λοιπόν CardManagement παρέχει τις υπηρεσίες εκείνες, η εφαρμογή των οποίων επηρεάζεται από τους παραπάνω περιορισμούς που θέτει ο εκδότης της κάρτας, μέσω μιας υψηλού επιπέδου διεπαφής και εκτελεί τις ακόλουθες λειτουργίες:

- Τοποθετεί και επιλέγει τις μόνιμα στην κάρτα εφαρμογές
- Παρέχει μια λίστα με όλες τις εφαρμογές που υποστηρίζει η κάρτα
- Εγκαθιστά και απεγκαθιστά εφαρμογές
- Μπλοκάρει και ξεμπλοκάρει εφαρμογές

Οι βασικές κλάσεις του συστατικού αυτού είναι οι ApplicationContext και ApplicationID. Μια εφαρμογή παρέχει ένα αντικείμενο ApplicationID στην CardManagementCardService η οποία στη συνέχεια ελέγχει αν η συγκεκριμένη εφαρμογή υπάρχει στην κάρτα. Στην περίπτωση επιτυχούς ελέγχου η CardManagementCardService επιστρέφει ένα αντικείμενο ApplicationContext στην επικαλούμενη εφαρμογή. Στη συνέχεια αυτό το αντικείμενο χρησιμοποιείται από την εφαρμογή για να ζητήσει κάποια συγκεκριμένη υπηρεσία CardService. Για παράδειγμα στην περίπτωση που μια έξυπνη κάρτα οργανώνει τις εφαρμογές τις με βάση το σύστημα αρχείων σύμφωνα με το πρότυπο ISO, μπορεί να ζητήσει από την FileSystemCardService να μεταβεί στο dedicated file που αποτελεί τη ρίζα του συστήματος αρχείων.

4.4 Προγραμματίζοντας με το Open Card Framework

Αρχικοποίηση του open card framework

Το πρώτο βήμα που θα πρέπει να γίνει προτού ξεκινήσει οποιαδήποτε εφαρμογή είναι να κληθεί η συνάρτηση start() της κλάσης SmartCard η οποία αρχικοποιεί το open card framework και εγκαθίστανται έτσι οι βασικές του ιδιότητες **[18].** Αφού η εφαρμογή τελειώσει πρέπει να κληθεί η μέθοδος shutdown() η οποία τερματίζει ολόκληρο το πλαίσιο εργασίας.

Η μέθοδος waitForCard()

Το αντικείμενο SmartCard είναι το βασικό αντικείμενο με το οποίο γίνεται η αλληλεπίδραση με τη φυσική έξυπνη κάρτα. Αφού λοιπόν έχει προηγηθεί η αρχικοποίηση του πλαισίου εργασίας το επόμενο βήμα είναι να αποκτήσουμε ένα αντικείμενο SmartCard και αυτό επιτυγχάνεται με την κλήση της μεθόδου waitForCard() της κλάσης SmartCard η οποία επιστρέφει ένα αντικείμενο SmartCard.
Η μέθοδος αυτή παίρνει ως όρισμα ένα αντικείμενο CardRequest με το οποίο μπορούμε να ορίσουμε επιπλέον λεπτομέρειες που αφορούν στον τύπο της κάρτας και στον χρόνο αναμονής μέχρι η κάρτα να εισαχθεί στο τερματικό. Πιο συγκεκριμένα μπορούμε να καθορίσουμε τα ακόλουθα:

- Κάθε κάρτα είναι δεκτή συμπεριλαμβανομένης και της κάρτας που έχει μόλις εισαχθεί στο τερματικό

- Μόνο η πιο πρόσφατη κάρτα που έχει εισαχθεί είναι αποδεκτή
- Την συνθήκη που πρέπει να ικανοποιεί η απάντηση ATR της έξυπνης κάρτας και η οποία καθορίζεται από ένα στιγμιότυπο του τύπου CardIDFilter
- Μία δοθείσα κλάση ή διεπαφή CardService που πρέπει να υποστηρίζει η έξυπνη κάρτα
- Ένα συγκεκριμένο τερματικό στο οποίο πρέπει να εισαχθεί η κάρτα
- Το χρονικό διάστημα μέσα στο οποίο η κλήση της μεθόδου πρέπει να επιστρέψει το αποτέλεσμα

Όλες οι παραπάνω λεπτομέρειες μπορούν να καθοριστούν μέσω παραμέτρων που περνούν στους κατασκευαστές ή μεθόδους της κλάσης CardRequest.

Απόκτηση ενός αντικειμένου CardService

Έχουμε ήδη πει ότι οι CardServices είναι οι έννοιες μέσω των οποίων οι λειτουργίες των έξυπνων καρτών γίνονται διαθέσιμες σε μια εφαρμογή. Έτσι το επόμενο βήμα είναι να αποκτήσουμε το κατάλληλο στιγμιότυπο ενός τέτοιου αντικειμένου.
Η διαδικασία που θα ακολουθήσουμε εξαρτάται από τη φύση της εφαρμογής. Έτσι διακρίνουμε δύο περιπτώσεις:

1. Το πρώτο είδος εφαρμογής μπορεί να αλληλεπιδρά με μια συγκεκριμένη μόνιμα στην κάρτα εφαρμογή και
2. Το δεύτερο είδος μπορεί να αλληλεπιδρά με ένα σύνολο (τουλάχιστον δύο) μόνιμα στην κάρτα εγκατεστημένων εφαρμογών.

Στην πρώτη περίπτωση ο καθορισμός της κλάσης CardService γίνεται με την κλήση της μεθόδου getCardService() η οποία παίρνει ως όρισμα την υπηρεσία στην οποία θέλουμε να αποκτήσουμε πρόσβαση, μια τέτοια υπηρεσία θα μπορούσε για παράδειγμα να είναι η FileAccessCardService.
Στην περίπτωση που η εφαρμογή αλληλεπιδρά με διάφορες μόνιμα στην κάρτα εφαρμογές θα πρέπει να δούμε ποιες είναι αυτές οι εφαρμογές. Αυτό γίνεται με το αντικείμενο AppletAccessCardService το οποίο παρέχει μια λίστα με τις διαθέσιμες εφαρμογές της κάρτας, ενώ στη συνέχεια μπορούμε να επιλέξουμε μια συγκεκριμένη εφαρμογή.

Διαχείριση Αρχείων

FileAccessCardService
Τα αρχεία μις έξυπνης κάρτας προσπελαύνονται μέσω της κλάσης FileAccessCardService. Αυτή παρέχει μια σειρά από μεθόδους για την εκτέλεση ενεργειών όπως η ανάγνωση αρχείων και η ενημέρωση τους. Όλες οι μέθοδοι χρειάζονται μια παράμετρο η οποία καλείται CardFilePath το οποίο καθορίζει το αρχείο στόχο, δηλαδή το αρχείο στο οποίο θέλουμε να αποκτήσουμε πρόσβαση. Όλη η πληροφορία η οποία σχετίζεται με ένα αρχείο της κάρτας παρέχεται από ένα στιγμιότυπο της διεπαφής CardFileInfo. Η διεπαφή αυτή παρέχει μεθόδους και χρησιμοποιούνται για να αποκτήσουμε πληροφορίες σχετικά με το είδος του αρχείου, δηλαδή αν πρόκειται για ένα directory αρχείο ή transparent κ.λ.π..

Επιπλέον μπορούμε να μάθουμε ακόμη πιο λεπτομερείς πληροφορίες όπως για παράδειγμα το μέγεθος του αρχείου.
Αρχικά θα πρέπει να καθορίσουμε το μονοπάτι που βρίσκεται το αρχείο master file (αποτελεί όπως ήδη έχουμε πει στο κεφάλαιο 2 τη ρίζα του δένδρου στο οποίο είναι οργανωμένα τα αρχεία της κάρτας) ενώ στη συνέχεια το μονοπάτι του αρχείου στο οποίο θέλουμε να αποκτήσουμε πρόσβαση. Ακολούθως μπορούμε να καλέσουμε τη μέθοδο getFileInfo η οποία επιστρέφει τις πληροφορίες των αρχείων.

Οι κλάσεις CardFileInput/OutputStream

Το πακέτο java.io ορίζει κλάσεις και διεπαφές για την πρόσβαση σε αρχεία που είναι αποθηκευμένα σε κλασικά συστήματα αρχείων όπως για παράδειγμα στο σκληρό δίσκο. Οι έξυπνες κάρτες διαθέτουν διάφορους τύπους αρχείων όπως είναι τα transparent files τα οποία μπορούν να συγκριθούν με τα αρχεία ενός κλασικού συστήματος αρχείων. Έτσι οι κλάσεις CardFileInputStream και CardFileOutputStream επεκτείνουν τις κλάσεις InputStream και OutputStream αντίστοιχα του πακέτου java.io και χρησιμοποιούν τις μεθόδους και του μηχανισμούς που αυτό παρέχει για να προσπελαστούν τα transparent files της κάρτας.
Ομοίως υπάρχουν οι κλάσεις CardFileReader και CardFileWriter οι οποίες επεκτείνουν τις κλάσεις του πακέτου java.io, InputStreamReader και OutputStreamWriter αντίστοιχα και χρησιμοποιούνται για τον χειρισμό streams χαρακτήρων, δηλαδή τα streams εισόδου για την ανάκτηση πληροφοριών και τα streams εξόδου για την αποθήκευση πληροφοριών.

Η κλάση CardRecordAccess

Οι κλάσεις CardRecordAccess και CardRecord παρέχουν τους μηχανισμούς για την πρόσβαση σε αρχεία τύπου record είτε είναι linear fixed είτε είναι linear variable. Η κλάση CardRecordAccess ορίζει ένα δείκτη ο οποίος δείχνει σε μια εγγραφή (record) και παρέχει μεθόδους για την ανάγνωση και την εγγραφή, είτε μιας record είτε ενός πίνακα από records.
Εδώ θα πρέπει να σημειώσουμε ότι η παραπάνω κλάση χρησιμοποιείται μόνο για αρχεία που είναι τύπου linear και όχι για transparent αρχεία. Τα τελευταία είδαμε ότι προσπελαύνονται μέσω των κλάσεων CardFileInputStream, CardFileOutputStream και CardFileReader, CardFileWriter.
Στην περίπτωση που τα αρχεία είναι τύπου cyclic αυτά δεν μπορούν να χρησιμοποιήσουν κάποια από τις παραπάνω μεθόδους προσπέλασης και προς το παρόν δεν υπάρχουν κάποιοι μηχανισμοί για την προσπέλαση τους.

Δημιουργία και πιστοποίηση υπογραφών

Η δημιουργία και η πιστοποίηση υπογραφών παρέχονται μέσω των κλάσεων SignatureCardService, KeyImportCardService, και KeyGenerationCardService ανάλογα με την λειτουργικότητα της κάθε κάρτας.

Δημιουργία υπογραφής

Η διαδικασία υπογραφής ενός μηνύματος απαιτεί αρχικά τον υπολογισμό μίας τιμής που καλείται hash value του μηνύματος ενώ αυτή η τιμή στη συνέχεια

κρυπτογραφείται με τη χρήση του ιδιωτικού κλειδιού ενός αλγορίθμου δημοσίου κλειδιού όπως είναι για παράδειγμα ο RSA.
Η λειτουργία αυτή γίνεται με τη μέθοδο signData που παρέχει η κλάση SignatureCardService. Η μέθοδος θα πρέπει να γνωρίζει το μήνυμα που πρόκειται να υπογραφεί όπως επίσης και ποίοι αλγόριθμοι κατακερματισμού και δημοσίου κλειδιού θα χρησιμοποιηθούν. Η αναφορά στο ιδιωτικό κλειδί περιλαμβάνει ένα αντικείμενο CardFilePath που δείχνει τη διεύθυνση του αρχείου που βρίσκεται το κλειδί και έναν ακέραιο που συσχετίζει το κλειδί με το path στο οποίο βρίσκεται.

Πιστοποίηση υπογραφής
Η πιστοποίηση της υπογραφής του μηνύματος γίνεται αποκρυπτογραφώντας την δοθείσα υπογραφή με χρήση του δημοσίου κλειδιού και υπολογίζοντας την τιμή hash value του "καθαρού (μη κρυπτογραφημένου)" μηνύματος και συγκρίνοντας αυτή την τιμή με την αποκρυπτογραφημένη υπογραφή. Η μέθοδος που χρησιμοποιείται είναι η VerifySignedData η οποία ομοίως με την παραπάνω διαδικασία της υπογραφής πρέπει να γνωρίζει το καθαρό μήνυμα και το υπογεγραμμένο μήνυμα όπως επίσης τους αλγορίθμους κατακερματισμού και δημοσίου κλειδιού που θα χρησιμοποιηθούν και τέλος το κλειδί.

Εισαγωγή κλειδιών
Η κλάση KeyImportCardService η οποία επεκτείνει την κλάση SignatureCardService παρέχει τις απαραίτητες μεθόδους για την εισαγωγή των κλειδιών που χρησιμοποιούνται από τους ασύμμετρους αλγορίθμους κατά τη διαδικασία των υπογραφών. Οι μέθοδοι που υποστηρίζονται είναι οι importPrivateKey και importPublicKey. Οι παράμετροι που παίρνουν είναι το ίδιο το κλειδί καθώς και την πληροφορία που καθορίζει την τοποθεσία όπου θα αποθηκευτεί το συγκεκριμένο κλειδί. Οι δύο παραπάνω μέθοδοι δεν ελέγχουν την εγκυρότητα των κλειδιών που εισάγονται, ωστόσο υπάρχουν δύο άλλες μέθοδοι που εκτελούν παράλληλα αυτή την επιπρόσθετη διαδικασία και είναι οι: importAndValidatePrivateKey importAndValidatePublicKey. Αυτές οι μέθοδοι λοιπόν προκαλούν την έξυπνη κάρτα να ελέγξει την εγκυρότητα των κλειδιών που εισάγονται και αυτό γίνεται με την πιστοποίηση της υπογραφής αυτού του κλειδιού που περνά ως επιπρόσθετη παράμερος στην μέθοδο μαζί με το δεύτερο κλειδί που χρησιμοποιείται για την πιστοποίηση(δηλαδή τα πρόσθετα ορίσματα που δέχεται η μέθοδος είναι η υπογραφή του κλειδιού του οποίου πρόκειται να γίνει η εισαγωγή καθώς και το κλειδί που απαιτείται για την εξακρίβωση της υπογραφής). Έτσι η κάρτα δέχεται το κλειδί που θέλουμε να εισάγουμε μόνο αν η πιστοποίηση είναι επιτυχής.

Δημιουργία κλειδιών
Η κλάση KeyGenerationCardService η οποία επίσης επεκτείνει την κλάση SignatureCardService παρέχει τις μεθόδους για τη δημιουργία του ζεύγους των κλειδιών που χρησιμοποιείται από τους ασύμμετρους αλγορίθμους όπως και μεθόδους για την ανάγνωση του δημοσίου κλειδιού που αντιστοιχεί σε ένα ιδιωτικό κλειδί. Η μέθοδος που δημιουργεί το ζεύγος των κλειδιών και ονομάζεται generateKeyPair και δέχεται ως ορίσματα τις διευθύνσεις στην κάρτα όπου θα αποθηκευτούν το δημόσιο και ιδιωτικό κλειδί καθώς και τον αριθμό των bytes του κλειδιού που δείχνει την ισχύ του και τον αλγόριθμο που θα χρησιμοποιηθεί. Στην

περίπτωση που η κάρτα δεν υποστηρίζει το μέγεθος του κλειδιού ή τον συγκεκριμένο αλγόριθμό προκαλούνται από την κάρτα οι κατάλληλες εξαιρέσεις.
Η μέθοδος που χρησιμοποιείται για την ανάγνωση του δημοσίου κλειδιού ονομάζεται readPublicKey και οι παράμετροί της είναι ο αλγόριθμος δημοσίου κλειδιού και μια αναφορά στο κλειδί που πρόκειται να διαβάσουμε (λέγοντας αναφορά στο κλειδί εννοούμε το μονοπάτι CardFilePath που βρίσκεται το κλειδί καθώς και τον ακέραιο αριθμό του κλειδιού, που συσχετίζει το κλειδί με το μονοπάτι στο οποίο αυτό βρίσκεται). Αντίστοιχη μέθοδος για την ανάγνωση του ιδιωτικού κλειδιού όπως είναι αναμενόμενο δεν υπάρχει αφού το ιδιωτικό κλειδί δεν πρέπει ποτέ να γνωστοποιείται στον εξωτερικό κόσμο και μόνο η κάρτα μπορεί να γνωρίζει για αυτό.

Κεφάλαιο 5

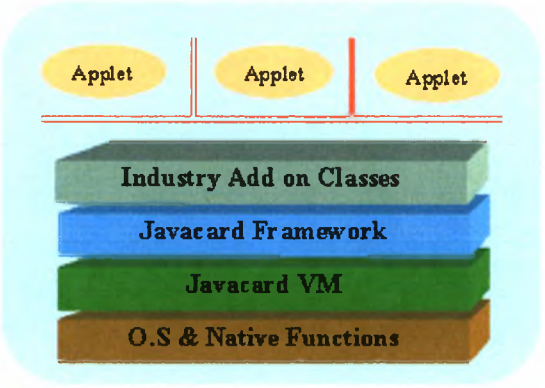
Java Card Technology

Το 5° κεφάλαιο ασχολείται με την τεχνολογία Java Card, η οποία παρέχει τα εργαλεία και τις προδιαγραφές για την ανάπτυξη Java Card εφαρμογών. Παρουσιάζεται η αρχιτεκτονική του συστήματος μιας Java Card, η οποία αποτελείται από τα ακόλουθα συστατικά : τα ***applets***, που είναι οι εφαρμογές που τρέχουν στην κάρτα και οι τρόποι επικοινωνίας αυτών με την κάρτα, η ***Java Card Virtual machine*** η οποία παίρνει τα μεταγλωττισμένα προγράμματα και παράγει έναν κώδικα byte (bytecode) καθώς και τα συστατικά από τα οποία αυτή αποτελείται, το ***Java Card Runtime Environment***, η διάρκεια ζωής και τα χαρακτηριστικά του, το ***Java Card Framework*** το οποίο παρέχει ένα σύνολο από κλάσεις και υπηρεσίες για την ανάπτυξη εφαρμογών και τέλος παρατίθενται κάποιοι από τους λόγους που καθιστούν την Java Card τεχνολογία ως το πιο ιδανικό και ασφαλές μέσο ανάπτυξης εφαρμογών πάνω σε έξυπνες κάρτες.

5.1 Εισαγωγή

Java Card

Η αρχιτεκτονική του συστήματος μιας Java Card φαίνεται στο σχήμα 5.1. Όπως διακρίνουμε, η *Virtual Machine* της Java Card είναι χτισμένη πάνω σε ένα εξειδικευμένο ολοκληρωμένο κύκλωμα, που χρησιμοποιεί ένα εγγενές λειτουργικό σύστημα. Το JVM επίπεδο κρύβει ουσιαστικά την τεχνολογία του συγκεκριμένου κατασκευαστή, χρησιμοποιώντας κοινή γλώσσα και interface. Το *Javacard Framework* ορίζει ένα σύνολο από κλάσεις για την ανάπτυξη εφαρμογών (API) και για την παροχή υπηρεσιών του συστήματος σε αυτές τις εφαρμογές. Μία συγκεκριμένη εταιρεία μπορεί να αναπτύξει βιβλιοθήκες που μπορούν να προστεθούν στο σύστημα για την προσφορά μιας νέας υπηρεσίας ή για τη βελτίωση του μοντέλου ασφάλειας ή του μοντέλου συστήματος. Τέλος, οι εφαρμογές που τρέχουν σε ένα Java Card ονομάζονται applets. Μία κάρτα μπορεί να έχει περισσότερα από ένα applets, καθένα από τα οποία έχει ένα μοναδικό κωδικό (AID).



5.1 Αρχιτεκτονική της Java Card

Σε αντίθεση με τη Java Virtual Machine σε έναν προσωπικό υπολογιστή, η Java Card VM εκτελείται συνεχώς. Η περισσότερη πληροφορία που κρατείται στην κάρτα πρέπει να διατηρηθεί ακόμα και όταν η τροφοδοσία παύσει, δηλαδή όταν η κάρτα αφαιρεθεί από τη συσκευή CAD. Η Java Card VM δημιουργεί αντικείμενα στην EEPROM για να διατηρήσει πληροφορία που δεν πρέπει να χαθεί (persistent information). Η διάρκεια ζωής της Java Card VM είναι ίδια με τη διάρκεια ζωής της κάρτας. Όταν δεν παρέχεται τροφοδοσία, η VM τρέχει σε έναν άπειρο κύκλο ρολογιού.
Η ζωή ενός applet ξεκινά όταν αυτό εγκαθίσταται στην κάρτα και δηλώνεται στο registry πίνακα του συστήματος, και τερματίζει όταν αφαιρείται από αυτόν τον πίνακα. Ο χώρος που καταλάμβανε ένα applet που αφαιρέθηκε από τον πίνακα μπορεί να επαναχρησιμοποιηθεί εάν η υπηρεσία garbage collection έχει υλοποιηθεί στην κάρτα. Ένα applet σε μία κάρτα βρίσκεται σε μη ενεργοποιημένη κατάσταση, έως ότου επιλεχθεί από το τερματικό.

Υποσύνολο γλώσσας Java Card

Τα Java Card προγράμματα είναι προφανώς γραμμένα σε Java. Γίνονται compiled χρησιμοποιώντας κοινούς Java compilers. Εξαιτίας, όμως, των περιορισμένων πόρων μνήμης και υπολογιστικής ισχύος της κάρτας, δεν έχουν περιληφθεί όλες οι ικανότητες της Java στο Java Card πρότυπο. Πιο συγκεκριμένα το Java Card δεν υποστηρίζει:

- Δυναμικό φόρτωμα κλάσεων.
- Διαχειριστή ασφάλειας.
- Threads και συγχρονισμό.
- Μεγάλους τύπους δεδομένων.
- Finalization
- Object Cloning

Java Card Framework

Το Java Card 2.0 Framework περιέχει τέσσερα packages:

1. Javacard.framework Πρόκειται για το βασικό package στην κάρτα. Ορίζει κλάσεις όπως Applet και PIN, που αποτελούν τα βασικά δομικά κομμάτια για Java Card προγράμματα. Επίσης ορίζει τις κλάσεις APDU, System και Util, που παρέχουν υπηρεσίες συστήματος στα προγράμματα, όπως χειρισμό APDU, διαμοιρασμό αντικειμένων, και άλλα.

2. Javacardx.framework Παρέχει μία αντικειμενοστραφή σχεδίαση για ένα σύστημα αρχείων συμβατό με το ISO 7816-4 πρότυπο.
3. Javacardx.crypto
4. Javacardx.cryptoENC Τα δύο τελευταία packages υποστηρίζουν την απαραίτητη σε smart cards κρυπτογραφική δυνατότητα.

Java Card –Ασφάλεια

Τα Java Applets υπόκεινται στους κανόνες ασφάλειας της Java, αν και το μοντέλο ασφάλειας σε Java Card συστήματα διαφέρει από το γενικό μοντέλο σε πολλά σημεία. Η κλάση *Security Manager* δεν υποστηρίζεται στη Java Card.
Τα Java applets δημιουργούν αντικείμενα που αποθηκεύουν και χειρίζονται πληροφορία. Ένα αντικείμενο είναι στην κατοχή του applet που το δημιουργεί. Ακόμα και αν κάποιο applet έχει το δείκτη αναφοράς στο αντικείμενο, δεν μπορεί να καλέσει τις μεθόδους του αντικειμένου, παρά μόνο αν είναι ο κάτοχός του ή αν στο αντικείμενο έχουν αποδοθεί ιδιότητες για διαμοίραση (sharing) . Η διαμοίραση των αντικειμένων επιτρέπεται μεταξύ applets.

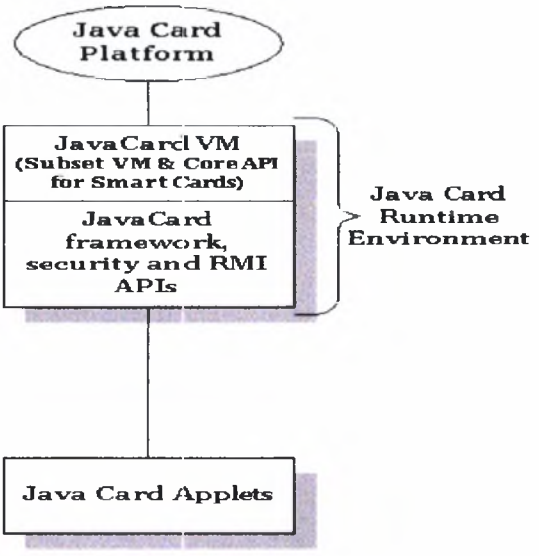
5.2 Java Card και προδιαγραφές Java Card τεχνολογίας

Η Sun Microsystems διαπίστωσε τη μεγάλη σημασία των έξυπνων καρτών και τα πλεονεκτήματα που αυτές παρέχουν και όρισε ένα σύνολο προδιαγραφών, για ένα υποσύνολο της Java Technology, ώστε να δημιουργήσει εφαρμογές που τρέχουν στις έξυπνες κάρτες και καλούνται Java Card applets. Μια συσκευή που υποστηρίζει αυτές τις προδιαγραφές καλείται Java Card Platform και σε αυτή μπορούν να συνυπάρχουν με ασφαλή τρόπο πολλαπλές εφαρμογές διαφόρων παροχέων εφαρμογών. Η Java Card Platform έχει αναπτυχθεί ειδικά για περιβάλλοντα έξυπνων καρτών και απεικονίζεται στην εικόνα 5.2.
Μια Java Card διαθέτει τα ακόλουθα χαρακτηριστικά:

- 8-bit ή 16-bit κεντρική μονάδα επεξεργασίας (CPU) η οποία τρέχει στα 3.7 MHz
- 1 Kbytes μνήμη RAM
- 16 Kbytes και άνω μνήμη EEPROM ή Flash memory

Το στοιχείο εκείνο που τις διαφοροποιεί από τις κοινές έξυπνες κάρτες είναι η δυνατότητα τους να τρέχουν προγράμματα σε Java.
Η τεχνολογία Java Card αποτελείται από τα ακόλουθα τρία τμήματα:

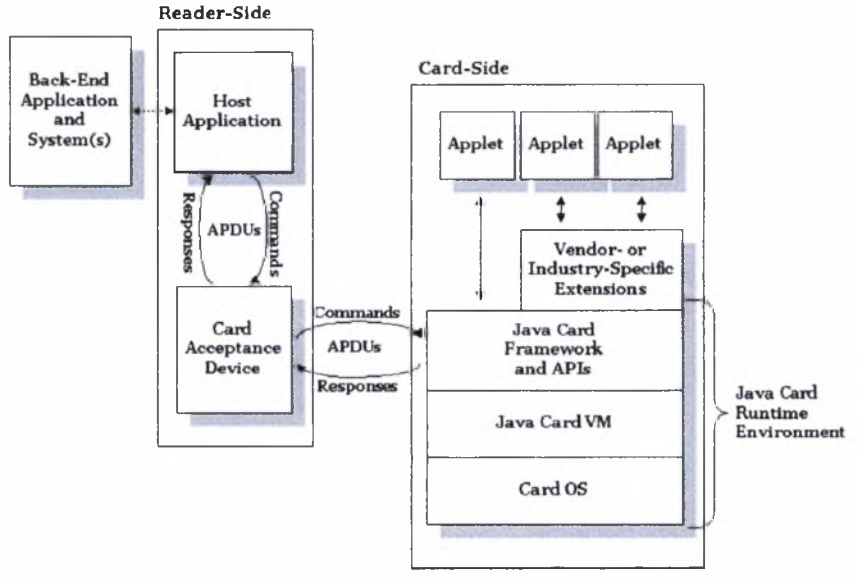
- Java Card Virtual Machine, η οποία ορίζει ένα υποσύνολο της γλώσσας προγραμματισμού Java και τη Virtual Machine για έξυπνες κάρτες
- Java Card Runtime Environment, ορίζει τη συμπεριφορά των java cards κατά τη διάρκεια της εκτέλεσης των εφαρμογών
- Java Card API, η οποία ορίζει το βασικό πλαίσιο εργασίας, τα πακέτα και τις κλάσεις που χρησιμοποιούνται για την υλοποίηση εφαρμογών έξυπνων καρτών.



5.2 Java Card Platform

5.3 Συστατικά μιας εφαρμογής Java Card (java card application)

Μια ολοκληρωμένη εφαρμογή Java Card αποτελείται από ένα back-end σύστημα, μια host (off- card) εφαρμογή, μια συσκευή διεπαφής (δηλαδή έναν card reader), ένα applet που τρέχει στην κάρτα, user credentials και τέλος ένα λογισμικό υποστήριξης (supporting software). Όλα τα παραπάνω συστατικά συνθέτουν μια ασφαλή και ολοκληρωμένη end to end εφαρμογή **[24].** Η παρακάτω εικόνα απεικονίζει την αρχιτεκτονική μιας Java Card εφαρμογής.



5.3 Αρχιτεκτονική Java Card εφαρμογής

Back End Σύστημα (Back End Applications and Systems)
Οι back end εφαρμογές παρέχουν υπηρεσίες που είναι απαραίτητες για την υποστήριξη των java applets που τρέχουν στην κάρτα. Μια τέτοια εφαρμογή παρέχει την σύνδεση με ασφαλή συστήματα τα οποία μαζί με τα credentials που βρίσκονται στη κάρτα δημιουργούν ένα αυξημένο επίπεδο ασφάλειας. Σε ένα ηλεκτρονικό σύστημα πληρωμών για παράδειγμα μια back end εφαρμογή εξασφαλίζει την πρόσβαση στην πιστωτική κάρτα καθώς και σε άλλες σχετικές πληροφορίες.

Host εφαρμογή (Reader-Side Host Application)
Η host εφαρμογή τοποθετείται σε ένα τερματικό όπως έναν προσωπικό υπολογιστή ή ένα τερματικό ηλεκτρονικών πληρωμών και διαχειρίζεται την επικοινωνία μεταξύ χρηστών, του Java Card applet και του παροχέα της back end εφαρμογής.
Τυπικά οι πωλητές έξυπνων καρτών παρέχουν εκτός από τα εργαλεία ανάπτυξης, APIs για να υποστηρίξουν τις εφαρμογές από την πλευρά του αναγνώστη, όπως επίσης και Java Card applets. Τέτοια παραδείγματα περιλαμβάνουν το Open Card Framework, ένα σύνολο από APIs βασισμένο σε Java το οποίο αποκρύπτει κάποιες λεπτομέρειες σχετικές με την αλληλεπίδραση αναγνωστών έξυπνων καρτών, διαφορετικών πωλητών και δύο άλλα μοντέλα τα οποία θα περιγράψουμε στη συνέχεια και είναι το Java Card Remote Method Invocation distributed-object model και το Security and Trust Services API (SATSA).

Η συσκευή υποδοχής κάρτας (Reader-side card acceptance device)
Οι συσκευές αποδοχής έξυπνων καρτών (card acceptance device - CAD) είναι οι συσκευές διεπαφής της Java Card και της host εφαρμογής. Είναι αυτή που τροφοδοτεί με ενέργεια την κάρτα και εξασφαλίζει την επικοινωνία τους είτε αυτή είναι ενσύρματη είτε ασύρματη ανάλογα με τον τύπο της κάρτας. Αυτή η συσκευή μπορεί να είναι ένας αναγνώστης καρτών ο οποίος συνδέεται σε έναν επιτραπέζιο υπολογιστή με την χρήση μιας σειριακής θύρας είτε μπορεί να είναι ενσωματωμένη σε ένα τερματικό όπως για παράδειγμα σε ένα τερματικό ηλεκτρονικών πληρωμών (ATM) και μπορεί να περιλαμβάνει πληκτρολόγιο για την εισαγωγή κάποιου κωδικού και να εμφανίζει επίσης ένα σχετικό διάλογο. Η συσκευή διεπαφής προωθεί τις εντολές APDU(application protocol data unit) από την host εφαρμογή στην κάρτα και αντίστοιχα τις αποκρίσεις της κάρτας προς την host εφαρμογή.

Περιβάλλον και applets της κάρτας (Card side Applets and Environment)
Η πλατφόρμα Java Card είναι ένα πολύ-εφαρμογικό περιβάλλον και όπως δείχνει το σχήμα αρχιτεκτονικής μιας Java Card εφαρμογής(εικόνα 5.3), σε μια κάρτα μπορούν να ανήκουν ένα ή περισσότερα Java Card applets μαζί με το κατάλληλο λογισμικό υποστήριξης – το λειτουργικό σύστημα της κάρτας και το Java Card Runtime Environment (JCRE). Το τελευταίο αποτελείται την Java Card Virtual Machine, το Java Card πλαίσιο εργασίας με τα απαραίτητα APIs και κάποια επιπλέον APIs που αποτελούν επεκτάσεις των προηγούμενων.
Όλα τα Java Card Applets πρέπει να επεκτείνουν την βασική κλάση Applet της γλώσσας Java και να χρησιμοποιούν τις μεθόδους install( ) και process( ). Το JCRE καλεί την πρώτη μέθοδο όταν γίνεται η εγκατάσταση του applet ενώ η δεύτερη μέθοδος καλείται κάθε φορά που υπάρχει μια εισερχόμενη εντολή APDU για το applet.

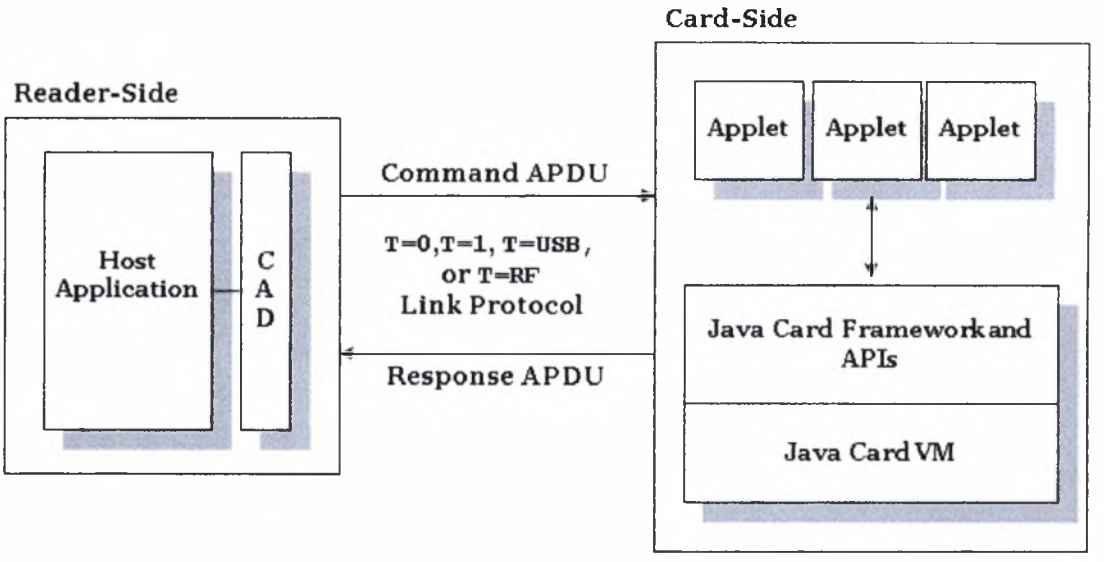
Τα Java Card Applets αρχικοποιούνται αφού πρώτα γίνει η φόρτωσή τους και παραμένουν 'ζωντανά' ακόμα και όταν δεν υπάρχει τροφοδοσία με ρεύμα. Ένα card applet συμπεριφέρεται όμοια με έναν εξυπηρετητή (server) και είναι παθητικό. Αφού γίνει η ενεργοποίηση της κάρτας κάθε applet μένει ανενεργό μέχρι τη στιγμή που θα επιλεγεί, δηλαδή τη στιγμή που θα γίνει και η αρχικοποίηση του, ενώ είναι ενεργό μόνο όταν ανατεθεί σε αυτό μια εντολή APDU.

5.4 Τρόποι επικοινωνίας με ένα Java Card Applet (Πρόσβαση έξυπνης κάρτας)

Μπορούν να χρησιμοποιηθούν δύο μοντέλα επικοινωνίας μεταξύ της εφαρμογής host και του Java Card applet. Το πρώτο μοντέλο είναι το θεμελιώδης μοντέλο με πέρασμα μηνυμάτων (message passing model) και το δεύτερο βασίζεται στο Java Card Remote Method Invocation (JCRMI), που αποτελεί υποσύνολο του J2SE RMI distributed-object model. Επιπροσθέτως το μοντέλο SATSA επιτρέπει την χρήση και των δύο μεθόδων πρόσβασης σε μια έξυπνη κάρτα μέσω ενός πιο αφηρημένου API το οποίο βασίζεται στο Generic Connection Framework (GCF) API.

5.4.1 Πρώτο μοντέλο επικοινωνίας – Message passing model

Το εν λόγω μοντέλο υλοποιείται στο σχήμα 5.4 και αποτελεί τη βάση όλων των Java Card επικοινωνιών. Στο κέντρο του όπως παρατηρούμε βρίσκεται το application protocol data unit, τα πακέτα δεδομένων δηλαδή που ανταλλάσσονται μεταξύ της συσκευής διεπαφής (CAD) και του Java Card Framework. Το Java Card Framework λαμβάνει και προωθεί στο κατάλληλο applet κάθε εισερχόμενη εντολή APDU η οποία αποστέλλεται από την CAD. Το applet επεξεργάζεται την εντολή APDU και επιστρέφει μια απόκριση APDU. Η επικοινωνία μεταξύ αναγνώστη και κάρτας γίνεται με τη χρήση πρωτοκόλλων διασύνδεσης για τα οποία έχουμε μιλήσει στο τρίτο κεφάλαιο και είναι είτε το πρωτόκολλο T=0 το οποίο είναι byte-oriented είτε το T=1 το οποίο είναι block- oriented. Ωστόσο υπάρχουν και άλλα πρωτόκολλα επικοινωνίας τα οποία αναφέρονται ως T=USB, T=RF. Η κλάση JCRE APDU αποκρύπτει από την εφαρμογή κάποιες λεπτομέρειες αυτών των πρωτοκόλλων όχι όμως όλες εξαιτίας της πολυπλοκότητας τους. Η δομή της εντολής APDU και της απόκρισης APDU ακολουθούν τα διεθνή πρότυπα ISO/IEC 7816-3 και 7816-4 τα οποία έχουν μελετηθεί εκτενώς στο τρίτο κεφάλαιο.
Αναφορικά με την επεξεργασία των APDUs από τα applets, κάθε φορά που υπάρχει μια εισερχόμενη εντολή για κάποιο συγκεκριμένο applet καλείται από το JCRE η μέθοδος process() της κλάσης applet. Αυτή παίρνει ως όρισμα την εισερχόμενη APDU, και στη συνέχεια το applet την αναλύει, επεξεργάζεται τα δεδομένα, δημιουργεί την απόκριση APDU και επιστρέφει τον έλεγχο στο JCRE.



5.4 Message Passing - Μοντέλο Επικοινωνίας

5.4.2 Δεύτερο μοντέλο επικοινωνίας – Java Card RMI (JCRMI)

Γενικά ένα μοντέλο RMI αποτελείται: 1. από μια εφαρμογή εξυπηρετητή η οποία δημιουργεί και καθιστά προσβάσιμα απομακρυσμένα αντικείμενα και 2. από μια εφαρμογή πελάτη η οποία αποκτά απομακρυσμένες αναφορές σε απομακρυσμένα αντικείμενα ενώ στη συνεχεία καλεί μεθόδους αυτών των αντικειμένων. Στην περίπτωση του JRMI ένα java card applet είναι ο εξυπηρετητής ενώ η host εφαρμογή είναι ο πελάτης.
Για την ανάπτυξη του μοντέλου JCRMI παρέχεται το πακέτο javacardx.rmi της κλάσης RMIService. Τα μηνύματα JCRMI ενθυλακώνονται στο αντικείμενο APDU και περνούν ως ορίσματα τις μεθόδους της RMIService. Με άλλα λόγια το μοντέλο JCRMI παρέχει ένα κατανεμημένο, αντικειμενοστραφή μηχανισμό όπου πελάτης και εξυπηρετητής επικοινωνούν ανταλλάσσοντας πληροφορίες μεθόδων, ορίσματα και επιστρέφοντας τιμές.

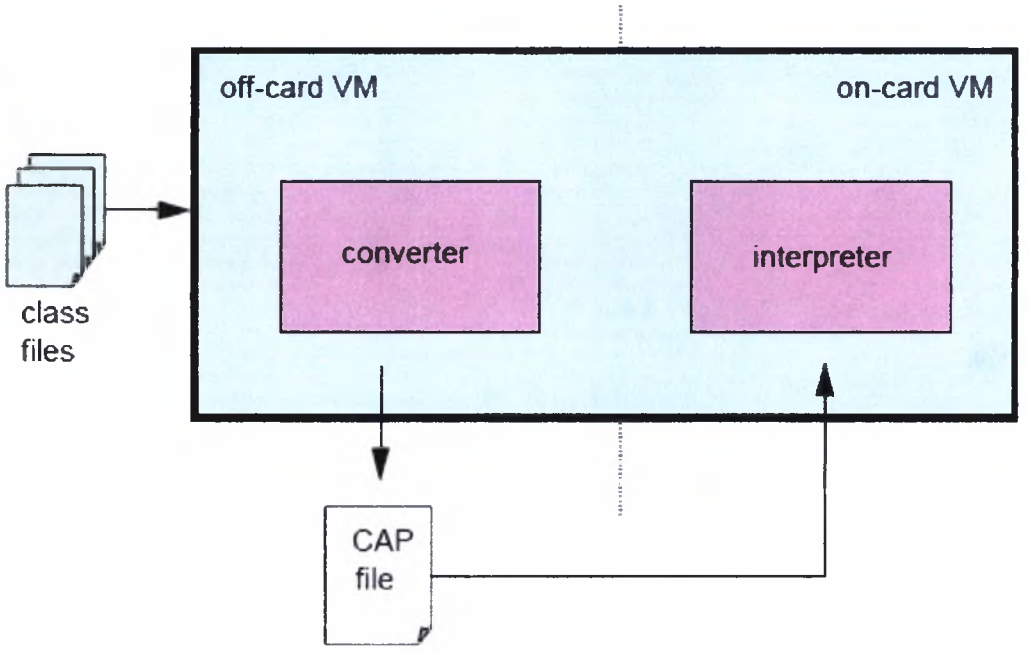
5.4.3 TheSecurity and Trust Services API (SATSA)

Ορίζει ένα προαιρετικό πακέτο και παρέχει ένα ασφαλές και έμπιστο API. Το API αυτό παρέχει πρόσβαση σε υπηρεσίες ασφαλών αντικειμένων, όπως είναι δηλαδή μια έξυπνη κάρτα και περιλαμβάνει ασφαλή αποθήκευση - ανάκτηση ευαίσθητων πληροφοριών, ενώ επιπλέον προσφέρει υπηρεσίες κρυπτογράφησης και αυθεντικοποίησης. Υποστηρίζει τόσο το μοντέλο επικοινωνίας message passing όσο και το JRMI ορίζοντας μια ακόμη πιο αφηρημένη διεπαφή. Το πρώτο επιτυγχάνεται με τα apdu:URL scheme και την APDUConnection ενώ το δεύτερο με τα jcrmi: scheme και το JavaCardRMIConnection.

5.5 Java Card Virtual Machine

Ορίζει ένα υποσύνολο της γλώσσας προγραμματισμού Java και μια εικονική μηχανή συμβατή με τη java ειδικά για έξυπνες κάρτες. Περιλαμβάνει

αναπαραστάσεις δυαδικών δεδομένων, τύπους αρχείων και το σύνολο εντολών JCVM.
Η εικονική μηχανή της Java Card πλατφόρμας υλοποιείται σε δύο τμήματα: το εξωτερικό που τρέχει εκτός κάρτας και το εσωτερικό που τρέχει μέσα στην κάρτα. Η εικόνα 5.5 δείχνει ακριβώς αυτό το διαχωρισμό.



5.5 Java Card Virtual Machine

Το εσωτερικό τμήμα της Java Card Virtual Machine παίρνει τα μεταγλωττισμένα προγράμματα και παράγει το λεγόμενο κώδικα byte(bytecode) δηλαδή εντολές που μπορεί να καταλάβει το λειτουργικό σύστημα και να τις εκτελέσει και διαχειρίζεται τις κλάσεις και τα αντικείμενα. Το εξωτερικό τμήμα είναι ένα εργαλείο ανάπτυξης το οποίο τυπικά καλείται *Java Card Converter tool*, και φορτώνει, πιστοποιεί και γενικά προετοιμάζει τις κλάσεις ενός applet ώστε να μπορεί να γίνει η εκτέλεση στο εσωτερικό της κάρτας. Η έξοδος αυτού του εργαλείου είναι ένα αρχείο που ονομάζεται Converted Applet (CAP) και περιέχει όλες τις κλάσεις ενός πακέτου της java σε μια εκτελέσιμη δυαδική αναπαράσταση. Τέλος πιστοποιεί ότι οι κλάσεις αυτές συμφωνούν με τις προδιαγραφές της Java Card.Το αρχείο CAP φορτώνεται στην java card και η εκτέλεση του γίνεται από τον διερμηνέα που βρίσκεται στο εσωτερικό τμήμα.
Η JCVM υποστηρίζει μόνο ένα περιορισμένο υποσύνολο της γλώσσας προγραμματισμού Java, ωστόσο όμως εξακολουθεί να διατηρεί κάποια βασικά και οικεία χαρακτηριστικά της γλώσσας όπως είναι: τα αντικείμενα, η κληρονομικότητα, τα πακέτα, η δυναμική δημιουργία αντικειμένων, οι διεπαφές και οι εξαιρέσεις. Επειδή η μνήμη μιας έξυπνης κάρτας είναι περιορισμένη, οι προδιαγραφές της JCVM απαγορεύουν τη χρήση κάποιων αντικειμένων της γλώσσας που θα απαιτούσαν αρκετή μνήμη. Κάποια από αυτά είναι τα ακόλουθα:

Χαρακτηριστικά της γλώσσας που δεν υποστηρίζονται	Δυναμική φόρτωση κλάσεων, java.lang.SecurityManager, νήματα, object clonning
Λέξεις κλειδιά που δεν υποστηρίζονται	native, synchronized, transient, volatile, strictfp
Τύποι δεδομένων που δεν υποστηρίζονται	char, double, float, long και πολυδιάστατοι πίνακες
Κλάσεις και διεπαφές	
Εξαιρέσεις	Κάποιες Exceptions και Error υποκλάσεις και λάθη έχουν παραληφθεί αφού δεν πρόκειται να προκληθούν στην περίπτωση της Java Card πλατφόρμας

Επιπλέον η Java Card VM επιβάλει περιορισμούς σε κάποια χαρακτηριστικά του προγράμματος. Πιο συγκεκριμένα έχουμε τους ακόλουθους περιορισμούς για τα πακέτα και τις κλάσεις:

- **Πακέτα**
  - Ένα πακέτο μπορεί να αναφέρεται το πολύ σε 128 άλλα πακέτα
  - Το όνομα του πακέτου πρέπει να έχει έως 255 bytes. Το μέγεθος των χαρακτήρων εξαρτάται πάντα από την κωδικοποίηση που χρησιμοποιείται.
  - Ένα πακέτο μπορεί να έχει έως και 255 κλάσεις
- **Κλάσεις**
  - Μια κλάση μπορεί να υλοποιεί έως και 15 διεπαφές
  - Μια διεπαφή να κληρονομείται το πολύ από 14 διεπαφές
  - Ένα πακέτο μπορεί να έχει έως και 256 μεθόδους τύπου static εάν αυτό περιέχει applets (ένα *applet package*) ή έως 255 εάν δεν περιέχει applets (ένα *library package*)
  - Μια κλάση μπορεί να υλοποιεί έως και 128 μεθόδους τύπου public ή protected

5.5.1 Διάρκεια ζωής JCVM

Η διάρκεια ζωής της JCVM ξεκινά από τη στιγμή που θα κατασκευαστεί η κάρτα, γίνουν οι απαραίτητοι έλεγχοι σωστής λειτουργίας και προτού αυτή εκδοθεί και παραδοθεί στον κάτοχο της. Ξεκινά με την αρχικοποίηση του JCRE το οποίο δημιουργεί τα JCRE framework αντικείμενα, και του οποίου η διάρκεια ζωής είναι όση και του JCVM. Αφού γίνει η αρχικοποίηση, οι αλληλεπιδράσεις με την κάρτα ελέγχονται από ένα applet της κάρτας. Όταν η κάρτα δεν τροφοδοτείται με ρεύμα όλα τα δεδομένα τα οποία είναι αποθηκευμένα στην μνήμη RAM χάνονται ενώ αυτά που είναι αποθηκευμένα σε μη διαγράψιμη μνήμη (όπως και η JCVM) διατηρούνται. Όταν επανατροφοδοτηθεί με ρεύμα η κάρτα η VM γίνεται ενεργή και η κατάσταση αυτής και των αντικειμένων επανέρχονται σε συνεπή κατάσταση.

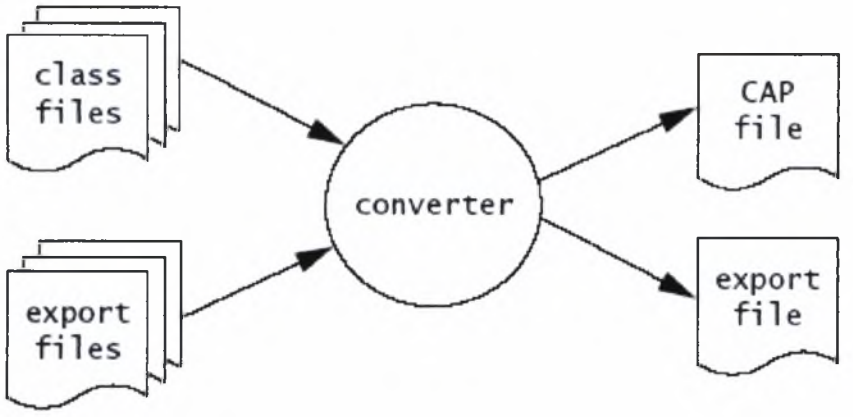
5.5.2 Αρχεία CAP και αρχεία εξόδου

Η Java Card τεχνολογία εισάγει δύο νέους τύπους δυαδικών αρχείων που επιτρέπουν την ανάπτυξη και εκτέλεση Java Card λογισμικού. Ένα αρχείο CAP περιέχει μια εκτελέσιμη δυαδική αναπαράσταση των κλάσεων ενός πακέτου java. Είναι ένα αρχείο JAR το οποίο περιέχει ένα σύνολο από συστατικά για κάθε ένα από τα οποία υπάρχει αποθηκευμένο ένα ξεχωριστό αρχείο μέσα στο αρχείο JAR. Κάθε τέτοιο συστατικό περιγράφει τα περιεχόμενα του CAP αρχείου δηλαδή τις κλάσεις, πληροφορίες πιστοποίησης και άλλα. Το λογισμικό που φορτώνεται στις Java έξυπνες κάρτες έχει τη μορφή των CAP αρχείων.
Τα αρχεία εξόδου δεν φορτώνονται στην έξυπνη κάρτα και για αυτό δεν χρησιμοποιούνται απευθείας από τον διερμηνέα (interpreter). Αντιθέτως παράγονται και χρησιμοποιούνται από τον converter για λόγους πιστοποίησης και διασύνδεσης. Αυτά τα αρχεία μπορούν να παρομοιαστούν με τα header files που χρησιμοποιούνται στην γλώσσα προγραμματισμού C. Δεν περιέχουν κώδικα απλά παρέχουν στον προγραμματιστή χρήσιμες πληροφορίες όπως τα ονόματα των κλάσεων, τα πεδία και τις μεθόδους ώστε να μπορεί να τα χρησιμοποιήσει χωρίς να γνωρίζει λεπτομέρειες για την υλοποίηση τους.

5.5.3 Java Card Converter

Αντίθετα με τη Java Virtual machine η οποία επεξεργάζεται ένα αρχείο class την κάθε χρονική στιγμή, η μονάδα επεξεργασίας ενός converter (μετατροπέα) είναι ένα πακέτο. Από τον πηγαίο κώδικα και μέσω του java compiler παράγονται τα αρχεία .class. Όλα τα αρχεία .class που συνθέτουν ένα πακέτο υφίστανται κάποια διαδικασία προεπεξεργασίας από τον μετατροπέα, ο οποίος εντέλει μετατρέπει το πακέτο σε αρχείο CAP.
Κατά τη διαδικασία της μετατροπής ο μετατροπέας αναλαμβάνει εργασίες που η Java virtual machine θα αναλάμβανε κατά τη διάρκεια του class- loading. Οι εργασίες αυτές είναι:

- Πιστοποιεί ότι τα ιδεατά αρχεία των java αρχείων .class είναι σωστά διαμορφωμένα
- Ελέγχει για την ορθότητα της γλώσσας με βάση τους κανονισμούς που θέτει η Java Card γλώσσα.
- Αρχικοποιεί τις στατικές μεταβλητές.
- Επιλύει αναφορές σε κλάσεις, μεθόδους ή πεδία διαμορφώνοντας αυτές με τέτοιο τρόπο ώστε η διαχείριση και επεξεργασία τους να γίνεται πιο αποδοτικά από την κάρτα.
- Βελτιστοποιεί τον κώδικα
- Δεσμεύει αποθηκευτικό χώρο και δημιουργεί ειδικές δομές δεδομένων για την αναπαράσταση των κλάσεων



5.6 Μετατροπή πακέτου σε CAP αρχείο

Όπως βλέπουμε ο μετατροπέας παίρνει ως είσοδο, τα αρχεία .class τα οποία θα πρέπει να μετατραπούν και ένα ή περισσότερα export files. Παράλληλα με τη παραγωγή του αρχείου CAP ο μετατροπέας δημιουργεί ένα export file που αντιστοιχεί στο πακέτο που μετατράπηκε. Η διαδικασία μετατροπής του πακέτου έχει ως εξής: Ο μετατροπέας φορτώνει τα αρχεία .class σε ένα java πακέτο. Εάν το πακέτο κάνει import κλάσεις άλλων πακέτων θα πρέπει να φορτώσει τα export files αυτών των πακέτων. Οι έξοδοι του μετατροπέα είναι φυσικά το αρχείο CAP και ένα export file για το πακέτο του οποίου έγινε η μετατροπή.

5.5.4 Java Card Interpreter

Οι εργασίες για τις οποίες είναι αρμόδιος είναι οι ακόλουθες:

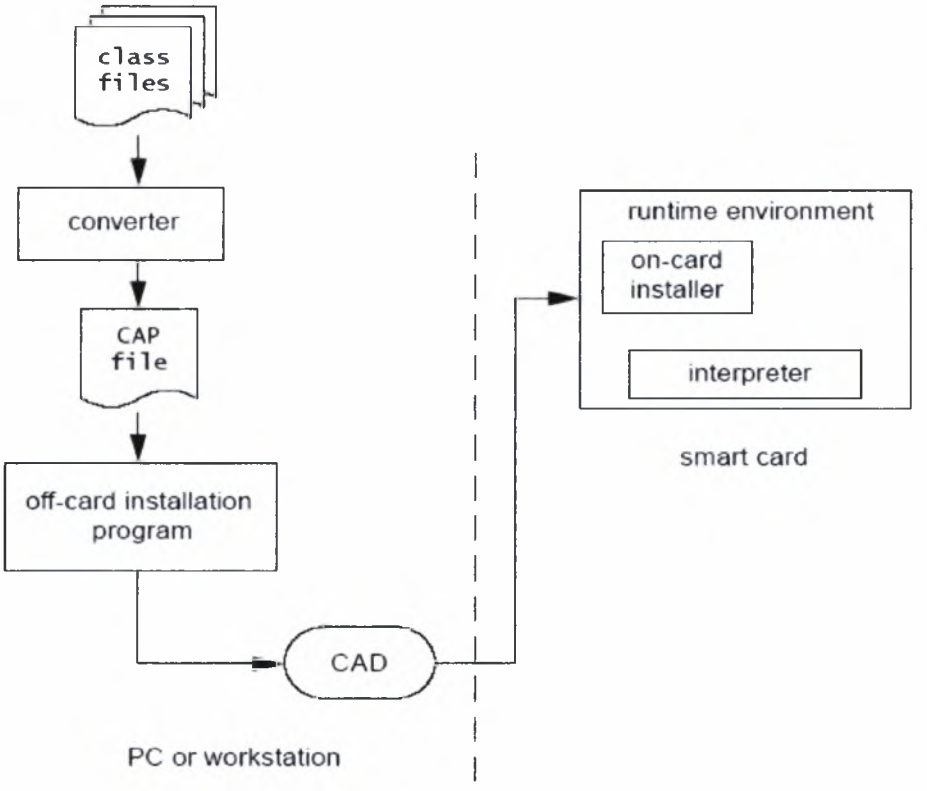
- Εκτέλεση των bytecode εντολών και ουσιαστικά εκτέλεση των applets
- Έλεγχος δέσμευσης μνήμης και δημιουργία αντικειμένων
- Παίζει σημαντικό ρόλο στην εξασφάλιση ασφαλούς συστήματος κατά τη διάρκεια της εκτέλεσης

Μέχρι στιγμής είδαμε ότι η Java Card virtual machine αποτελείται από τον μετατροπέα και τον διερμηνέα. Ωστόσο σε πολλές δημοσιεύσεις έχει επικρατήσει η χρήση του διερμηνέα ως συνώνυμο της java card εικονικής μηχανής. Εκείνο που θα πρέπει να τονιστεί όμως είναι ότι όλες οι λειτουργίες εκτέλεσης των java class αρχείων διενεργούνται από τον μετατροπέα και τον διερμηνέα μαζί.

5.5.5 Java Card Installer και Off-card πρόγραμμα εγκατάστασης

Ο Java Card διερμηνέας δεν εκτελεί μόνος του την διαδικασία φόρτωσης των αρχείων CAP. Αρμοδιότητα του είναι η εκτέλεση του κώδικα που περιέχεται στα αρχεία CAP. Η τεχνολογία Java Card χρησιμοποιεί ειδικούς μηχανισμούς για τις λειτουργίες της φόρτωσης και εγκατάστασης των αρχείων CAP. Οι μηχανισμοί αυτοί ενσωματώνονται σε μια μονάδα η οποία καλείται installer και είναι τοποθετημένη στο εσωτερικό της κάρτας. Αυτό συνεργάζεται με ένα πρόγραμμα εγκατάστασης που βρίσκεται εκτός της κάρτας και για το λόγο αυτό καλείται off-card program. Το πρόγραμμα αυτό μεταφέρει τον εκτελέσιμο κώδικα που

περιέχεται στο CAP αρχείο, μέσω της συσκευής υποδοχής της κάρτας (CAD) στον installer που τρέχει μέσα στην κάρτα. Ο installer γράφει τα δεδομένα στην μνήμη της έξυπνης κάρτας, τα διασυνδέει με τις υπόλοιπες κλάσεις που βρίσκονται ήδη τοποθετημένες στην κάρτα και τέλος δημιουργεί και αρχικοποιεί τις δομές δεδομένων που χρησιμοποιούνται εσωτερικά από το java card runtime environment. Ο installer και το installation πρόγραμμα, το πώς σχετίζονται μεταξύ τους και με το υπόλοιπο σύστημα Java Card απεικονίζονται παρακάτω.

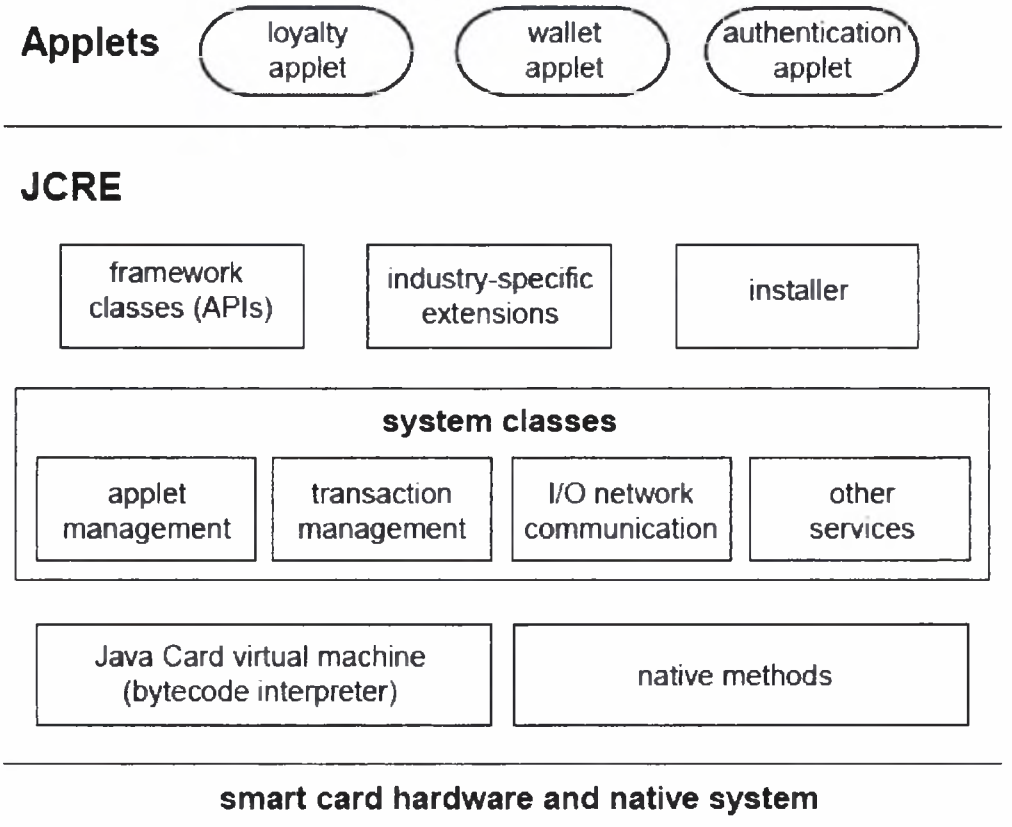


Εικόνα 5.7

5.6 Java Card Runtime Environment

Το Java Card Environment (JCRE) αποτελείται από εκείνα τα συστατικά του συστήματος Java Card τα οποία τρέχουν στο εσωτερικό της έξυπνης κάρτας. Είναι υπεύθυνο για την διαχείριση των πόρων της κάρτας, για τις δικτυακές επικοινωνίες, για την εκτέλεση και ασφάλεια των applets. Για μια έξυπνη κάρτα λοιπόν η σημασία του είναι τόσο ουσιώδης όσο είναι και το λειτουργικό της σύστημα.
Η αρχιτεκτονική του on-card συστήματος υλοποιείται στην παρακάτω εικόνα. Βλέπουμε ότι το JCRE τοποθετείται στην κορυφή του υλικού της έξυπνης κάρτας και τα τμήματα από τα οποία αποτελείται είναι η Java Card Virtual Machine (δηλαδή τον bytecode διερμηνέα), τις κλάσεις του Java Card Framework (δηλαδή τα APIs), συγκεκριμένες βιομηχανικές επεκτάσεις και κλάσεις του JCRE συστήματος. Βλέπουμε ότι τα applets διαχωρίζονται από το υπόλοιπο σύστημα και αυτό συμβαίνει για να είναι ανεξάρτητα από την κάθε κάρτα, να μπορούν να

προγραμματίζονται εύκολα και να είναι συμβατά με διάφορες αρχιτεκτονικές έξυπνων καρτών.
Το τελευταίο κατώτατο επίπεδο του JCRE περιλαμβάνει την Java Card Virtual Machine και τις μητρικές μεθόδους (native) Το πρώτο συστατικό εκτελεί τις εντολές του κώδικα byte (bytecode) , ελέγχει την δέσμευση της μνήμης, διαχειρίζεται αντικείμενα και συμβάλει σε ένα ασφαλές περιβάλλον εκτέλεσης. Οι μητρικές μέθοδοι σχετίζονται με τις ακόλουθες ενέργειες, τη διαχείριση των χαμηλού – επιπέδου πρωτοκόλλων επικοινωνίας, τη διαχείριση της μνήμης, κρυπτογραφικές λειτουργίες και άλλα.



5.8 On - card system architecture

Οι κλάσεις του συστήματος συμπεριφέρονται όπως το περιβάλλον εκτέλεσης JCRE. Οι λειτουργίες τους εμφανίζουν πολλές αναλογίες με τον πυρήνα του λειτουργικού συστήματος. Σκοπός τους είναι η διαχείριση συναλλαγών, η διαχείριση επικοινωνιών μεταξύ host εφαρμογών και Java Card applets και τέλος ο έλεγχος δημιουργίας και επιλογής ενός applet. Για την ολοκλήρωση των εργασιών τους συνήθως οι κλάσεις συστήματος επικαλούνται μητρικές μεθόδους.
Το πλαίσιο εργασίας java card (Java Card application framework) παρέχει τέσσερα βασικά πακέτα απαραίτητα για την εγγραφή java προγραμμάτων και την ανάπτυξη applets έξυπνων καρτών. Το βασικό πλεονέκτημα αυτού του πλαισίου εργασίας είναι ότι καθιστά τη διαδικασία δημιουργίας ενός applet ιδιαίτερα εύκολη χωρίς να απαιτείται από τον προγραμματιστή να γνωρίζει λεπτομέρειες της δομής του συστήματος της έξυπνης κάρτας. Τα applets μπορούν να αποκτήσουν πρόσβαση στο περιβάλλον JCRE μέσω των API κλάσεων που παρέχει το περιβάλλον εργασίας Java Card application framework

Μια συγκεκριμένη βιομηχανία ή επιχείρηση μπορεί να παρέχει επιπρόσθετες βιβλιοθήκες ώστε να προσφέρει επιπλέον υπηρεσίες ή ακόμα ένα πιο βελτιωμένο, σε θέματα ασφάλειας, μοντέλο συστήματος.
Ο installer εξασφαλίζει την ασφαλή φόρτωση λογισμικού και applets μέσα στην κάρτα αμέσως μετά την κατασκευή και την έκδοσή της. Ο installer συνεργάζεται με το off-card πρόγραμμα εγκατάστασης τα οποία μαζί φέρουν εις πέρας την διαδικασία φόρτωσης των δυαδικών περιεχομένων του CAP αρχείου. Το συστατικό installer του JCRE είναι προαιρετικό, αλλά χωρίς αυτό όλο το λογισμικό της κάρτας, συμπεριλαμβανομένων και των applets πρέπει να εγγραφούν στην μνήμη της κάρτας κατά τη διάρκεια κατασκευής της.
Τα java card applets είναι εφαρμογές χρηστών πάνω στην πλατφόρμα java card. Είναι γραμμένα σε γλώσσα java, συγκεκριμένα στο υποσύνολο της γλώσσας java που ορίζεται από την τεχνολογία java card, ενώ ο έλεγχος και η διαχείριση τους γίνεται από το JCRE. Μπορούν να φορτωθούν και να προστεθούν στην έξυπνη κάρτα αμέσως μετά τη βιομηχανοποίηση της.

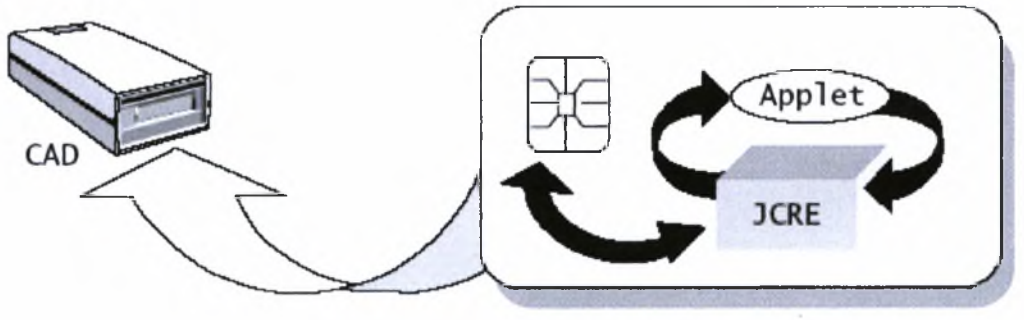
5.6.1 Διάρκεια ζωής του JCRE

Σε ένα προσωπικό υπολογιστή ή σταθμό εργασίας, η java virtual machine τρέχει όπως μια διεργασία του λειτουργικού συστήματος. Δεδομένα και αντικείμενα δημιουργούνται στη μνήμα RAM. Όταν η διεργασία του λειτουργικού συστήματος τερματίσει οι εφαρμογές Java και τα αντικείμενα τους αυτόματα καταστρέφονται.
Σε μια java έξυπνη κάρτα, η java card virtual machine τρέχει εντός του java card runtime environment. Το περιβάλλον JCRE αρχικοποιείται, τη στιγμή που γίνεται η αρχικοποίηση της κάρτας. Η διαδικασία αυτή της αρχικοποίησης του JCRE πραγματοποιείται μια και μοναδική φορά σε όλη τη διάρκεια ζωής της κάρτας. Η διαδικασία αρχικοποίησης περιλαμβάνει την αρχικοποίηση της virtual machine, την δημιουργία αντικειμένων για την παροχή των JCRE υπηρεσιών και τη διαχείριση των applets. Καθώς γίνεται η εγκατάσταση των applets το JCRE δημιουργεί στιγμιότυπα των applets και τα applets δημιουργούν αντικείμενα για την αποθήκευση δεδομένων.
Τα περισσότερα δεδομένα και πληροφορίες που περιέχονται στην κάρτα πρέπει να διατηρούνται ακόμα και όταν δεν παρέχεται ρεύμα στην κάρτα. Για την διαφύλαξη δεδομένων χρησιμοποιείται η μνήμη EEPROM και σε αυτή δημιουργούνται τα δεδομένα και τα αντικείμενα. Η διάρκεια ζωής του JCRE ταυτίζεται με τη διάρκεια ζωής της κάρτας. Όταν δεν παρέχεται ρεύμα η λειτουργία της virtual machine απλά αναστέλλεται ενώ η κατάσταση του JCRE και των αντικειμένων που δημιουργούνται στην κάρτα διαφυλάσσονται.
Την επόμενη φορά που η κάρτα θα ενεργοποιηθεί, το JCRE επανεκινεί την virtual machine φορτώνοντας δεδομένα από τη μνήμη EEPROM. Σε αυτό το σημείο θα πρέπει να τονίσουμε ότι το JCRE δεν ξαναρχίζει τη λειτουργία της virtual machine ακριβώς από το σημείο εκείνο όπου βρισκόταν όταν διακόπηκε η παροχή ρεύματος. Η διαδικασία της επανεκκίνησης γίνεται από την αρχή, αυτό όμως διαφέρει από την αρχικοποίηση. Δηλαδή το JCRE δεν είναι ανάγκη να αρχικοποιηθεί για δεύτερη φορά αφού τα applets και τα αντικείμενα που έχουν δημιουργηθεί διαφυλάσσονται. Στην περίπτωση της επανεκκίνησης εάν μια συναλλαγή δεν έχει προλάβει να

ολοκληρωθεί με επιτυχία το JCRE εκτελεί όλες τις απαραίτητες ενέργειες ώστε να φέρει το JCRE σε συνεπή κατάσταση.

5.6.2 Συμπεριφορά του JCRE κατά τη διάρκεια της CAD session

Η χρονική περίοδος, από τη στιγμή που η κάρτα εισάγεται στη συσκευή υποδοχής έξυπνων καρτών (CAD) και τροφοδοτείται με ρεύμα, μέχρι τη στιγμή που η κάρτα θα αφαιρεθεί από αυτή, καλείται 'CAD session'. Κατά τη διάρκεια αυτής της περιόδου, το JCRE συμπεριφέρεται όπως μια τυπική έξυπνη κάρτα. Δηλαδή επικοινωνεί με την host εφαρμογή μέσω των εντολών APDU. Πιο συγκεκριμένα η ανταλλαγή των πακέτων δεδομένων γίνεται μεταξύ της host εφαρμογής και των applets. Κάθε APDU περιέχει είτε μια εντολή από την host εφαρμογή προς ένα applet είτε ένα μήνυμα απάντησης από το applet στην host εφαρμογή.
Αμέσως μετά την επανεκκίνηση του JCRE, αυτό μπαίνει σε ένα loop περιμένοντας μια εντολή APDU από την host εφαρμογή. Η τελευταία στέλνει εντολές APDU χρησιμοποιώντας τη σειριακή διεπαφή επικοινωνίας μέσω του σημείου εισόδου/ εξόδου της κάρτας.
Μόλις φτάσει μια εντολή, το JCRE είτε επιλέγει ένα applet να τρέξει με βάση τα ορίσματα τής εντολής, είτε προωθεί την εντολή στο applet που είναι επιλεγμένο την τρέχουσα στιγμή και αναθέτει τον έλεγχο της επεξεργασίας της εντολής σε αυτό. Μόλις τελειώσει η επεξεργασία το applet στέλνει μια απάντηση στην host εφαρμογή και επιστρέφει τον έλεγχο στο JCRE. Αυτή η διαδικασία επαναλαμβάνεται μόλις φτάσει η επόμενη προς εκτέλεση εντολή.



5.9 APDU I/O communication

5.6.3 Χαρακτηριστικά του Java Card Runtime Environment

Κάποια από τα βασικά χαρακτηριστικά του JCRE είναι τα ακόλουθα:

- ***Μόνιμα και παροδικά αντικείμενα-*** Γενικά τα αντικείμενα μιας Java Card είναι μόνιμα και δημιουργούνται σε μνήμη μη διαγράψιμη. Ωστόσο για λόγους ασφαλείας και απόδοσης, τα applets μπορούν να δημιουργήσουν αντικείμενα στη μνήμη RAM. Αυτού του είδους τα αντικείμενα καλούνται 'παροδικά', γιατί περιέχουν προσωρινά δεδομένα.

- ***Ατομικές λειτουργίες και συναλλαγές-*** Η Java Card virtual machine επιβεβαιώνει ότι κάθε λειτουργία εγγραφής σε ένα πεδίο αντικειμένου ή σε μια κλάση είναι ατομικό.

5.7 Java Card APIs

Τα Java Card APIs αποτελούνται από ένα σύνολο ειδικών κλάσεων που χρησιμοποιούνται για τον προγραμματισμό εφαρμογών έξυπνων καρτών, οι οποίες είναι συμβατές με το πρότυπο ISO 7816. Υπάρχουν τρία βασικά πακέτα τα οποία είναι: java.lang., javacard.framework.και το javacard.security, ενώ επίσης υπάρχει και ένα επιπλέον extension πακέτο και είναι το java-cardx.crypto.
Πολλές από τις κλάσεις της java πλατφόρμας δεν υποστηρίζονται στα Java Card APIs. Για παράδειγμα κλάσεις της java που χρησιμοποιούνται για GUI διεπαφές, για δίκτυα I/O και για επιτραπέζια συστήματα αρχείων εισόδου / εξόδου δεν υποστηρίζονται. Ο λόγος είναι ότι οι έξυπνες κάρτες δεν έχουν κάποιο display δηλαδή δεν μπορεί να γίνει εισαγωγή στοιχείων από τον χρήστη και χρησιμοποιούν διαφορετικά πρωτόκολλα δικτύου και διαφορετική δομή συστήματος αρχείων.
Οι κλάσεις των Java Card APIs είναι συμπαγείς και περιεκτικοί. Περιλαμβάνουν κλάσεις οι οποίες έχουν προσαρμοστεί από τη Java πλατφόρμα και μπορούν να υποστηρίζουν τη γλώσσα αλλά και κρυπτογραφικές υπηρεσίες.

Το πακέτο java.lang
Αποτελεί ένα υποσύνολο του ομότυπου πακέτου java.lang της java. Οι κλάσεις που υποστηρίζονται είναι οι: Object, Throwable και κάποιες κλάσεις εξαιρέσεων σχετικών με την virtual machine, οι οποίες απεικονίζονται στον παρακάτω πίνακα. Για τις συγκεκριμένες κλάσεις, πολλές από τις μεθόδους της java δεν είναι πλέον διαθέσιμες. Για παράδειγμα η κλάση Object διαθέτει μόνο έναν constructor και τη μέθοδο equals.
Η κλάση Object αποτελεί τη ρίζα στην ιεραρχία κλάσεων της Java Card και η κλάση Trowable παρέχει ένα κοινό 'πρόγονο' για όλες τις εξαιρέσεις

Object	Throwable	Exception
RuntimeException	ArithmeticException	ArrayIndexOutOfBoundsException
ArrayStoreException	ClassCastException	IndexOutOfBoundsException
NullPointerException	SecurityException	NegativeArraySizeException

Εικόνα 5.10

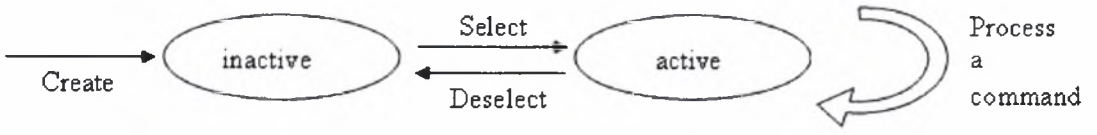
Το πακέτο javacard.framework
Παρέχει κλάσεις και διεπαφές χρήσιμες για την λειτουργία ενός Java Card applet, και κατά συνέπεια η ύπαρξή του είναι ουσιώδης. Η σημαντικότερη κλάση που ορίζει είναι η βασική κλάση Applet η οποία παρέχει μεθόδους για την εκτέλεση του applet και την αλληλεπίδραση του με το JCRE σε όλη τη διάρκεια ζωής του applet. Ο ρόλος του είναι όμοιος με το ρόλο που έχει ένα Java Applet σε έναν browser.
Μια άλλη και εξίσου σημαντική κλάση του πακέτου είναι η κλάση APDU. Τα APDUs όπως ήδη έχουμε αναφέρει είναι τα πακέτα δεδομένων που

ανταλλάσσονται μέσω των πρωτοκόλλων μεταφοράς. Η κλάση αυτή έχει σχεδιαστεί κατά τέτοιο τρόπο ώστε να είναι ανεξάρτητη από το πρωτόκολλο που χρησιμοποιείται κάθε φορά ενώ οι μέθοδοι που παρέχει καθιστούν τη διαχείριση των εντολών APDU πολύ ευκολότερη.
Η κλάση java.lang.System της Java δεν υποστηρίζεται αντί αυτής όμως υπάρχει η κλάση javacard.framework.JCSystem. η οποία περιλαμβάνει μια συλλογή από μεθόδους για τον έλεγχο εκτέλεσης του applet, τη διαχείριση των πόρων και των συναλλαγών και τέλος για το μηχανισμό διαμοιρασμού αντικειμένων μεταξύ των applets(inter-applet object sharing).
Επιπλέον υπάρχει η κλάση PIN για την εξακρίβωση της ταυτότητας του κατόχου της κάρτας.

Τα πακέτα javacard.security και javacardx.crypto
Παρέχουν ένα πλαίσιο εργασίας για την υποστήριξη των κρυπτογραφικών λειτουργιών και η σχεδίαση τους έχει βασιστεί στο πακέτο java.security.
Το συγκεκριμένο πακέτο ορίζει την κλάση KeyBuilder και πολλαπλές διεπαφές για την αναπαράσταση κρυπτογραφικών κλειδιών που χρησιμοποιούνται είτε σε συμμετρικούς (DES) είτε σε ασύμμετρους (DSA, RSA) αλγορίθμους. Επιπλέον παρέχει τις κλάσεις RandomData, Signature και MessageDigest, που χρησιμοποιούνται για τη δημιουργία τυχαίων αριθμών και για τον υπολογισμό συνόψεων μηνυμάτων και υπογραφών. Συναρτήσεις κρυπτογράφησης και αποκρυπτογράφησης παρέχονται μέσω της αφηρημένης κλάσης Cipher.

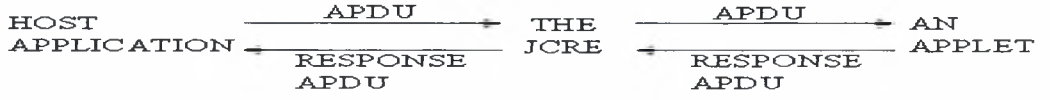
5.8 Java Card Applets

Τα java card applets αν κα έχουν την ίδια ονομασία με τα κοινά java applets δεν θα πρέπει να συγχέονται με αυτά. Ένα java card applet είναι ένα πρόγραμμα σε java το οποίο έχει τη δυνατότητα να τρέχει στο Java Card runtime environment ενώ δεν απαιτείται ένας web browser όπως συμβαίνει με τα java applets. Ο λόγος που έχει επιλεχθεί η ονομασία applets για τις εφαρμογές java card είναι ότι αυτά μπορούν να φορτωθούν στο java card runtime environment αμέσως μετά την κατασκευή της κάρτας. Τα applets δεν χρειάζεται να αποθηκευτούν στην μνήμη ROM της κάρτας κατά τη διάρκεια της κατασκευής της αλλά μπορούν να φορτωθούν δυναμικά σε κάποια μετέπειτα χρονική στιγμή.
Κάθε κλάση applet πρέπει να επεκτείνει την κλάση javacard.framework.Applet. Από τη στιγμή που θα δημιουργηθεί ένα στιγμιότυπο της κλάσης Applet, όπως συμβαίνει και με κάθε είδους αντικείμενο που δημιουργείται αυτό παραμένει 'ζωντανό' μέσα στην κάρτα για πάντα**[25].** Σε μια κάρτα μπορούν να συνυπάρχουν περισσότερα από ένα applet και κάθε ένα από αυτά μπορεί να έχει πολλαπλά στιγμιότυπα.
Αφού αρχικά τα πακέτα που ορίζουν ένα applet φορτωθούν στην Java Card και συνδεθούν με τα υπόλοιπα πακέτα της κάρτας, η ζωή ενός applet αρχίζει τη στιγμή που θα δημιουργηθεί και θα καταχωρηθεί στον JCRE ένα στιγμιότυπο του applet.



5.11 Καταστάσεις του applet

Η επικοινωνία μεταξύ των applets και της host εφαρμογής γίνεται με ανταλλαγή εντολών APDU όπως δείχνει το ακόλουθο διάγραμμα.



5.12 Διάγραμμα επικοινωνίας

5.8.1 Ονομασία των πακέτων και των applets

Στην πλατφόρμα java card κάθε στιγμιότυπο ενός applet έχει ένα μοναδικό αναγνωριστικό αριθμό ο οποίος καλείται application identifier (AID). Το ίδιο συμβαίνει και με τα πακέτα όπου στο κάθε ένα από αυτά ανατίθεται ένας τέτοιος αναγνωριστικός αριθμός AID. Από τη στιγμή λοιπόν που θα φορτωθεί στην κάρτα ένα πακέτο αυτό συνδέεται με τα υπόλοιπα πακέτα μέσω των αριθμών AIDs.
Το πρότυπο ISO 7816 ορίζει ότι τα AIDs μπορούν να χρησιμοποιηθούν ως αναγνωριστικά των εφαρμογών των καρτών, όπως επίσης και συγκεκριμένων τύπων αρχείων, του συστήματος αρχείων της κάρτας. Ένα AID είναι ένας πίνακας από bytes όπως δείχνει το ακόλουθο σχήμα, όπου το πρώτο τμήμα που αποτελείται από 5 bytes ονομάζεται RID(resource identifier), ενώ το δεύτερο τμήμα που έχει μέγεθος από 0 έως 11 bytes ονομάζεται PIX (priority identifier extension). Έτσι συνολικά το μέγεθος ενός AID κυμαίνεται από 5 έως 16 bytes.

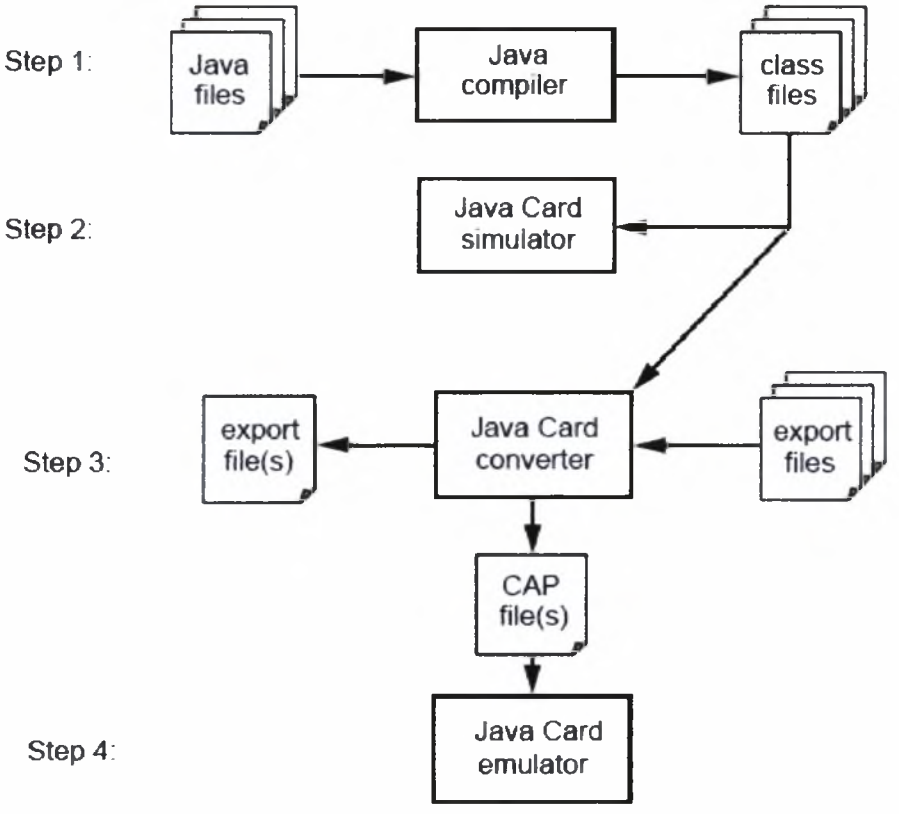
5.13 Application identifier (AID)

Το διεθνές πρότυπο ISO έχει τον έλεγχο ανάθεσης των RIDs στις διάφορες εταιρίες, κάθε μια από τις οποίες έχει ένα μοναδικό κωδικό RID. Στη συνέχεια οι εταιρίες αναλαμβάνουν την διαχείριση και την ανάθεση του τμήματος PIX του κωδικού AID το οποίο μάλιστα περιλαμβάνει μια σύντομη περιγραφή του AID.
Στη Java card πλατφόρμα ο κωδικός AID για ένα πακέτο κατασκευάζεται με σύνθεση του RID της εταιρίας και του PIX για το συγκεκριμένο πακέτο. Κατά παρόμοιο τρόπο δημιουργείται και ο κωδικός AID ενός applet. Πιο συγκεκριμένα αυτό είναι μια σύνθεση του RID του παροχέα του applet και του PIX που αντιστοιχεί στο συγκεκριμένο applet. Ο κωδικός AID ενός applet δεν πρέπει να έχει ίδια τιμή είτε με τον κωδικό AID ενός πακέτου είτε με τον κωδικό AID ενός άλλου applet. Ωστόσο από τη στιγμή που το τμήμα RID ενός κωδικού AID καθορίζει τον παροχέα του applet, ο κωδικός AID του πακέτου και οι κωδικοί AID(s) των applet(s) που ορίζονται μέσα στο συγκεκριμένο πακέτο θα έχουν το ίδιο RID.

Ο κωδικός AID του πακέτου και ο προκαθορισμένος κωδικός AID του κάθε applet που ορίζεται μέσα στο πακέτο περιέχονται στο αρχείο CAP. Αυτά μεταβιβάζονται στον converter αμέσως μετά τη δημιουργία του CAP αρχείου.

5.8.2 Διαδικασία ανάπτυξης applet

Η ανάπτυξη ενός java card applet ξεκινά με τον ίδιο τρόπο που ξεκινά ένα απλό πρόγραμμα σε java. Δηλαδή ο προγραμματιστής γράφει το πρόγραμμα που αποτελείται από μία ή περισσότερες κλάσεις, μεταγλωττίζει τον πηγαίο κώδικα χρησιμοποιώντας έναν java compiler και στην συνέχεια παράγει ένα ή περισσότερα αρχεία με κατάληξη .class. Η παρακάτω εικόνα αναπαριστά τα βήματα ανάπτυξης ενός applet.



5.14 Διαδικασία ανάπτυξης ενός applet

Το applet τρέχει, ελέγχεται και γίνεται το debugging σε ένα περιβάλλον προσομοίωσης όπου προσομοιώνεται το java card runtime environment σε έναν προσωπικό υπολογιστή. Ουσιαστικά δηλαδή το applet τρέχει σε java virtual machine οπότε και παράγεται ο εκτελέσιμος κώδικας δηλαδή τα αρχεία .class. Με τον τρόπο αυτό μπορεί να γίνει η προσομοίωση της λειτουργίας και με χρήση άλλων εργαλείων της java, να ελεγχθεί η συμπεριφορά του applet χωρίς να χρειάζεται να γίνει κάποια διαδικασία μετατροπής τουλάχιστον μέχρι αυτό το στάδιο (δηλαδή μέχρι και το βήμα 2).
Εν συνεχεία το αρχείο .class που παράγεται μετατρέπεται με τη χρήση του Java Card converter σε ένα αρχείο τύπου CAP. Ο Java Card converter μπορεί να πάρει ως είσοδο εκτός από αρχεία τύπου .class και ένα ή περισσότερα αρχεία εξόδου.

Μετά την μετατροπή του πακέτου παράγει ένα αρχείο εξόδου για το συγκεκριμένο πακέτο. Είναι σημαντικό να τονίσουμε ότι κάθε αρχείο CAP ή αρχείο εξόδου αναπαριστά ένα πακέτο java. Αν δηλαδή ένα πακέτο αποτελείται από διαφορετικά πακέτα τότε για κάθε ένα πακέτο από αυτά δημιουργείται και το αντίστοιχο αρχείο CAP ή αρχείο εξόδου.
Στο αμέσως επόμενο βήμα τα αρχεία CAP(s) που παράγονται και αναπαριστούν το applet φορτώνονται και ελέγχονται σε ένα περιβάλλον emulation όπου προσομοιώνεται για δεύτερη φορά το Java Card runtime environment σε ένα προσωπικό υπολογιστή. Η συμπεριφορά του applet στο παραπάνω περιβάλλον προσομοίωσης θα πρέπει να είναι ίδια με τη συμπεριφορά του σε μια πραγματική έξυπνη κάρτα. Σε αυτή τη φάση ανάπτυξης γίνονται μετρήσεις και ελέγχεται η συμπεριφορά του applet καθ' όλη τη διάρκεια εκτέλεσης.
Τελικά αφού ολοκληρωθούν όλοι οι απαραίτητοι έλεγχοι και τα αποτελέσματα είναι ικανοποιητικά τα applet μπορούν πλέον να φορτωθούν και να εγκατασταθούν στην πραγματική έξυπνη Java card.

5.8.3 Διάρκεια ζωής των Java Card Applets

Η ζωή ενός applet ξεκινά από τη στιγμή που θα φορτωθεί στην κάρτα και το JCRE θα καλέσει την στατική μέθοδο Applet.install( ). Στην συνέχεια το applet καταχωρεί τον ευατό του στο JCRE καλώντας την μέθοδο Applet.register( ). Αφού έχει γίνει η εγκατάσταση του και η καταχώρηση του, βρίσκεται σε κατάσταση 'μη επιλεγμένο' αλλά διαθέσιμο για επιλογή και επεξεργασία των APDU εντολών.
Ένα applet επιλέγεται για επεξεργασία μιας APDU εντολής όταν η host εφαρμογή ζητήσει από το JCRE να επιλέξει ένα συγκεκριμένο applet της κάρτας. Αυτό γίνεται με την αποστολή από τον card reader είτε της εντολής SELECT APDU είτε της MANAGE CHANNEL APDU ενώ το JCRE καλει την μέθοδο select( ). Αφού έχει γίνει η επιλογή το JCRE ζητά από το applet να επεξεργαστεί την εντολή APDU με κλήση της μεθόδου του process( ). Εάν η host εφαρμογή ζητήσει ένα άλλο applet από το JCRE τότε αυτό καλεί την μέθοδο του deselect( ) και το applet επιστρέφει σε κατάσταση μη ενεργή και παύει να είναι πια επιλεγμένο.

5.8.4 Διαμοιρασμός αντικειμένων (object sharing) στην Java Card

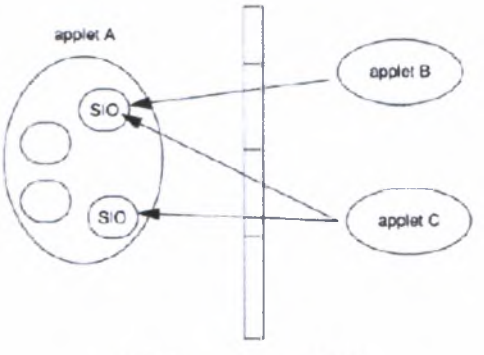
Με την έννοια 'object sharing' εννοούμε την δυνατότητα που έχει ένα applet να αποκτήσει πρόσβαση όχι μόνο σε απλά δεδομένα αλλά και σε μεθόδους αντικειμένων άλλων applets**[27]**. Αυτό γίνεται με τη χρήση του αναγνωριστικού του κάθε applet (AID) έτσι ώστε να καθορίζεται ποιο applet θέλει να αποκτήσει πρόσβαση σε αντικείμενα που έχουν δημιουργηθεί από άλλα applets. Παρακάτω θα δούμε ότι μεταξύ των applets υπάρχει ένα firewall που αποτρέπει ένα applet να αποκτήσει πρόσβαση σε δεδομένα ενός άλλου applet. Ωστόσο υπάρχει η ανάγκη πολλές φορές ένα applet να χρειάζεται να αποκτήσει μια διεπαφή που ανήκει σε ένα άλλο applet ώστε να μπορεί να καλέσει μια μέθοδο αυτής της διεπαφής. Για αυτόν ακριβώς το λόγο είναι απαραίτητος ο μηχανισμός 'object sharing'
Γενικά στην java γνωρίζουμε ότι εάν μια μέθοδος είναι public τότε αυτή είναι προσβάσιμη από άλλα πακέτα και μάλιστα δεν χρειάζεται κάποιος ιδιαίτερος μηχανισμός διαμοιρασμού μεθόδων. Αυτό όμως για την java card πλατφόρμα αποτελεί ένα πρόβλημα. Υπάρχουν κάποια αντικείμενα τα οποία πρέπει να είναι αναγκαστικά public και τα οποία ονομάζονται JCRE entry points objects, όπως για

παράδειγμα είναι το αντικείμενο APDU. Όμως η πρόσβαση σε αυτά δεν θα πρέπει να επιτρέπεται πάντα και για οποιονδήποτε, γιατί τότε οποιοδήποτε applet θα μπορούσε να καλέσει μεθόδους άλλων applets χωρίς να έχει άδεια για αυτό.
Έτσι το JCRE δεν επιτρέπει την κλήση καμιάς μεθόδου μέσα σε άλλα applets εκτός και αν εκτελεστεί πρώτα ο μηχανισμός ***Shareable Interface Objects (SIO)*** .
Στη java card ο χώρος που χρησιμοποιείται για τα αντικείμενα χωρίζεται σε διάφορες περιοχές οι οποίες ονομάζονται contexts. Όλα τα applets τα οποία βρίσκονται στο ίδιο πακέτο μοιράζονται το ίδιο context και απαγορεύεται να αποκτήσουν πρόσβαση σε αντικείμενα που ανήκουν σε διαφορετικό context μέσω του firewall που υπάρχει μεταξύ των πακέτων. Βέβαια το JCRE μπορεί να αποκτήσει πρόσβαση σε αντικείμενα οποιουδήποτε context ή σε ολικούς πίνακες όπως είναι ο APDU buffer οι οποίοι μπορούν να προσπελαστούν από applets οποιουδήποτε context.
Το σύνολο των μεθόδων που ανήκουν στο πακέτο javacard.frame.Shareable δίνει τη δυνατότητα διαμοιρασμού αντικειμένων μεταξύ των applets.
Ένα αντικείμενο το οποίο θα διαμοιραστεί, δηλαδή θα επιτρέψει σε άλλα applets να το προσπελάσουν καλείται shareable interface object (SIO). Αυτά τα αντικείμενα ουσιαστικά είναι applets τα οποία έχουν το ρόλο του server αφού μέσω του μηχανισμού SIO προσφέρουν τις υπηρεσίες τους σε άλλα applets, δηλαδή σε πελάτες. Το ίδιο applet μπορεί να είναι server κάποιων άλλων applets και παράλληλα να είναι πελάτης ενός άλλου server.

5.8.5 Βήματα μηχανισμού SIO

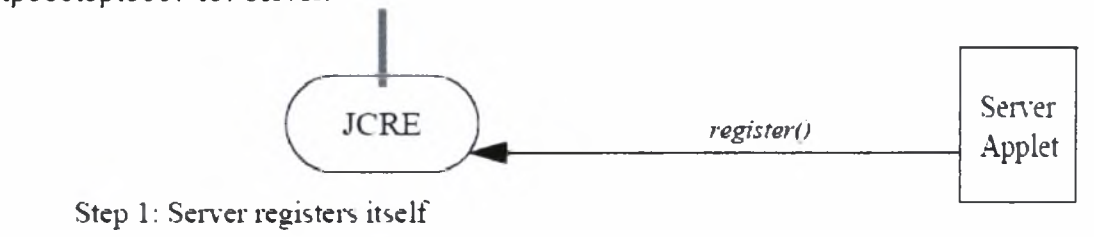
Βήμα 1°

Δημιουργία μιας SIO:
Για να δημιουργηθεί μια νέα SIO θα πρέπει το applet που αποτελεί τον server να ορίσει μια διεπαφή έστω X η οποία να επεκτείνει την διεπαφή javacard.framework.Shareable. Στην συνέχεια ο server θα πρέπει να ορίσει μια κλάση έστω C η οποία να υλοποιεί την παραπάνω διεπαφή X και να δημιουργήσει ένα στιγμιότυπο O της κλάσης C. Το αντικείμενο O είναι τώρα ένα αντικείμενο SIO.



5.15 Αντικείμενα SIO

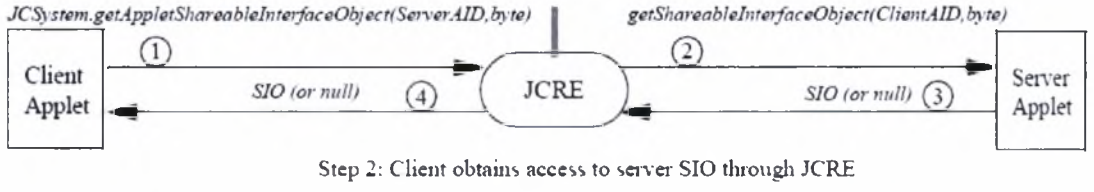
Ένα στιγμιότυπο του applet server με το αναγνωριστικό AID καταχωρείται στον JCRE. Αυτό το αναγνωριστικό χρησιμοποιείται από τα applets πελάτες για να προσδιορίσουν τον server.



5.16 Βήμα 1°

Βήμα 2°

Απόκτηση μιας SIO:
Ένα applet που είναι πελάτης για να προσπελάσει το αντικείμενο Ο θα πρέπει να αποκτήσει τη SIO. Αυτή η διαδικασία απεικονίζεται παρακάτω:

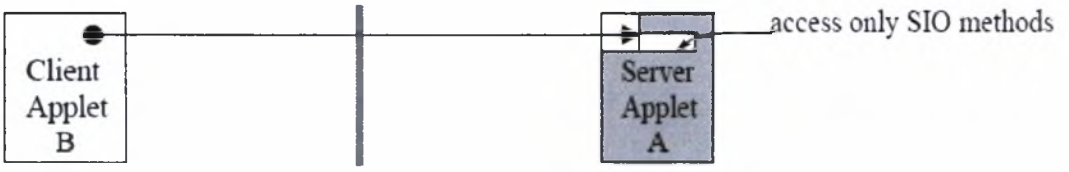


5.17 Βήμα 2°

Για να αποκτήσει ένας πελάτης πρόσβαση στο αντικείμενο Ο θα πρέπει να δημιουργήσει μια αναφορά σε αντικείμενο έστω BO, τύπου X και να καλέσει τη μέθοδο getAppletShareableInterfaceObject με ορίσματα το AID και ένα byte που θα ορίζει ποια διεπαφή θα χρησιμοποιηθεί (για εκείνους τους servers οι οποίοι διαθέτουν πάνω από μια διεπαφές). Το JCRE ψάχνει το applet-server το οποίο σχετίζεται με το AID και προωθεί την αίτηση στον server, αντικαθιστώντας το πρώτο όρισμα με το AID του πελάτη. Ο server λαμβάνει την αίτηση και το AID του αποστολέα και αποφασίζει αν θα μοιραστεί το αντικείμενο Ο. Εάν ο server αποφασίσει θετικά επιστρέφει μια αναφορά στο αντικείμενο Ο διαφορετικά επιστρέφει null. Έπειτα το JCRE προωθεί την αναφορά αυτή στον πελάτη που ζήτησε το αντικείμενο Ο. Τέλος ο πελάτης λαμβάνει την αναφορά αυτή που είναι τύπου SIO κάνει casting και το μετατρέπει σε τύπου X και το αποθηκεύει στο BO.

Βήμα 3°

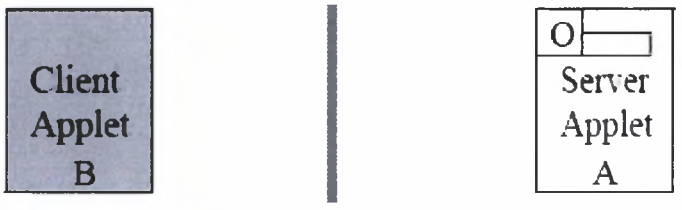
Χρήση ενός αντικειμένου SIO:
Αφού έχει γίνει η απόκτηση ενός αντικειμένου SIO, ο πελάτης δηλαδή το applet B όπως βλέπουμε στην εικόνα μπορεί να καλέσει κάθε μέθοδο της διεπαφής X με την αναφορά BO, το οποίο στη συνέχεια μπορεί να προσπελάσει το αντικείμενο Ο. Βλέπουμε ότι μεταξύ του πελάτη και του server υπάρχει ένα firewall και μάλιστα το μόνο ορατό κομμάτι του server, στο firewall είναι το αντικείμενο Ο.



Step 3: Client has 'transparent' access only to the server SIO

5.18 Χρήση του SIO

Οταν ο πελάτης καλεί μια μέθοδο του BO τότε πραγματοποιείται εναλλαγή περιβάλλοντος 'context switch' στο JCRE η οποία οδηγεί στην κατάσταση που παρουσιάζεται σχηματικά στην εικόνα



5.19 Context Switch

Στην παραπάνω κατάσταση ο πελάτης δηλαδή το applet B δεν είναι καθόλου ορατό στον server δηλαδή στο applet A, τα μόνα δεδομένα τα οποία είναι ορατά είναι τα ορίσματα που περνούν στη στοίβα και ο πίνακας APDU που είναι καθολική μεταβλητή.

5.9 Java Card Technology και Ασφάλεια

Η Java Card Technology σχεδιάστηκε εξαρχής με τέτοιο τρόπο ώστε να παρέχει πιο ασφαλή και αξιόπιστα συστήματα – εφαρμογές. Η γλώσσα προγραμματισμού Java παρέχει έναν μεγάλο αριθμό πλεονεκτημάτων σε σύγκριση με άλλες γλώσσες προγραμματισμού. Έτσι σε αντίθεση με τις άλλες γλώσσες προγραμματισμού που χρησιμοποιούνται για την ανάπτυξη εφαρμογών έξυπνων καρτών όπως είναι η assemply ή C οι java card εφαρμογές ενθυλακώνουν ευαίσθητα δεδομένα και αλγορίθμους σε αντικείμενα, αφού πρόκειται για αντικειμενοστραφής γλώσσα προγραμματισμού, οι οποίες έχουν αποδεδειγμένη συμπεριφορά και αυξημένη ασφάλεια. Εκτός από τα πλεονεκτήματα της γλώσσας, η java card πλατφόρμα περιλαμβάνει χαρακτηριστικά που παρέχουν επιπρόσθετη ασφάλεια όπως είναι η ύπαρξη firewall μεταξύ των applets, η transaction atomicity, οι κρυπτογραφικές κλάσεις και τέλος η διαχωρισμένη αρχιτεκτονικά java card virtual machine**[26].**

5.9.1 Πλεονεκτήματα της γλώσσας Java

- Πρόκειται για αντικειμενοστραφή γλώσσα προγραμματισμού, η οποία εισάγει στον προγραμματιστή τις έννοιες ενθυλάκωσης δεδομένων με διεργασίες που έχουν αποκλειστική πρόσβαση σε αυτά και μπορούν να τα επεξεργαστούν.
- Για κάθε μέθοδο, κλάση και τύπο δεδομένων υπάρχουν τα χαρακτηριστικά 'public','private','protected' και 'package-protected' έτσι ώστε να υπάρχει

έλεγχος πρόσβασης στις διάφορες λειτουργίες – μεθόδους που παρέχουν τα αντικείμενα.
- Η ενθυλάκωση αντικειμένων επιτρέπει στους προγραμματιστές την επαναχρησιμοποίηση κώδικα ο οποίος έχει είδη ελεγχθεί.
- Δεν υποστηρίζει δείκτες όπως η C ή C++ και έτσι αποφεύγονται κίνδυνοι ασφαλείας που ενδέχεται να προκύψουν με χρήση τέτοιων τύπων.
- Παρέχει προσωρινή δέσμευση χώρου αποτρέποντας προγραμματιστικά λάθη που μπορούν να προκύψουν από άσκοπη επαναχρησιμοποίηση μνήμης.

5.9.2 Χαρακτηριστικά ασφάλειας της πλατφόρμας Java Card

Η πλατφόρμα Java Card κληρονομώντας κάποια χαρακτηριστικά από τη γλώσσα προγραμματισμού Java, έχει βελτιώσει σημαντικά το επίπεδο ασφάλειας. Στοιχεία όπως : transaction atomicity, τείχος προστασίας(firewall) μεταξύ των applet, κλάσεις που υποστηρίζουν κρυπτογραφικές λειτουργίες, ψηφιακές υπογραφές και αυθεντικοποίηση των CAP αρχείων και τέλος ο διαχωρισμός της java virtual machine σε δύο τμήματα συμβάλουν στην ανάπτυξη ασφαλών συστημάτων έξυπνων καρτών.

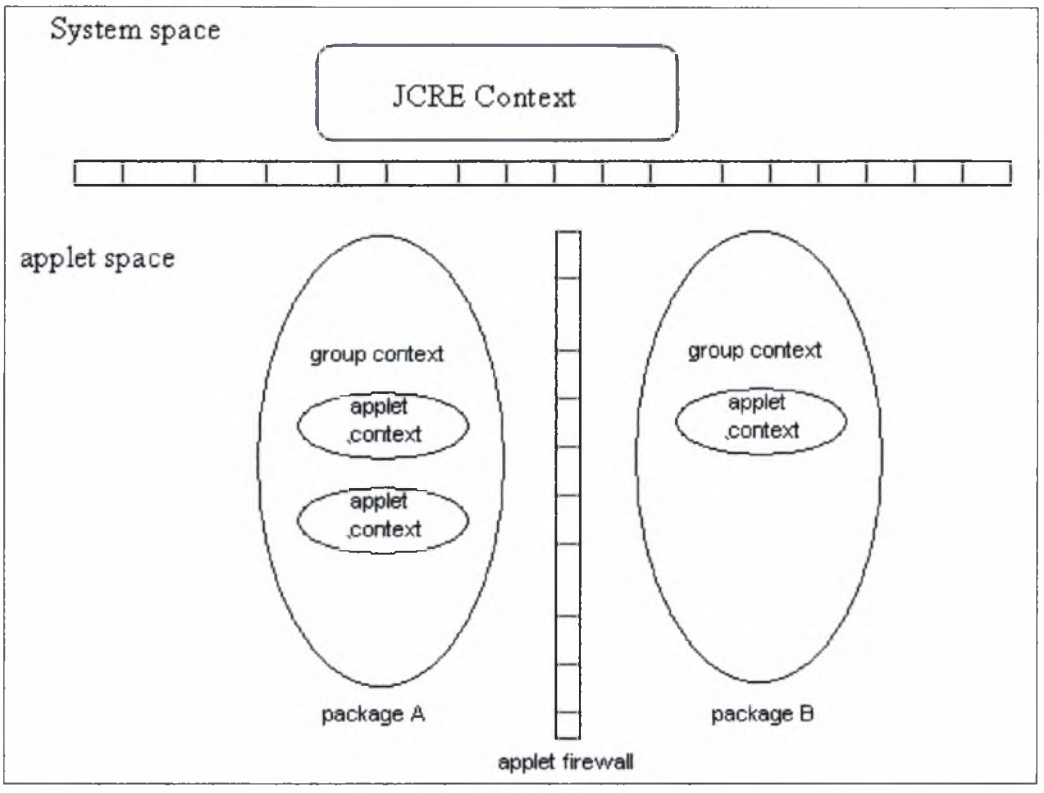
Transaction Atomicity

Σε μια διαδικασία συναλλαγής είτε θα πραγματοποιηθούν όλες οι ενημερώσεις των δεδομένων, δηλαδή τα δεδομένα που σχετίζονται με την συγκεκριμένη συναλλαγή θα πάρουν τις νέες τιμές τους, εάν η συναλλαγή ολοκληρωθεί με επιτυχία ή θα παραμείνουν όλες όπως ήταν πριν την συναλλαγή εάν συμβεί κάποιο απρόβλεπτο γεγονός όπως διακοπή ρεύματος ή απότομη αφαίρεση της κάρτας από τη συσκευή ανάγνωσης .

Η ιδιότητα αυτή παίζει σπουδαίο ρόλο στην ασφάλεια της κάρτας. Ας υποθέσουμε για παράδειγμα ότι η κάρτα βγαίνει από τον αναγνώστη απότομα κατά τη διάρκεια μιας συναλλαγής μεταξύ κάρτας και συσκευής ανάγνωσης, όπου υποτίθεται ότι θα έπρεπε να ενημερωθεί το μυστικό κρυπτογραφικό κλειδί. Η κάρτα σε μια τέτοια περίπτωση επιστρέφει στην προηγούμενη της κατάσταση και η εφαρμογή που τρέχει στην συσκευή υποδοχής γνωρίζει τότε ότι η διαδικασία ενημέρωσης δεν έχει γίνει.

Τείχος προστασίας μεταξύ των applets

Η πλατφόρμα Java Card παρέχει ένα ασφαλές περιβάλλον εκτέλεσης. Αυτό επιτυγχάνεται με την ύπαρξη ενός firewall μεταξύ των διαφορετικών applets που βρίσκονται μέσα στην ίδια κάρτα. Έτσι ένα java card applet απομονώνεται από τα υπόλοιπα applets μέσω του firewall, με το οποίο το java card runtime environment ελέγχει την χρήση των δεδομένων που είναι αποθηκευμένα σε αντικείμενα που διαμοιράζονται. Με τον μηχανισμό αυτό δίνεται σε ένα applet, παροδικά βέβαια, η αποκλειστική πρόσβαση στην μνήμη της κάρτας έτσι ώστε κανένα άλλο applet να μην μπορεί να επηρεάσει την λειτουργικότητα της κάρτας.



5.20 Applet firewall

Κλάσεις κρυπτογραφίας και ασφάλειας

Οι κλάσεις του java card framework που σχετίζονται με θέματα κρυπτογραφίας και ασφάλειας παρέχουν τα ακόλουθα:

- Συμμετρικούς αλγορίθμους κρυπτογράφησης και αποκρυπτογράφησης
- Ασύμμετρους αλγορίθμους κρυπτογράφησης και αποκρυπτογράφησης
- Δημιουργία ψηφιακών υπογραφών και πιστοποίηση
- Συνόψεις μηνυμάτων
- Δημιουργία τυχαίων αριθμών
- Διαχείριση κωδικών PIN

Η αρχιτεκτονική σχεδίαση της java card virtual machine

Έχουμε ήδη αναφέρει ότι η JCVM είναι διαχωρισμένη σε δύο τμήματα το ένα εσωτερικά στην κάρτα και το άλλο εκτός αυτής. Ο λόγος που σχεδιάστηκε με αυτόν τον μοναδικό και πρωτοπόρο τρόπο είναι ότι μειώνεται το μέγεθος του applet που θα πρέπει να φορτωθεί στην κάρτα και επίσης μειώνονται οι απαιτήσεις σε μνήμη κατά τη διάρκεια της εκτέλεσης.

Το εξωτερικό κομμάτι εκτός από τον converter που μετατρέπει τα applets σε CAP αρχεία μπορεί να περιλαμβάνει και εργαλεία όπως για παράδειγμα έναν Verifier, ρόλος του οποίου είναι να πιστοποιεί τα αρχεία CAP. Θα πρέπει δηλαδή να βεβαιώνει ότι τα περιεχόμενα της κάρτας είναι συμβατά με τις προδιαγραφές της java card τεχνολογίας και ότι ο εκτελέσιμος κώδικας που περιλαμβάνεται στα αρχεία δεν εκθέτει την ακεραιότητα της java card virtual machine.

Τέλος προτού ένα applet εγκατασταθεί σε μια κάρτα περνάει από το στάδιο της πιστοποίησης και της αυθεντικοποίησης που υλοποιούνται μέσω μηχανισμών που καλούνται 'code loading' και εξασφαλίζουν ότι ένα java card applet δεν έχει τροποποιηθεί πριν τη φόρτωσή του στην κάρτα.

Κεφάλαιο 6

Πειραματική Υλοποίηση Αυθεντικοποίησης με Έξυπνες Κάρτες

6.1 Η έξυπνη φοιτητική κάρτα

Στο κεφάλαιο 3 Αρχιτεκτονική και Τεχνολογία Έξυπνων καρτών, μελετήσαμε τα βήματα της διαδικασίας κατασκευής της έξυπνης κάρτας. Είδαμε ότι προτού η κάρτα εκδοθεί στον χρήστη θα πρέπει να γίνει η προσωποποίηση της δηλαδή να αποθηκευτούν στην μνήμη EEPROM τα προσωπικά δεδομένα του κατόχου της κάρτας όπως επίσης και οι απαραίτητοι κωδικοί ώστε αυτός να μπορεί να πιστοποιεί την ταυτότητα του και να αποκτά πρόσβαση στους πόρους και τις υπηρεσίες του πληροφοριακού συστήματος που χρησιμοποιείται.
Μια έξυπνη κάρτα μοιάζει εξωτερικά με τις γνωστές πλαστικές κάρτες όπως είναι οι πιστωτικές κάρτες των τραπεζών. Εκείνο που τις κάνει να ξεχωρίζουν από αυτές τις κοινές κάρτες μνήμης είναι το ενσωματωμένο ολοκληρωμένο κύκλωμα που είναι τυπωμένο στη μια τους όψη. Αυτό το chip δεν είναι απλά μια αποθήκη δεδομένων και πληροφοριών αλλά είναι έναν μικροεπεξεργαστής ο οποίος έχει τη δυνατότητα να χειρίζεται τις πληροφορίες που είναι αποθηκευμένες στη μνήμη της κάρτας, δηλαδή να προσθέτει δεδομένα, να διαγράφει και να τροποποιεί, όπως επίσης να εκτελεί αριθμητικές ή και πιο σύνθετες πράξεις. Ακόμη έχει τη δυνατότητα επικοινωνίας μέσω μια θύρας εισόδου/ εξόδου (I/O port) και κατ' επέκταση της υλοποίησης αλγορίθμων κρυπτογράφησης και αυθεντικοποίησης αυξάνοντας σημαντικά το επίπεδο ασφάλειας της κάρτας.

6.1.1 Data Set έξυπνης φοιτητικής κάρτας

Όπως αναφέραμε στην πρώτη παράγραφο οι έξυπνες κάρτες έχουν τη δυνατότητα να αποθηκεύουν δεδομένα και να τα επεξεργάζονται. Ανάλογα με το σκοπό της εφαρμογής η οποία χρησιμοποιεί τις έξυπνες κάρτες, τα δεδομένα και οι πληροφορίες που πρόκειται να αποθηκευτούν στη μνήμη της κάρτας ποικίλουν.
Έτσι για παράδειγμα εάν πρόκειται για μια έξυπνη κάρτα η οποία χρησιμοποιείται ως ηλεκτρονικό πορτοφόλι τότε σε αυτή θα πρέπει να αποθηκεύεται το χρηματικό υπόλοιπο με το οποίο ο κάτοχος θα μπορεί να πραγματοποιεί τις συναλλαγές του καθώς επίσης και οι εφαρμογές με τις οποίες θα γίνεται η μεταφορά των χρημάτων η αύξηση και μείωση του υπολοίπου.
Ένα άλλο παράδειγμα είναι οι κάρτες υγείας οι οποίες θα πρέπει να περιλαμβάνουν προσωπικά και ασφαλιστικά στοιχεία, κωδικοποιημένα στοιχεία για το ιατρικό ιστορικό του κατόχου της κάρτας, κωδικούς πρόσβασης σε βάσεις δεδομένων με το πλήρες ιατρικό αρχείο και άλλες εφαρμογές όπως η ηλεκτρονική συνταγογράφηση φαρμάκων .

Στην περίπτωση των *έξυπνων φοιτητικών καρτών*, τα δεδομένα και πληροφορίες που θα περιέχονται θα πρέπει να προσδιορίζουν μονοσήμαντα τον κάθε φοιτητή. Έτσι η κάρτα θα πρέπει να περιλαμβάνει κάποια προσωπικά δεδομένα όπως το όνομα και το επώνυμο, στοιχεία σχετικά με την εκπαίδευση, όπως για παράδειγμα το έτος εισαγωγής στο πανεπιστήμιο καθώς και δεδομένα τα οποία σχετίζονται με υπηρεσίες που παρέχει το πανεπιστήμιο, όπως είναι οι βιβλιοθήκες, οι φοιτητικές εστίες και οι φοιτητικές λέσχες.
Με βάση λοιπόν τα παραπάνω διακρίνουμε τα ακόλουθα πεδία τα οποία θα πρέπει να περιέχονται στην έξυπνη φοιτητική κάρτα :

- **Όνομα**: *το όνομα του φοιτητή και κατόχου της κάρτας*
- **Επώνυμο**: *το επώνυμο του φοιτητή και κατόχου της κάρτας*
- **Έτος εισαγωγής**: *το έτος εισαγωγής του φοιτητή στο πανεπιστήμιο*
- **Διεύθυνση κατοικίας**: *η διεύθυνση κατοικίας του φοιτητή*
- **E-mail**: *η ηλεκτρονική διεύθυνση του φοιτητή*
- **Τηλέφωνο**: *το τηλέφωνο του φοιτητή*
- **Access key (password)**: *ο κωδικός τον οποίο θα χρησιμοποιεί ο φοιτητής σε συνδυασμό με την κάρτα του για την χρήση των παρεχόμενων υπηρεσιών*
- **Certificates**: *ψηφιακά πιστοποιητικά*
- **Ελεύθερα πεδία**: *για οποιαδήποτε μελλοντική χρήση και νέα πρόσθετες εφαρμογές*

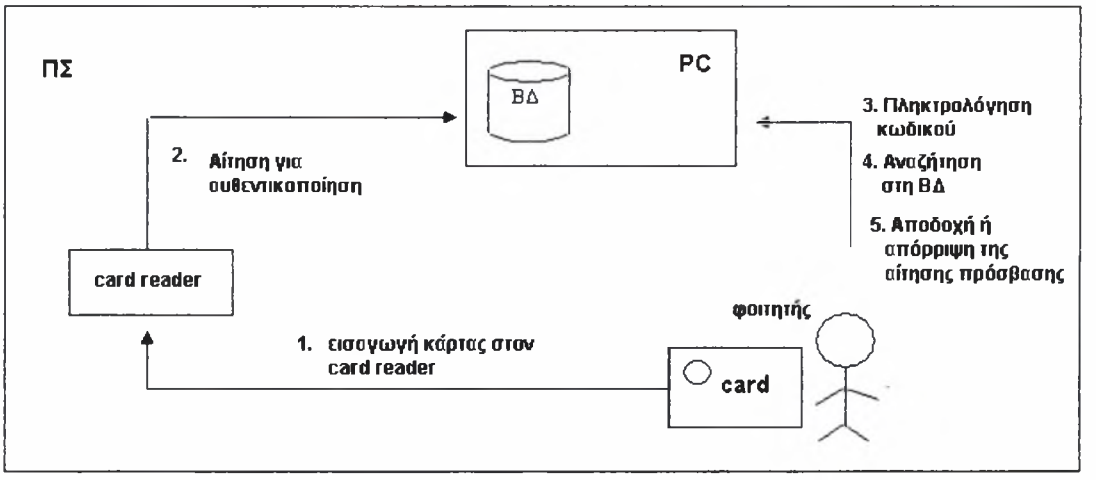
6.2 Σενάριο υλοποίησης

Οι υπηρεσίες στις οποίες θα μπορούν να αποκτούν πρόσβαση οι σπουδαστές με χρήση της κάρτας τους, πηγάζουν από τις ανάγκες τους ως φοιτητές, στην πανεπιστημιακή κοινότητα και καλούνται να βελτιώσουν τις συνθήκες φοίτησης τους. Αυτές οι εφαρμογές θα πρέπει να περιλαμβάνουν τα ακόλουθα :

1. **Αυθεντικοποίηση του χρήστη για απόκτηση πρόσβασης στα PC των εργαστηρίων**

Αποτελεί μια από τις βασικότερες εφαρμογές για τους σπουδαστές καθώς με αυτό τον τρόπο μπορούν να αποκτήσουν πρόσβαση στους υπολογιστές των εργαστηρίων, να παρακολουθούν τα εργαστηριακά τους μαθήματα, να χρησιμοποιούν τα προγράμματα που είναι εγκατεστημένα, να αποθηκεύουν στον προσωπικό τους χώρο (που τους διατίθεται από το πανεπιστήμιο) αρχεία και προσωπικά δεδομένα και τέλος να μπορούν να συνδέονται με ηλεκτρονικά δίκτυα όπως είναι το internet.

Το σενάριο υλοποίησης αυτής τη υπηρεσίας παριστάνεται στο ακόλουθο διάγραμμα:



6.1 Σενάριο υλοποίησης της αυθεντικοποίησης ενός φοιτητή με έξυπνη κάρτα

Τα συστατικά από τα οποία θα αποτελείται το πληροφοριακό αυτό σύστημα όπως φαίνονται στην παραπάνω εικόνα θα είναι τα ακόλουθα:

- **Βάση δεδομένων :** η οποία αποθηκεύει τα στοιχεία των φοιτητών που είναι εγγεγραμμένοι στο τμήμα και καταχωρεί τους κωδικούς τους ώστε να μπορεί να γίνει η σύγκριση με τον κωδικό που εισάγει ο φοιτητής και να επιτραπεί ή αποτραπεί η πρόσβασή του στους ηλεκτρονικούς υπολογιστές.
- **Έξυπνη κάρτα φοιτητή :** την οποία θα πρέπει να έχει στην κατοχή του ο κάθε φοιτητής και θα την χρησιμοποιεί σε συνδυασμό με τον προσωπικό του κωδικό για να μπορεί να χρησιμοποιεί τους υπολογιστές του εργαστηρίου.
- **Smart Card Reader:** η συσκευή ανάγνωσης των έξυπνων καρτών η οποία θα πρέπει να συνδέεται με τον υπολογιστή στον οποίο θέλει να αποκτήσει πρόσβαση ο φοιτητής.
- **Προσωπικός υπολογιστής:** ο οποίος θα παρέχει το interface για την αυθεντικοποίηση των φοιτητών, σύμφωνα με την οποία ο χρήστης θα μπορεί να εισάγει τον προσωπικό του κωδικό.
- **Administrator:** ο οποίος θα διαχειρίζεται τη βάση δεδομένων και τους κωδικούς των φοιτητών και θα μπορεί να προσθέτει νέους χρήστες ή να αφαιρεί χρήστες που έχουν ολοκληρώσει τις σπουδές τους και τέλος να μπορεί να προσδίδει σε αυτούς δικαιώματα πρόσβασης στους πόρους του υπολογιστικού συστήματος όπως για παράδειγμα να επιτρέπει ή όχι την εγγραφή ή ανάγνωση αρχείων, την εγκατάσταση ή όχι προγραμμάτων και να απαγορεύει την πρόσβαση σε προσωπικά δεδομένα άλλων χρηστών.

Ακολουθούν τα *βήματα* του σεναρίου υλοποίησης :

1. Ο φοιτητής και κάτοχος της java card εισάγει την κάρτα στον card reader
2. Ξεκινά η επικοινωνία της κάρτας με τον card reader και του card reader με τον υπολογιστή.
3. Ο φοιτητής ζητά πρόσβαση στον υπολογιστή

4. Πληκτρολογεί τον προσωπικό του κωδικό στην εφαρμογή που εμφανίζεται στην οθόνη του υπολογιστή.
5. Γίνεται η επεξεργασία των δεδομένων, αναζητείται ο κωδικός του χρήστη στη βάση δεδομένων του administrator και αν υπάρχει ο κωδικός επιτρέπεται η πρόσβαση. Διαφορετικά εάν ο χρήστης έχει δώσει λάθος κωδικό μπορεί να επαναλάβει την διαδικασία. Εάν ο κωδικός δεν είναι σωστός απαγορεύεται η πρόσβαση.

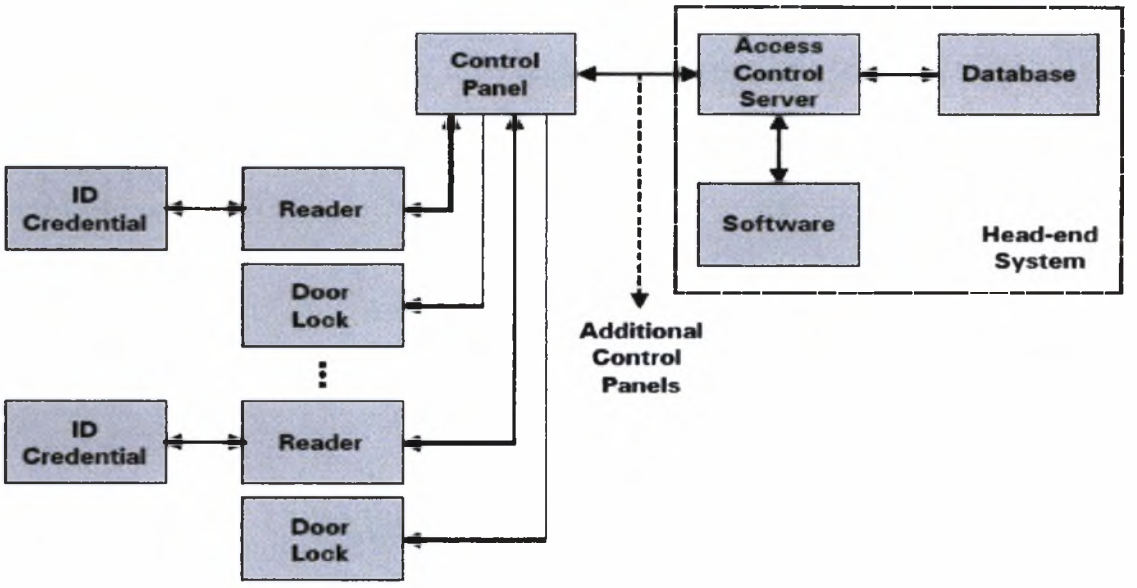
2. Πρόσβαση στους φυσικούς χώρους του πανεπιστημίου

Στους χώρους της πανεπιστημιούπολης δικαίωμα πρόσβασης πρέπει να έχουν οι σπουδαστές και φυσικά το προσωπικό του πανεπιστημίου, ενώ αντίθετα η είσοδος σε αυτούς τους χώρους πρέπει να απαγορεύεται σε οποιονδήποτε τρίτο που δεν έχει είτε την ιδιότητα του σπουδαστή, είτε δεν ανήκει στο προσωπικό του τμήματος. Έτσι με χρήση της κάρτας, οποιοσδήποτε φοιτητής μπορεί να εισέρχεται για παράδειγμα στα εργαστήρια οποιαδήποτε στιγμή είναι αυτό απαραίτητο. Επιπλέον αυτός ο τρόπος εξασφαλίζει και βελτιώνει την προστασία του υλικού εξοπλισμού του πανεπιστημίου.

Ένα σύστημα ελέγχου φυσικής πρόσβασης στους πανεπιστημιακούς χώρους, ώστε να επιτρέπεται η είσοδος σε αυτούς, μόνο σε εξουσιοδοτημένους χρήστες, με χρήση έξυπνης κάρτας θα πρέπει να αποτελείται από τα ακόλουθα συστατικά :

- **Έξυπνη κάρτα**
- **Αναγνώστης έξυπνης κάρτας (ο οποίος θα είναι τοποθετημένος δίπλα στην πόρτα εισόδου)**
- **Door lock**
- **Control panel**
- **Access control server (ο οποίο περιλαμβάνει τα ακόλουθα)**
    - **Το απαραίτητο λογισμικό**
    - **Βάση δεδομένων**

Η παρακάτω εικόνα δείχνει πως διασυνδέονται τα βασικά αυτά συστατικά και εξηγείται η διαδικασία της αυθεντικοποίησης.



6.2 Συστατικά συστήματος ελέγχου φυσικής πρόσβασης με έξυπνες κάρτες

Η διαδικασία ελέγχου φυσικής πρόσβασης, ξεκινά από τη στιγμή που ο χρήστης εισάγει την έξυπνη κάρτα στον αναγνώστη ο οποίος συνήθως βρίσκεται τοποθετημένος δίπλα στην πόρτα εισόδου. Ο αναγνώστης εξάγει δεδομένα από την κάρτα, τα επεξεργάζεται και τα αποστέλλει στο συστατικό control panel.
Ακολούθως το συστατικό control panel στέλνει τα δεδομένα στον access control server όπου εκεί λαμβάνει χώρα η σύγκριση των δεδομένων που περιέχονται στην κάρτα και έχουν παραληφθεί από τον access control server με τα δεδομένα που βρίσκονται τοποθετημένα για τον συγκεκριμένο χρήστη στη βάση δεδομένων. Το λογισμικό του access control server αποφασίζει για τα δικαιώματα πρόσβασης και αυθεντικοποίησης του χρήστη. Εάν η διαδικασία αυθεντικοποίησης είναι επιτυχής ο server στέλνει σήμα στο control panel ώστε να ξεκλειδωθεί η πόρτα. Έπειτα το control panel στέλνει σήμα στον card reader ο οποίος μπορεί είτε να παράγει έναν ήχο είτε να σηματοδοτήσει με έναν άλλο τρόπο τον χρήστη ότι μπορεί να εισέλθει στον χώρο.

3. Πρόσβαση στις βάσεις δεδομένων της Γραμματείας – interactive λειτουργίες με τις γραμματείες

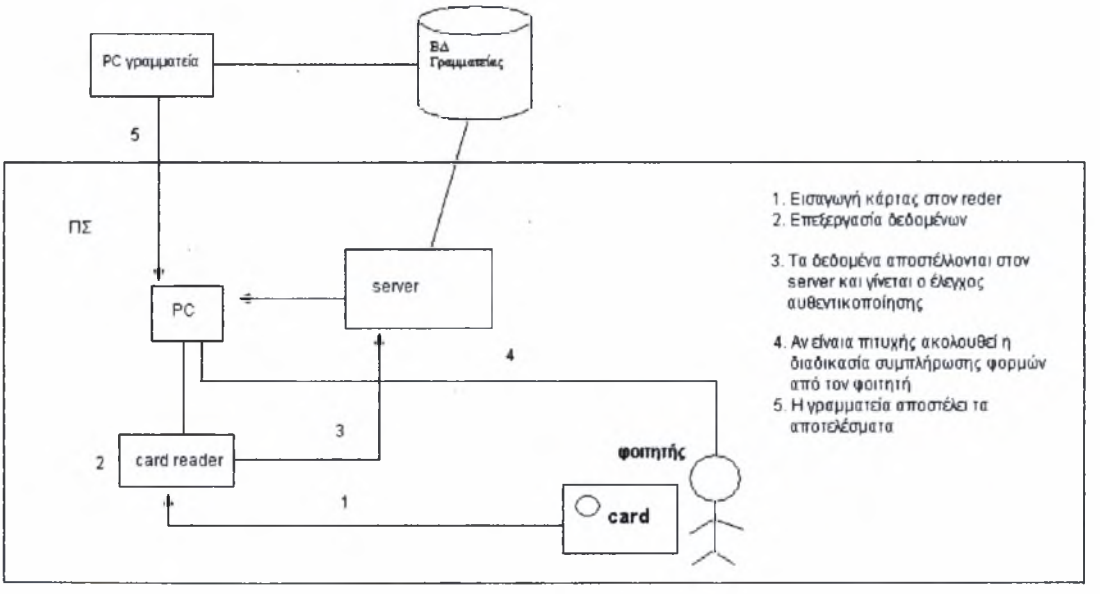
Υπηρεσίες όπως για παράδειγμα:

- η εγγραφή ενός φοιτητή στο τμήμα ή σε κάποια μαθήματα του εξαμήνου στο οποίο βρίσκεται
- η έκδοση πιστοποιητικών, βεβαιώσεων και αναλυτικών βαθμολογιών
- η ενημέρωση για την πρόοδο του κάθε φοιτητή ή η αξιολόγηση του στο κάθε μάθημα

θα μπορούσαν να παρέχονται μέσω ενός πληροφοριακού συστήματος με χρήση της έξυπνης κάρτας του κάθε φοιτητή. Έτσι οι διαδικασίες που απαιτούνται στην παρούσα φάση για την παροχή όλων των παραπάνω υπηρεσιών οι οποίες μάλιστα είναι αρκετά χρονοβόρες, θα μπορούσαν να αυτοματοποιηθούν και να συμβάλουν με αυτό τον τρόπο στην άμεση εξυπηρέτηση των φοιτητών.

Τα βασικά συστατικά από τα οποία θα πρέπει να αποτελείται ένα τέτοιο πληροφοριακό σύστημα το οποίο θα επιτρέπει στους φοιτητές να αλληλεπιδρούν με τις βάσεις δεδομένων της γραμματείας ώστε να μπορούν να τους εκδίδονται πιστοποιητικά, αναλυτικές βαθμολογίες και να τους παρέχονται υπηρεσίες εγγραφής και δήλωσης μαθημάτων άμεσα και με αυτόματο τρόπο, περιλαμβάνουν τα ακόλουθα:

- **Έξυπνη κάρτα φοιτητή**
- **Smart Card Reader**
- **Βάση δεδομένων της γραμματείας :** η οποία αποθηκεύει τα στοιχεία των φοιτητών που είναι εγγεγραμμένοι στο τμήμα και όλες τις απαραίτητες πληροφορίες στις οποίες θα μπορούν και θα επιτρέπεται να έχουν πρόσβαση.
- **PC:** με το κατάλληλο λογισμικό που θα παρέχει τα απαραίτητα interfaces για την διεκπεραίωση της υπηρεσίας, όπως για παράδειγμα κάποιες ηλεκτρονικές φόρμες που θα πρέπει να συμπληρώνει ο φοιτητής ανάλογα με την κάθε περίπτωση.
- **Access server:** συστατικό απαραίτητο για την διαχείριση των χρηστών και των δικαιωμάτων τους. Θα συνδέεται με τη βάση δεδομένων της γραμματείας και θα αποφασίζει για την αυθεντικοποίηση των χρηστών και την ικανοποίηση των αιτήσεων τους.



6.3 Σενάριο υλοποίησης αλληλεπίδρασης φοιτητών με την γραμματεία

Ακολουθούν τα *βήματα* του σεναρίου υλοποίησης της συγκεκριμένης υπηρεσίας :

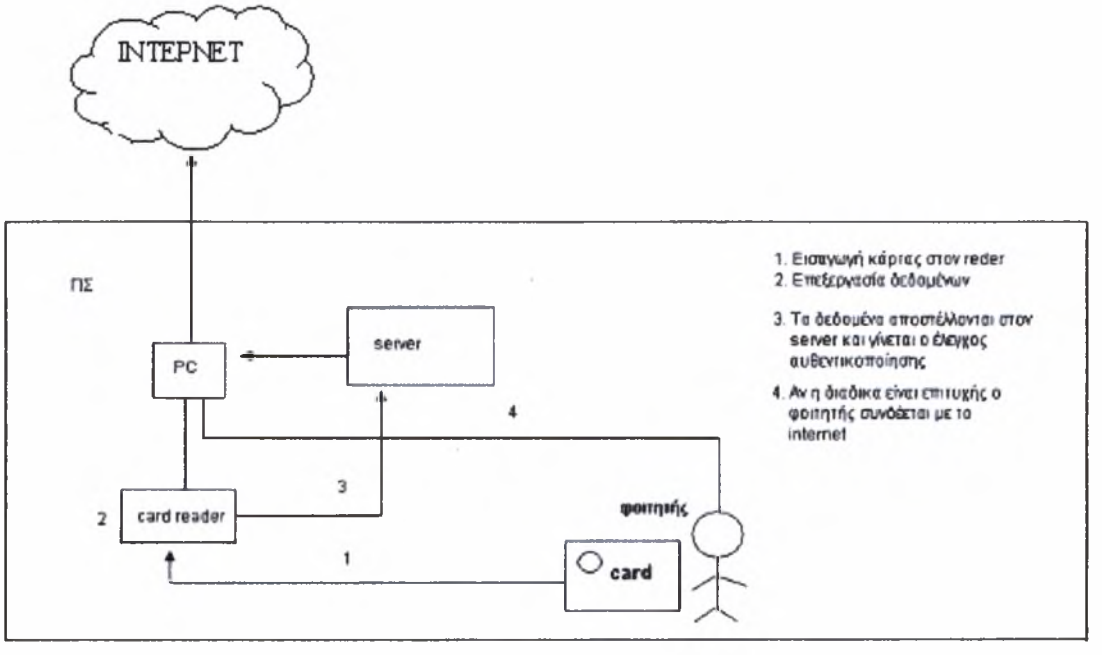
1. Ο φοιτητής και κάτοχος της java card εισάγει την κάρτα στον card reader ο οποίος συνδέεται με κάποιον υπολογιστή
2. Ξεκινά η επικοινωνία της κάρτας με τον card reader και του card reader με τον υπολογιστή.
3. Ο card reader εξάγει τα απαραίτητα δεδομένα από την κάρτα και τα επεξεργάζεται.
4. Εν συνεχεία τα στέλνει στον server ο οποίος αποφασίζει για το αν θα επιτραπεί η πρόσβαση του χρήστη στις υπηρεσίες τις οποίες ζητά.

5. Εάν η αυθεντικοποίηση είναι επιτυχής θα πρέπει να εμφανίζονται στην οθόνη του υπολογιστή κάποιες φόρμες τις οποίες θα πρέπει να συμπληρώνει όπως για παράδειγμα φόρμες για εγγραφή, αναλυτική βαθμολογία , δήλωσης μαθημάτων, αξιολόγησης μαθημάτων (ώστε να μπορεί να βλέπει την βαθμολογία του σε κάποιο συγκεκριμένο μάθημα).
6. Αφού συμπληρωθούν οι φόρμες θα πρέπει να αποστέλλονται στην γραμματεία.
7. Τέλος η γραμματεία θα πρέπει να στέλνει τα αποτελέσματα στους φοιτητές για παράδειγμα μέσω ηλεκτρονικού ταχυδρομείου.

4. Πρόσβαση στο internet

Κάθε σπουδαστής έχει το δικό του λογαριασμό για πρόσβαση στο internet, με χρήση dialup σύνδεσης. Η σύνδεση στο internet γίνεται αφού λάβει χώρα πρώτα η αυθεντικοποιήση του η οποία γίνεται με τη χρήση ενός προσωπικού κωδικού που παραχωρείται στο κάθε φοιτητή. Ωστόσο αυτός ο τρόπος αυθεντικοποίησης μειονεκτεί από άποψη ασφάλειας καθώς είναι ιδιαίτερα ευπαθής σε απειλές μη εξουσιοδοτημένης πρόσβασης (π.χ. hacking). Αντίθετα η μέθοδος αυθεντικοποίησης με έξυπνες κάρτες είναι αρκετά αξιόπιστη αν λάβουμε υπόψη ένα από τα βασικά χαρακτηριστικά των έξυπνων καρτών που είναι η υλοποίηση αλγορίθμων κρυπτογράφησης κατά την ανταλλαγή των δεδομένων.

Η υπηρεσία αυτή παρουσιάζει αρκετές ομοιότητες με την διαδικασία αυθεντικοποίησης για την απόκτηση πρόσβασης στους υπολογιστές των εργαστηρίων. Αν η διαδικασία της αυθεντικοποίησης δείξει ότι ο χρήστης είναι εξουσιοδοτημένος τότε αυτός μπορεί να συνδεθεί με το internet. Τα συστατικά αυτού του συστήματος και τα βήματα του σεναρίου υλοποίησης απεικονίζονται παρακάτω.



6.4 Σενάριο σύνδεσης στο internet με έξυπνη κάρτα

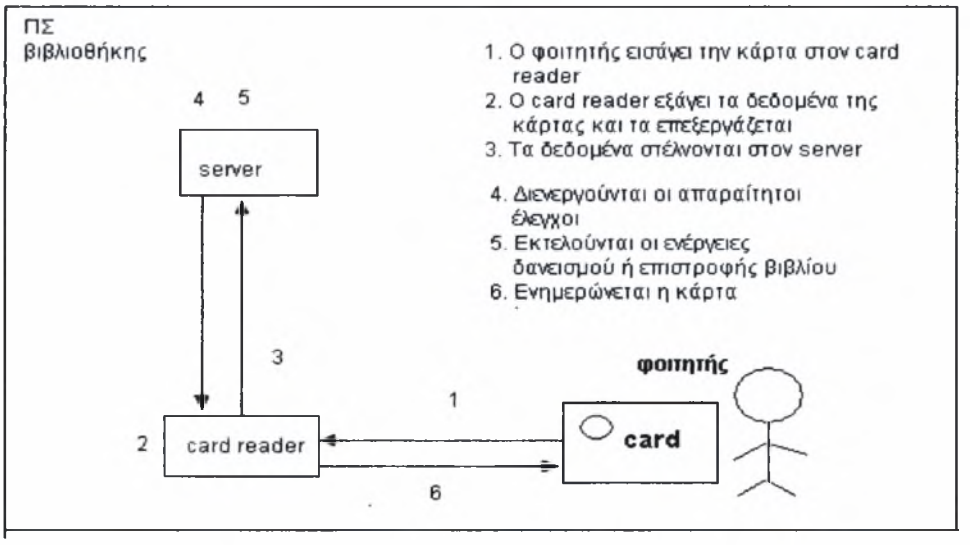
5. Πρόσβαση στις βιβλιοθήκες του πανεπιστημίου

Οι φοιτητές του τμήματος θα μπορούν να χρησιμοποιούν την έξυπνη κάρτα για το δανεισμό – επιστροφή βιβλίων. Στην περίπτωση αυτή βέβαια η κάρτα θα πρέπει να αποθηκεύει και κάποια επιπλέον δεδομένα όπως τον αριθμό των βιβλίων που έχει δανειστεί ο φοιτητής καθώς και τις ημερομηνίες δανεισμού και επιστροφής.

Τα βασικά συστατικά για την υλοποίηση ενός τέτοιου συστήματος δανεισμού και επιστροφής βιβλίων με χρήση έξυπνης κάρτας είναι τα ακόλουθα :

- **Έξυπνη κάρτα**: η οποία θα πρέπει να περιέχει εκτός των βασικών δεδομένων που αναφέραμε στην αρχή και κάποια επιπλέον στοιχεία όπως αριθμός βιβλίων που έχει δανειστεί ο φοιτητής και ημερομηνίες δανεισμού και επιστροφής αυτών.
- **Card Reader:** για την ανάγνωση της κάρτας
- **Server**: ο οποίος θα ευθύνεται για τους διάφορους ελέγχους που πρέπει να γίνονται, όπως για παράδειγμα αν ο συγκεκριμένος χρήστης είναι εγγεγραμμένος στο τμήμα, αν μπορεί να δανειστεί βιβλία, αν έχει λήξει η προθεσμία επιστροφής βιβλίων ώστε να επιβάλλεται πρόστιμο κ.λ.π.

Το σύστημα αυτό υλοποιείται στο ακόλουθο διάγραμμα:



6.5 Σενάριο υλοποίησης χρήσης της βιβλιοθήκης με έξυπνη κάρτα

Τα βήματα που χρειάζεται να γίνουν για την συγκεκριμένη υπηρεσία είναι τα ακόλουθα:

- Ο φοιτητής επιδεικνύει την κάρτα του στον υπεύθυνο της βιβλιοθήκης ο οποίος και την εισάγει στον αναγνώστη. Δεν είναι απαραίτητη η εισαγωγή κάποιου κωδικού από την πλευρά του φοιτητή.
- Ακολουθεί η επεξεργασία των δεδομένων τα οποία στέλνει ο card reader στο server και δίνει έτσι τη δυνατότητα στον υπεύθυνο να ελέγξει αν πρόκειται για εξουσιοδοτημένο χρήστη και να επεξεργαστεί τα δεδομένα της κάρτας με το κατάλληλο λογισμικό που διαθέτει.
- Έτσι μπορεί να βλέπει τον αριθμό των βιβλίων που έχει δανειστεί ο φοιτητής και αν αυτός υπερβαίνει ένα ανώτατο όριο η διαδικασία δανεισμού καθίσταται

μη επιτρεπτή. Αν ο φοιτητής μπορεί να δανειστεί θα πρέπει να ενημερώνονται τα πεδία της κάρτας δηλαδή ο αριθμός των δανειζομένων βιβλίων, οι ημερομηνίες δανεισμού και επιστροφής. Επιπλέον αν ο φοιτητής δεν έχει επιστρέψει το βιβλίο στην ημερομηνία επιστροφής όπως αυτή έχει οριστεί θα πρέπει να υποστεί τις συνέπειες όπως για παράδειγμα την επιβολή ενός προστίμου (όπως και ισχύει).

- Οι παραπάνω διαδικασίες ολοκληρώνονται με την ενημέρωση της κάρτας .

6.2.1 Χαρακτηριστικά του περιβάλλοντος εργασίας

Τα πληροφοριακά συστήματα που παρέχονται από το τμήμα και στα οποία έχουν πρόσβαση οι χρήστες του είναι τα ακόλουθα :

- ***e-class*** : Μέσω αυτής της εφαρμογής οι φοιτητές μπορούν να αποκτούν πρόσβαση στις πληροφορίες και υλικό του κάθε μαθήματος που παρακολουθούν, να αποστέλλουν τις εργασίες τους και να λαμβάνουν mails που σχετίζονται με το συγκεκριμένο μάθημα. Αρχικά θα πρέπει να εκτελέσουν της διαδικασία της εγγραφής τους στα μαθήματα που πρόκειται να παρακολουθήσουν, δηλαδή να διαμορφώσουν το προφίλ τους το οποίο απαιτεί όνομα, επώνυμο, username, password και e-mail address. Οι διδάσκοντες του τμήματος μπορούν να καταχωρούν υλικό και πληροφορίες σχετικές με το μάθημα όπως επίσης και τις βαθμολογίες των φοιτητών.

- ***NFS- Network File System (Για λειτουργικό σύστημα Linux)***: Πρόκειται για έναν ειδικό φλοιό συστήματος αρχείων μέσω του οποίου υποστηρίζεται η απομακρυσμένη πρόσβαση σε αρχεία που βρίσκονται σε άλλους υπολογιστές. Επιτρέπει την εξαγωγή ενός τοπικού συστήματος αρχείων για απομακρυσμένη χρήση από άλλους σταθμούς εργασίας. Αυτό γίνεται με την προσάρτηση του απομακρυσμένου καταλόγου σε έναν τοπικό κατάλογο αφού προσδιοριστεί η ακριβής διεύθυνση του απομακρυσμένου συστήματος αρχείων.

- ***Samba (Για λειτουργικό σύστημα Windows)***: Πρόκειται για την αντίστοιχη υπηρεσία NFS αλλά σε περιβάλλον Windows.

- ***CUPS-Common Unix Printer System***: Το σύστημα με το οποίο διαχειρίζονται οι ουρές εκτύπωσης για την ικανοποίηση των αιτήσεων εκτύπωσης των φοιτητών.

- ***LDAP***: Χρησιμοποιείται για την πρόσβαση των χρηστών σε υπηρεσίες καταλόγου

- ***Mailing lists***: Συνήθως για κάθε μάθημα υπάρχει η αντίστοιχη λίστα αλληλογραφίας στην οποία μπορεί να εγγραφεί ο κάθε ενδιαφερόμενος φοιτητής για να λαμβάνει τα mail που στέλνονται σε αυτή τη λίστα

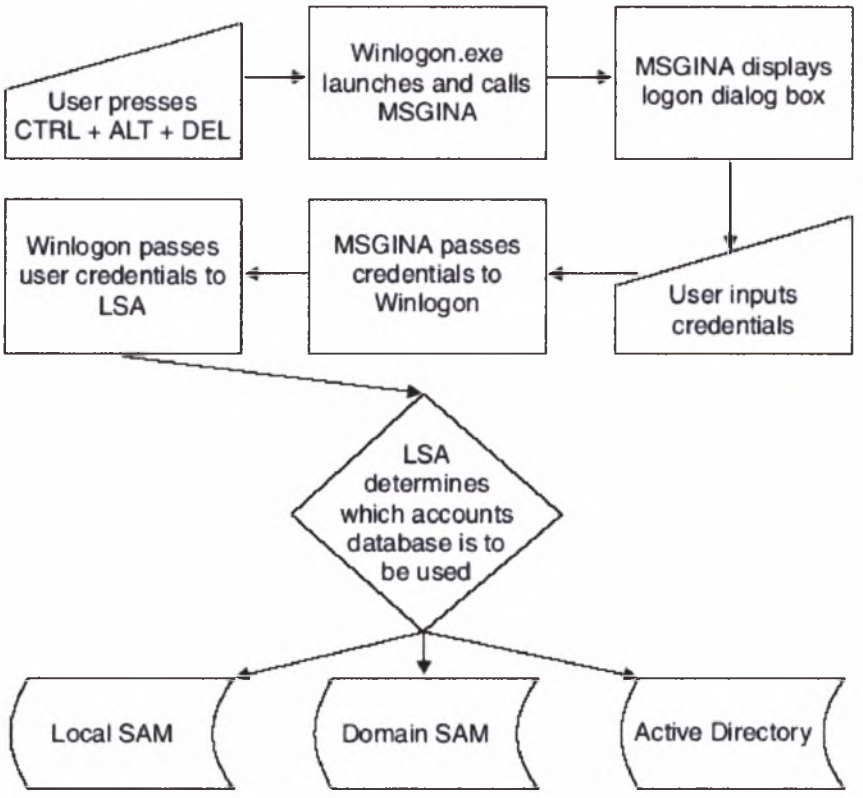
Επιπλέον κάποια άλλα πληροφοριακά συστήματα που χρησιμοποιούνται είναι ο ***Apache –Web Server,***η ***MySql, Php*** και ***Dhcp***

6.2.2 Μηχανισμός αυθεντικοποίησης στα Windows XP

Στην παράγραφο αυτή περιγράφουμε το μηχανισμό αυθεντικοποίησης που χρησιμοποιείται από το τμήμα για τον έλεγχο πρόσβασης των χρηστών στους υπολογιστές των εργαστηρίων.
Όλοι οι χρήστες, ομάδες χρηστών και υπολογιστές που συμμετέχουν έχουν έναν λογαριασμό (account) ο οποίος καλείται security principal. Τα δικαιώματα και οι άδειες πρόσβασης που αντιστοιχούν σε έναν λογαριασμό καθορίζονται από το λεγόμενο security context.
Τα windows χρησιμοποιούν το username και το password ως αποδεικτικά στοιχεία τα οποία πιστοποιούν την οντότητα ενός χρήστη. Η Local Security Authority (LSA) είναι υπεύθυνη για την εγκυρότητα των πιστοποιητικών που χρησιμοποιούνται στα Windows ενώ ο Security Accounts Manager (SAM) δηλαδή ο Windows 2000 server είναι υπεύθυνος για τη διαχείριση της βάσης δεδομένων των λογαριασμών των χρηστών.

Διαδικασία Logon

Υπάρχουν διάφοροι τύποι logon όπως : interactive, network, dial up authentication κ.α.. Στη συγκεκριμένη περίπτωση χρησιμοποιείται ο πρώτος τύπος όπου ο χρήστης προσπαθεί να αποκτήσει πρόσβαση σε έναν φυσικό σταθμό εργασίας στον οποίο και διαδραματίζεται η διαδικασία logon.
Η πιστοποίηση των αποδεικτικών στοιχείων που παρουσιάζει ο κάθε χρήστης δηλαδή το username και το password του γίνεται είτε από την τοπική βάση δεδομένων των λογαριασμών είτε από τον Windows 2000 server είτε από την Active Directory που στη συγκεκριμένη περίπτωση είναι ο LDAP.
Τα interactive logons χρησιμοποιούν ένα σύνολο από συστατικά για την μεταφορά των username και passwords, στην κατάλληλη βάση δεδομένων των λογαριασμών για την αυθεντικοποίηση. Το πρώτο από αυτά τα συστατικά είναι η Winlogon process, η οποία χρησιμοποιεί το εκτελέσιμο αρχείο Winlogon.exe. Αυτό εκτελείται όταν ο χρήστης πιέσει Control + Alt + Delete. Αφού ο χρήστης πιέσει το συνδυασμό αυτών των πλήκτρων η Winlogon καλεί το Microsoft Graphical Identification και το Authentication DLL (MSGINA) το οποίο συλλέγει το username και το password. Το συστατικό MSGINA παρέχει το Windows log-on παράθυρο διαλόγου. Μετά την εισαγωγή των username και password και αφού ο χρήστης πατήσει enter το MSGINA περνά αυτή την πληροφορία πίσω στο Winlogon το οποίο στη συνέχεια τα περνά στην Local Security Authority η οποία τρέχει ως LSAS.exe και η οποία θα αποφασίσει με ποιο τρόπο θα γίνει η αυθεντικοποίηση όπως φαίνεται στο διάγραμμα που ακολουθεί.



6.6 Συστατικά των interactive logons

Διαδικασία αυθεντικοποίησης

Η διαδικασία της αυθεντικοποίησης στα Windows XP χρησιμοποιεί δύο πρωτόκολλα τον Κέρβερο και τον Windows NT LAN Manager (NTLM). Ο κέρβερος είναι το default πρωτόκολλο που χρησιμοποιείται ενώ ο NTLM είναι η δεύτερη επιλογή των Windows XP σε περίπτωση που η αυθεντικοποίηση αποτύχει με τον πρώτο τρόπο.

Διαδικασία αυθεντικοποίησης με τον Κέρβερο

Ο κέρβερος είναι μια υπηρεσία αυθεντικοποίησης. Κάθε χρήστης που κάνει αίτηση σε έναν εξυπηρετητή για να αποκτήσει πρόσβαση σε μια υπηρεσία θα πρέπει πρώτα να αυθεντικοποιηθεί από τον εξυπηρετητή. Την εργασία αυτή αναλαμβάνει ο Authentication Server (AS) ο οποίος γνωρίζει τους κωδικούς όλων των χρηστών και τους έχει αποθηκευμένους σε μια βάση δεδομένων. Ο AS και ο εξυπηρετητής μοιράζονται ένα μοναδικό μυστικό κλειδί. Μια άλλη υπηρεσία η οποία καλείται Ticket Granting Server (TGS) αναλαμβάνει να χορηγεί tickets σε χρήστες που έχουν αυθεντικοποιηθεί από τον AS ώστε να τα χρησιμοποιούν για να αποκτήσουν πρόσβαση. Αυτά τα tickets είναι κρυπτογραφημένα με βάση το μυστικό κλειδί που γνωρίζουν ο εξυπηρετητής και ο AS.
Για την αυθεντικοποίηση λοιπόν ενός χρήστη με τον Κέρβερο θα πρέπει να χορηγηθούν σε αυτόν

- ένα Ticket Granting Ticket ώστε να αποκτήσει πρόσβαση στην υπηρεσία Ticket granting service
- και ένα ticket το οποίο θα του επιτρέπει την πρόσβαση στον σταθμό εργασίας στον οποίο έκανε log in.

Έτσι τα βήματα που ακολουθούνται έχουν ως εξής :

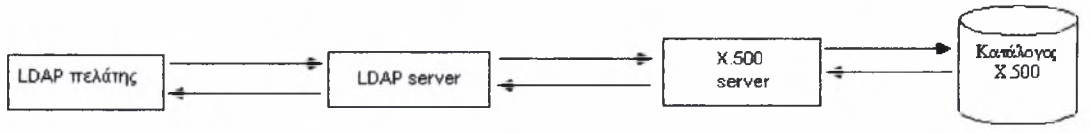
- Ο χρήστης πιέζει τα πλήκτρα Control + Alt + Delete, εμφανίζεται το παράθυρο διαλόγου όπου και πληκτρολογεί το username και το password και πατάει OK. Το MSGINA περνά αυτές τις πληροφορίες στο Winlogon το οποίο στη συνεχεία τα στέλνει στην LSA.
- Η LSA μετατρέπει τον κωδικό του χρήστη σε ένα preauthentication κρυπτογραφημένο κλειδί και το στέλνει στο Kerberos SSP AS η οποία στη συνέχεια επικοινωνεί με το Key Distribution Center.
- Στη συνέχεια ο Kerberos SSP στέλνει το preauthentication κρυπτογραφημένο κλειδί στην AS. Αυτή η αίτηση περιλαμβάνει τα ακόλουθα αντικείμενα :
    1. Το id του χρήστη και την υπηρεσία την οποία θέλει να προσπελάσει που στη συγκεκριμένη περίπτωση είναι η υπηρεσία αυθεντικοποίησης.
    2. Το κρυπτογραφημένο password και ένα time stamp για τον χρήστη
- Ως απάντηση της παραπάνω αίτησης το KDC στέλνει στον Kerberos SSP το ticket granting ticket για να το χρησιμοποιήσει ο χρήστης. Τα δεδομένα που στέλνονται είναι :
    1. Το session key για τον χρήστη
    2. Το ticket granting ticket για τον KDC το οποίο είναι κρυπτογραφημένο με το μυστικό κλειδί του τελευταίου.
- Ο Kerberos SSP στέλνει τα ακόλουθα δεδομένα ως απάντηση στον KDC :
    1. Το όνομα του σταθμού εργασίας στο οποίο ο χρήστης έκανε login και το domain στο οποίο ανήκει ο συγκεκριμένος σταθμός εργασίας.
    2. Το ticket granting ticket το οποίο κρυπτογραφείται με το session key του χρήστη.
- Ο KDC απαντά στέλνοντας τα ακόλουθα στον Kerberos SSP:
    1. Ένα session key κρυπτογραφημένο με το μυστικό κλειδί του χρήστη το οποίο θα χρησιμοποιείται από τον χρήστη και τον σταθμό εργασίας για την μεταξύ τους επικοινωνία.
    2. Ένα session ticket κρυπτογραφημένο με το μυστικό κλειδί του σταθμού εργασίας για την προσπέλαση του.
- Τέλος ο Kerberos SSP περνά τις παραπάνω πληροφορίες στον LSA ο οποίος επικοινωνεί με τον SAM (Windows 2000 server) για να καθοριστούν τα δικαιώματα του χρήστη για τον συγκεκριμένο σταθμό εργασίας. Το αποτέλεσμα αυτής της επικοινωνίας είναι η δημιουργία ενός token προσπέλασης το οποίο ο LSA στέλνει στο Winlogon οπότε ολοκληρώνεται η διαδικασία logon με την εμφάνιση του Windows XP desktop.

Διαδικασία αυθεντικοποίησης με το Windows NT LAN Manager

Ο δεύτερος τρόπος αυθεντικοποίησης γίνεται με τη χρήση του πρωτοκόλλου Windows NT LAN Manager και η μέθοδος αυθεντικοποίησης που χρησιμοποιείται είναι η μέθοδος πρόκλησης – απόκρισης (challenge – response).

LDAP

Το πρωτόκολλο LDAP (Lightweight Directory Access Protocol) είναι ένας μηχανισμός επικοινωνίας ο οποίος παρέχει στα προγράμματα πελάτη που το χρησιμοποιούν τη δυνατότητα πρόσβασης σε υπηρεσίες κατάλογου που ακολουθούν το πρότυπο X.500 με τη χρήση ενός προγράμματος εξυπηρέτη LDAP. Το γενικό μοντέλο στο οποίο στηρίζεται η λειτουργία του πρωτοκόλλου είναι το γνωστό μοντέλο πελάτη - εξυπηρέτη. Αυτό απεικονίζεται αμέσως παρακάτω.

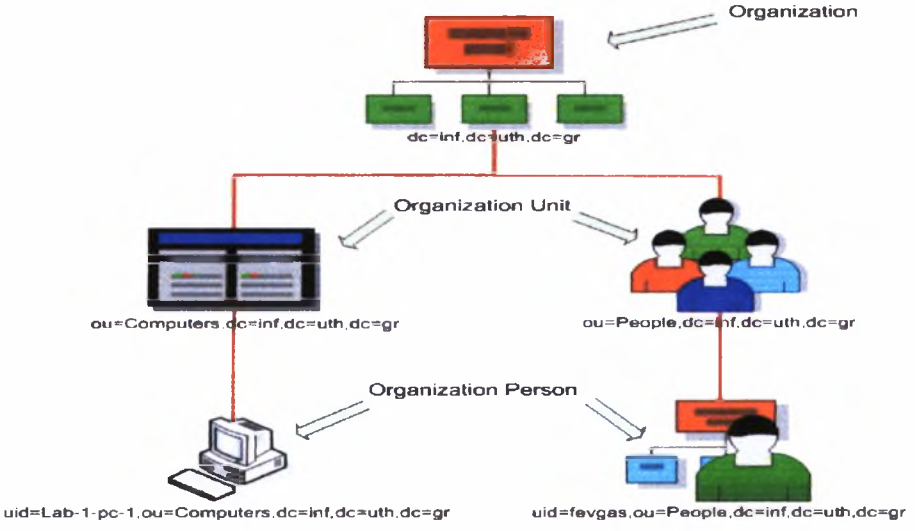


6.7 Γενικό μοντέλο στο οποίο στηρίζεται η λειτουργία του LDAP

Σύμφωνα με το μοντέλο αυτό, το πρόγραμμα πελάτη μεταδίδει μια αίτηση για να λάβει κάποιας μορφής εξυπηρέτηση από ένα πρόγραμμα εξυπηρέτη, ο οποίος είναι υπεύθυνος να εκτελέσει τις λειτουργίες που περιγράφονται στην αίτηση του πελάτη και να του επιστρέψει κάποια αποτελέσματα ή απάντηση.

Το LDAP στηρίζει τη λειτουργία του μοντέλου καταλόγου που χρησιμοποιεί στην αναγνώριση μοναδικών εγγραφών με βάση το μοναδικό διαχωριστικό όνομα τους (globally-unique Distinguished NameDN). Κάθε εγγραφή περιλαμβάνει μια συλλογή από χαρακτηριστικά τα οποία είναι ενός συγκεκριμένου τύπου και έχουν μία ή περισσότερες τιμές. Για τον τύπο των χαρακτηριστικών χρησιμοποιούνται μνημονικά αλφαριθμητικά όπως για παράδειγμα το 'cn'(common name) για ονόματα ή 'mail' για την e-mail address. Οι τιμές που παίρνει το κάθε χαρακτηριστικό εξαρτάται από τον τύπο του. Έτσι για το χαρακτηριστικό cn η τιμή θα μπορούσε να είναι ένα ονοματεπώνυμο (για π.χ. Ζιάκα Ευαγγελία) ενώ για το χαρακτηριστικό mail η τιμή θα μπορούσε να είναι μια e-mail address (eziaka@uth.gr).

Στο LDAP οι εγγραφές καταλόγου οργανώνονται σε μια δομή ιεραρχικού δένδρου όπως φαίνεται για παράδειγμα παρακάτω



6.8 LDAP tree

Για τον έλεγχο των χαρακτηριστικών που απαιτούνται και επιτρέπεται να χρησιμοποιηθούν σε μια εγγραφή χρησιμοποιείται το ειδικό χαρακτηριστικό που καλείται object class. Οι τιμές που παίρνει αυτό το χαρακτηριστικό καθορίζουν τους κανόνες του 'schema' στους οποίους πρέπει να υπακούει η κάθε εγγραφή. Με την έννοια schema εννοούμε κάποια αρχεία (schema files) τα οποία ορίζουν τον τρόπο με τον οποίο είναι οργανωμένες οι πληροφορίες . Έτσι για παράδειγμα υπάρχουν τα εξής schema files : core.schema, cosine.schema, inetorgperson.schema, misc.schema, nis.schema, openldap.schema κ.α.. Ανάλογα με την χρήση της εγγραφής αυτά τα αρχεία θα πρέπει να εγκατασταθούν στην διεύθυνση /usr/local/etc/openldap/schema. Συγκεκριμένα από το τμήμα χρησιμοποιούνται για τον μηχανισμό της αυθεντικοποίησης, τα ακόλουθα schema files:

- ***nis.schema*** για λειτουργικό σύστημα Linux
- ***samba.schema*** για λειτουργικό σύστημα Windows

Οι πληροφορίες των χρηστών μπορούν να αναπαρασταθούν σε text μορφή με το LDIF (LDAP Data Interchange Format). Ένα παράδειγμα αυτής της αναπαράστασης φαίνεται παρακάτω:

```
dn:
uid=****,ou=People,dc=inf,dc=uth,dc=gr
gecos: *****,D5-11,4965
gidNumber:
homeDirectory:
loginShell:
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
objectClass: sambaAccount
uidNumber:
mail: ****@inf.uth.gr
mobile::
facsimileTelephoneNumber::
postalCode: .....
userPassword::
shadowLastChange:
uid: *****
pwdLastSet:
```

6.9 LDIF αναπαράσταση

Η βιβλιοθήκη Pluggable Authentication Modules library (PAM_LDAP) είναι ένα API το οποίο παρέχει υπηρεσίες αυθεντικοποίησης και επιτρέπει στον administrator χρησιμοποιώντας τα PAM configuration files να γράψει μια λίστα απαιτήσεων που πρέπει να ικανοποιεί ένας χρήστης ώστε να προσπελάσει κάποιον πόρο. Μια τέτοια λίστα μπορεί να έχει την ακόλουθη μορφή

```
/etc/pam.d/system_auth
auth        sufficient    /lib/security/pam_ldap.so use_first_pass
auth        sufficient    /lib/security/pam_unix.so #use_first_pass
auth        required      /lib/security/pam_deny.so

account     required      /lib/security/pam_unix.so
account     sufficient    /lib/security/pam_ldap.so

password    required      /lib/security/pam_cracklib.so retry=3
password    sufficient    /lib/security/pam_unix.so nullok md5 shadow use_authtok
password    sufficient    /lib/security/pam_ldap.so use_authtok
password    required      /lib/security/pam_deny.so

session     required      /lib/security/pam_limits.so
session     required      /lib/security/pam_unix.so
session     required      /lib/security/pam_mkhomedir.so skel=/etc/skel/ umask=0
session     optional      /lib/security/pam_ldap.so
```

6.10 Λίστα απαιτήσεων

Τέλος η υπηρεσία Name Service Switch παρέχει στο λειτουργικό σύστημα όλες τις πληροφορίες για έναν χρήστη, τα groups όπου ανήκει κ.α. ενώ με το configuration file /etc/nsswitch.conf ορίζεται που βρίσκονται οι πληροφορίες και η σειρά με την οποία θα γίνει ο έλεγχος για την ανάκτηση αυτών. Έτσι για παράδειγμα αν έχουμε το ακόλουθο αρχείο

```
/etc/nsswitch.conf
passwd:      files ldap
shadow:      files ldap
group:       files ldap
```

σημαίνει ότι ο έλεγχος θα πρέπει πρώτα να γίνει στα files και μετά στον LDAP.

6.3 Πειραματική Υλοποίηση Αυθεντικοποίησης με χρήση έξυπνης κάρτας

6.3.1 Αρχικοποίηση - ενεργοποίηση έξυπνης κάρτας φοιτητή

Το λογισμικό που χρησιμοποιήθηκε για την υλοποίηση είναι το CHIPDRIVE Smart Card Office της SCM Microsystems Inc.

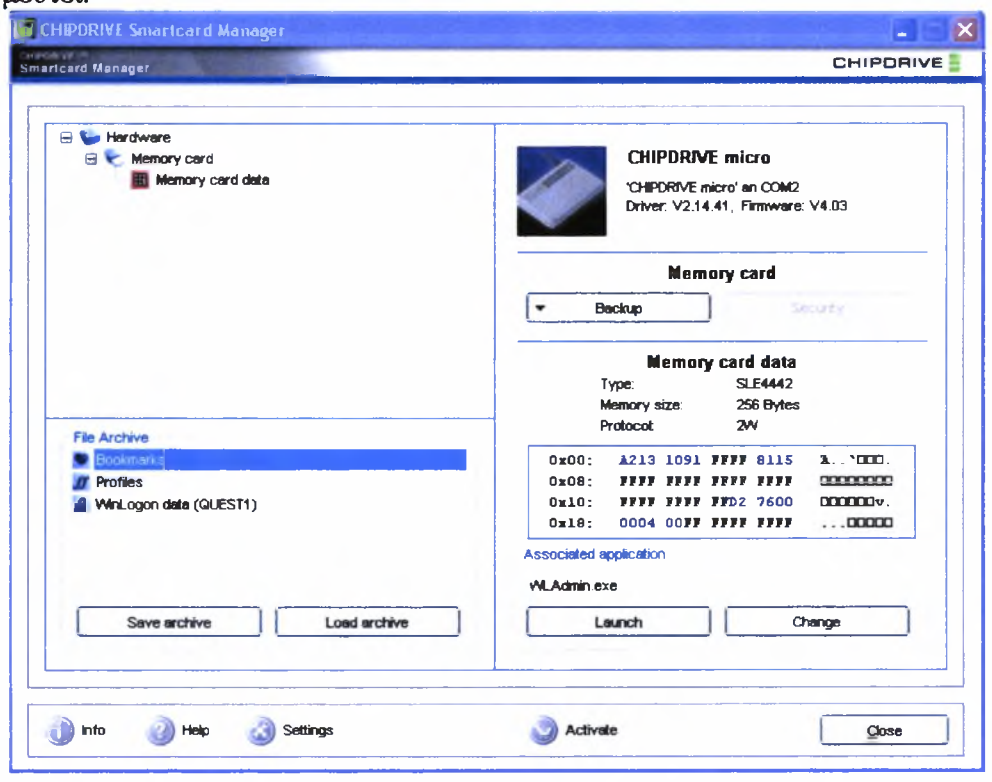
6.11 Smart Card Office

Τα εργαλεία που περιλαμβάνει και χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι τα ακόλουθα :

6.3.1.1. CHIPDRIVE Smart Card Manager

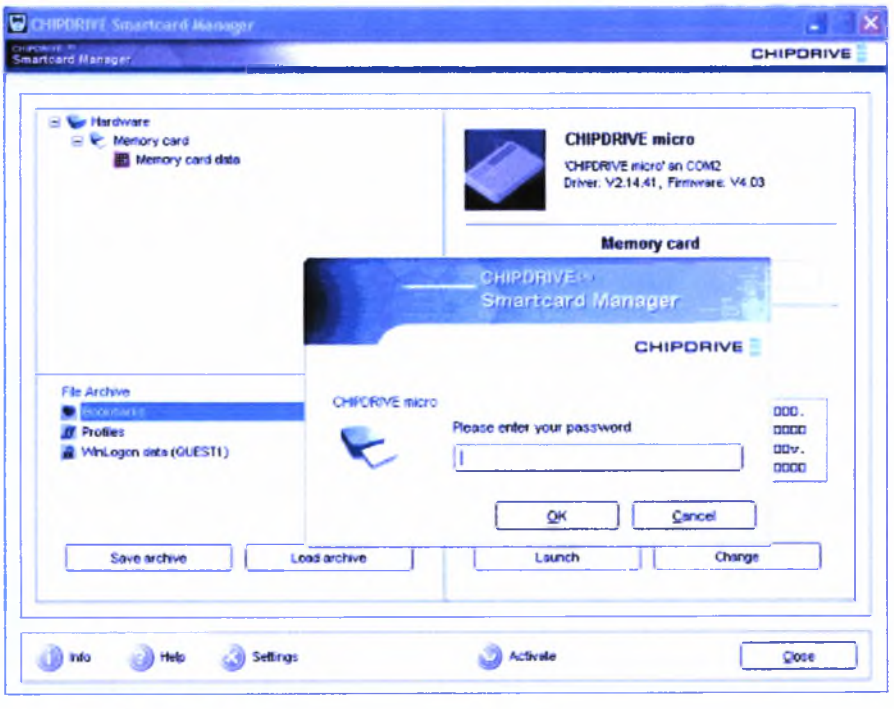
Είναι το εργαλείο που παρέχει τις σημαντικότερες πληροφορίες σχετικά με το υλικό που χρησιμοποιείται (hardware-card reader) και τις έξυπνες κάρτες. Παρέχει τις λειτουργίες για τη διαχείριση των κωδικών που αποθηκεύονται στην κάρτα όπως για παράδειγμα με την επιλογή *Backup* δημιουργούνται αντίγραφα των δεδομένων της κάρτας ενώ με την επιλογή *Security* ο κωδικός της κάρτας και αφού καθοριστεί για πρώτη φορά στη συνέχεια δικαίωμα μεταβολής των ρυθμίσεων ασφάλειας ή αλλαγής του κωδικού έχει μόνο ο κάτοχος της κάρτας και γνώστης του κωδικού της.
Έτσι η παρακάτω εικόνα δείχνει τον card reader που χρησιμοποιείται, τα περιεχόμενα της κάρτας που είναι τα Bookmarks, τα Profiles, το Winlogon Data (GUEST1) το οποίο είναι ο αποθηκευμένος κωδικός του χρήστη GUEST1 και τέλος

η συνολική χωρητικότητα της κάρτας και το ποσοστό του χώρου που έχει ήδη δεσμευτεί.



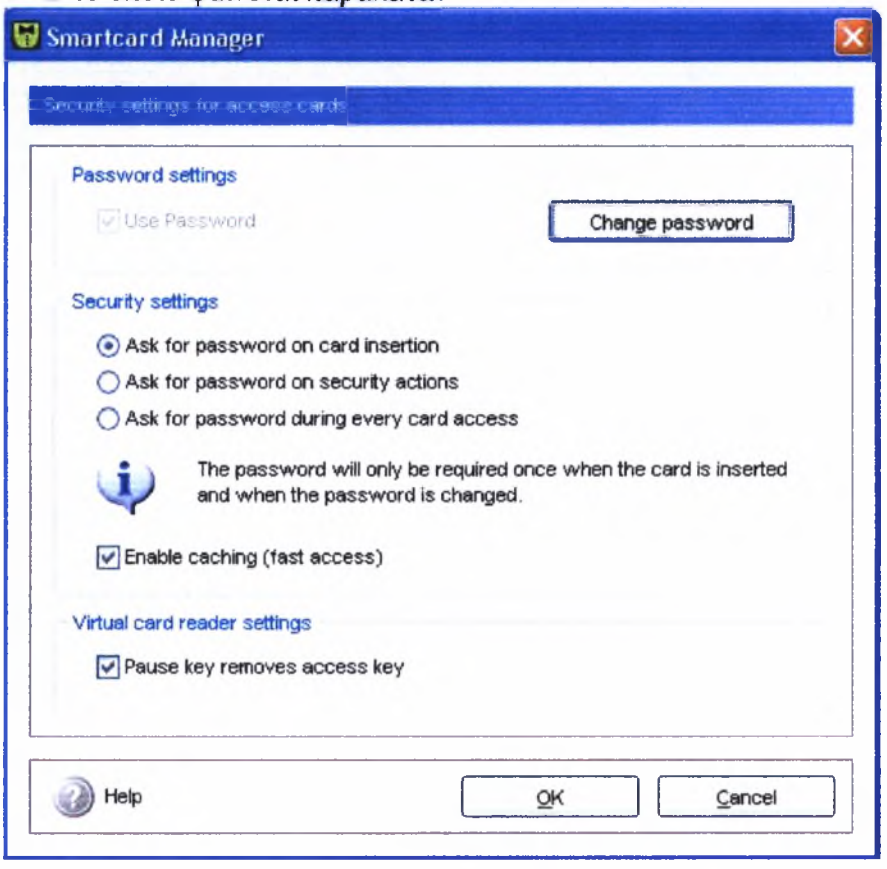
6.12 CHIPDRIVE Smart Card Manager

Όπως αναφέραμε παραπάνω, για να γίνουν οποιεσδήποτε ρυθμίσεις ασφάλειας ο χρήστης πρέπει να επιλέξει την επιλογή *Security*. Τότε εμφανίζεται το παράθυρο διαλόγου στο οποίο ο χρήστης πρέπει να δώσει τον κωδικό της κάρτας.



6.13 Smart Cart Manager Security Settings

Αφού γίνει η εισαγωγή του κωδικού εμφανίζεται το παράθυρο *Security settings for access cards* το οποίο φαίνεται παρακάτω.

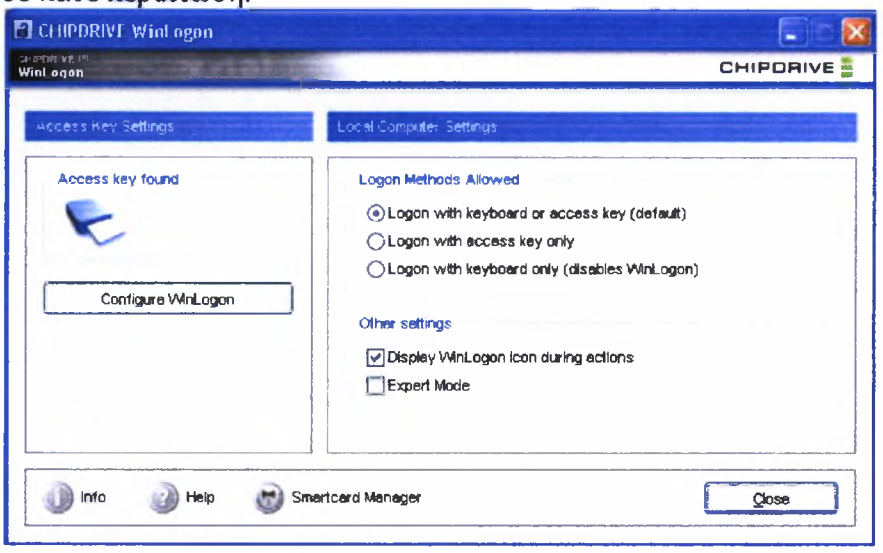


6.14 Password and Security Settings

6.3.1.2. CHIPDRIVE WinLogon

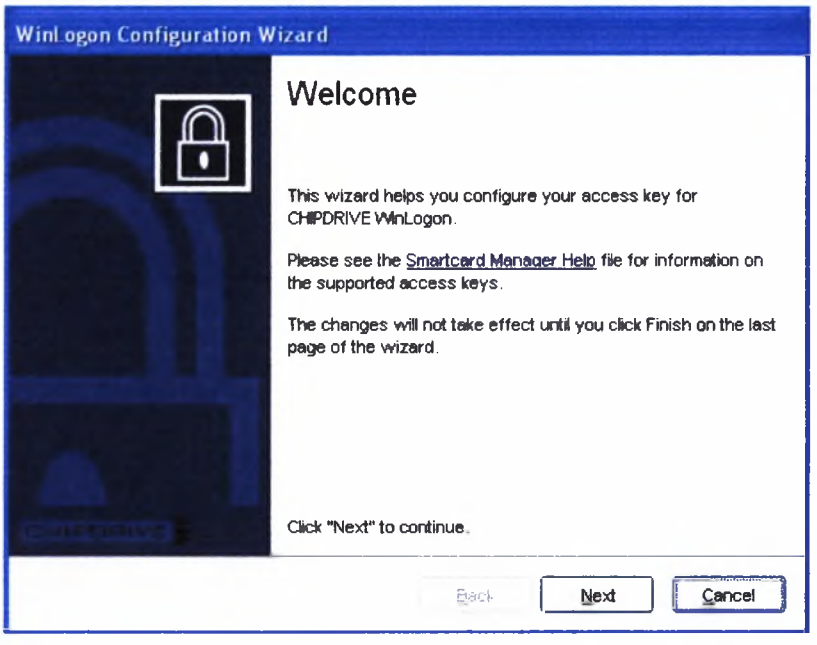
Είναι το εργαλείο που επιτρέπει την πρόσβαση σε ένα υπολογιστή με τη χρήση μιας έξυπνης κάρτας. Η κάρτα περιέχει όλη την απαραίτητη πληροφορία για την διαδικασία logon όπως δηλαδή το Windows logon name (username) τον κωδικό πρόσβασης (password) και το domain εάν αυτό κρίνεται απαραίτητο. Όλη αυτή η πληροφορία αποθηκεύεται φυσικά σε κρυπτογραφημένη μορφή.
Τα βήματα υλοποίησης της διαδικασίας logon στα Windows φαίνονται στις παρακάτω εικόνες.
Αρχικά θα πρέπει να γίνει η διαμόρφωση της κάρτας, το οποίο λαμβάνει χώρα την πρώτη φορά που εισάγεται η κάρτα στον αναγνώστη και αναγνωρίζεται από αυτόν.
Στη συνέχεια επιλέγεται *Configure WinLogon* το οποίο εμφανίζει το *Winlogon Configuration Wizard* με το οποίο καθορίζεται με ποιο τρόπο θα αυθεντικοποιεί η κάρτα τον χρήστη. Αν τσεκάρουμε την επιλογή Expert mode (του εργαλείου CHIPDRIVE WinLogon) τότε έχουμε την δυνατότητα να καθορίσουμε πως θα αντιδρά ο υπολογιστής στην περίπτωση εισαγωγής ή εξαγωγής της κάρτας όπως για

παράδειγμα αν θα κλειδώνει ή θα ξεκλειδώνει τον υπολογιστή ή αν θα ρωτά το χρήστη σε κάθε περίπτωση.

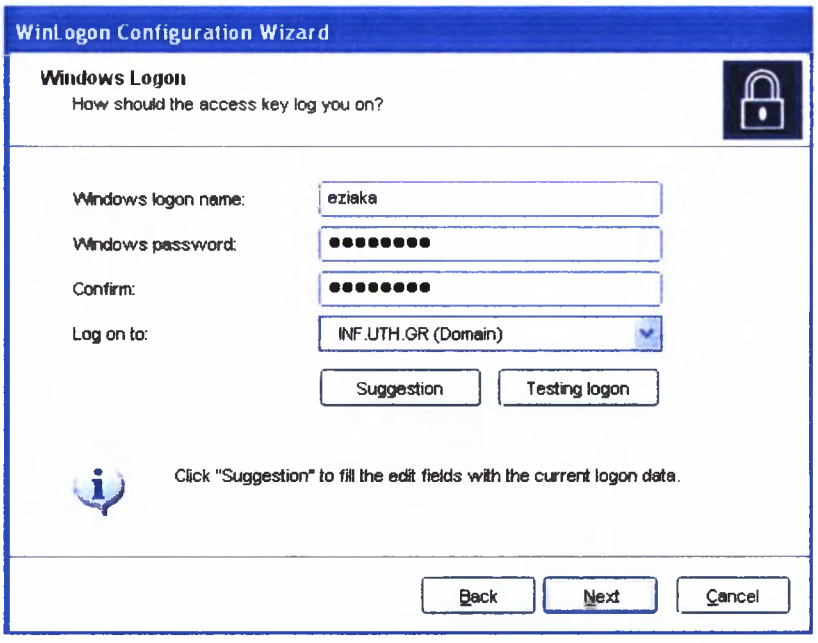


6.15 Διαδικασία WinLogon Βήμα 1

Όπως βλέπουμε παραπάνω έχουμε τσεκάρει την πρώτη επιλογή ώστε ένας χρηστής να μπορεί να κάνει logon είτε με χρήση κάρτας είτε με το πληκτρολόγιο.

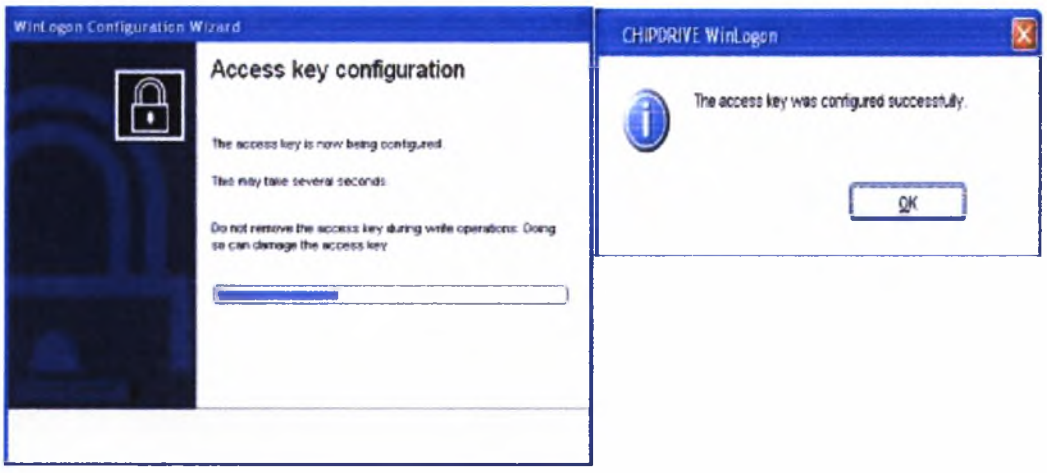


6.16 Εκκίνηση του Winlogon Configuration Wizard



6.17 Αποθήκευση των username, password και domain

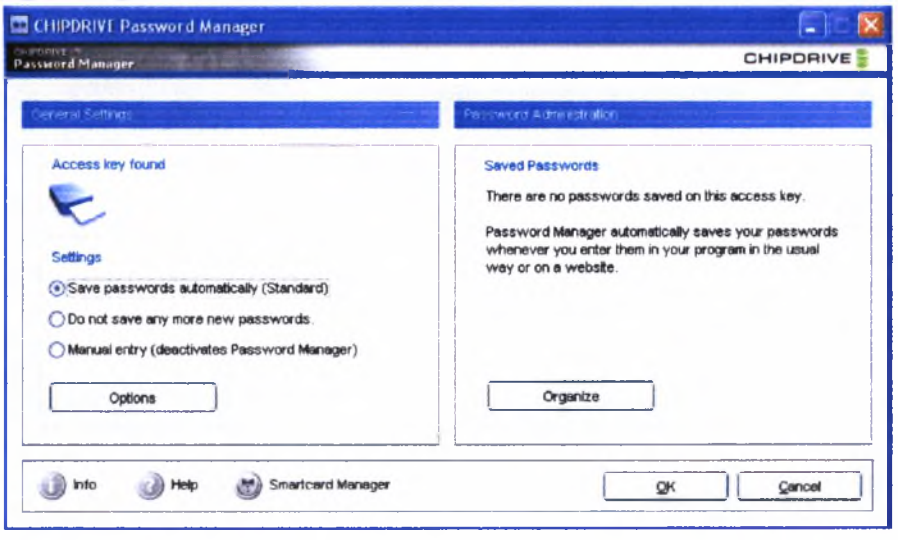
6.18 Η αποθήκευση των στοιχείων σε εξέλιξη



6.19 Ολοκλήρωση Διαδικασίας

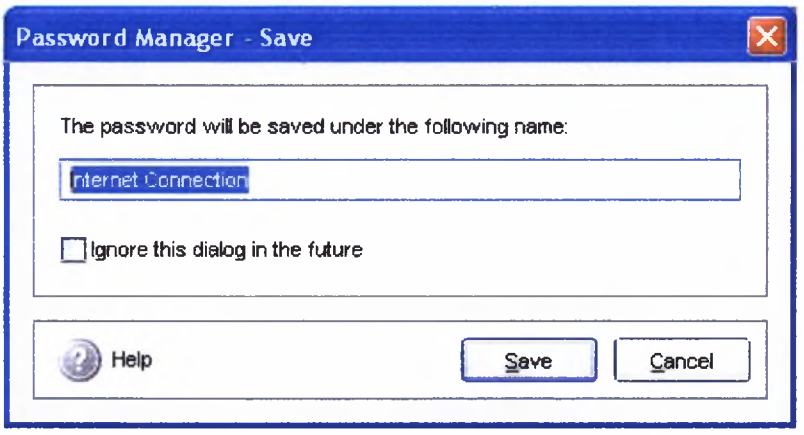
6.3.1.3. CHIPDRIVE Password Manager

Είναι το εργαλείο το οποίο αποθηκεύει κωδικούς σε κρυπτογραφημένη μορφή και τους εισάγει αυτόματα όταν χρησιμοποιείται η κάρτα. Για να γίνει αυτό θα πρέπει να έχει τσεκαριστεί η επιλογή *Save passwords automatically* όπως φαίνεται στην παρακάτω εικόνα:



6.20 Password Manager

Έτσι εάν ο χρήστης πληκτρολογήσει κάποιο κωδικό για παράδειγμα στην περίπτωση που θέλει να συνδεθεί στο internet θα εμφανιστεί το ακόλουθο παράθυρο διαλόγου

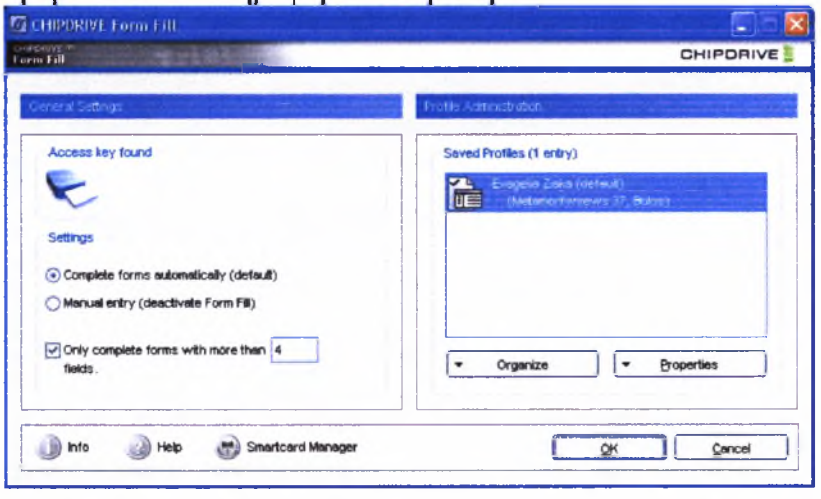


Εικόνα 6.21

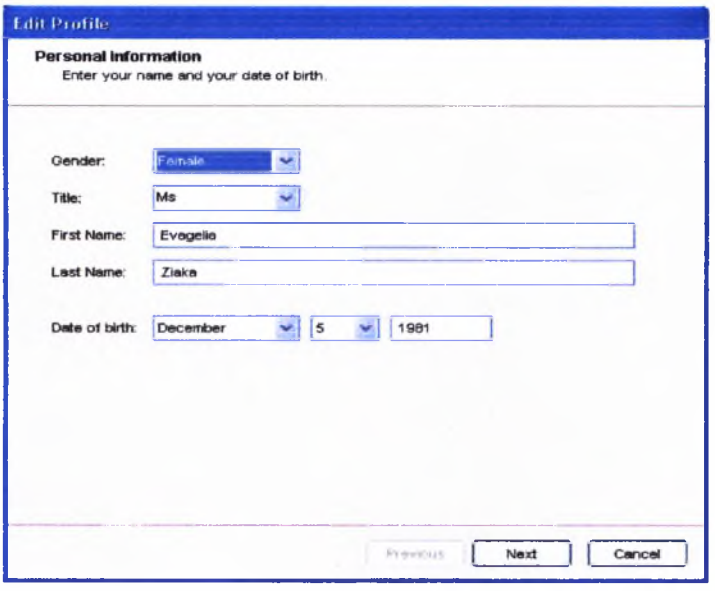
το οποίο ζητά από το χρήστη ένα όνομα για το συγκεκριμένο password και κατόπιν το αποθηκεύει. Την επόμενη φορά η σύνδεση στο internet θα γίνεται με την κάρτα.

6.3.1.4. CHIPDRIVE Form Fill

Παρέχει έτοιμες φόρμες για την δημιουργία του profile των κατόχων και γενικά για αποθήκευση προσωπικών πληροφοριών στην κάρτα.



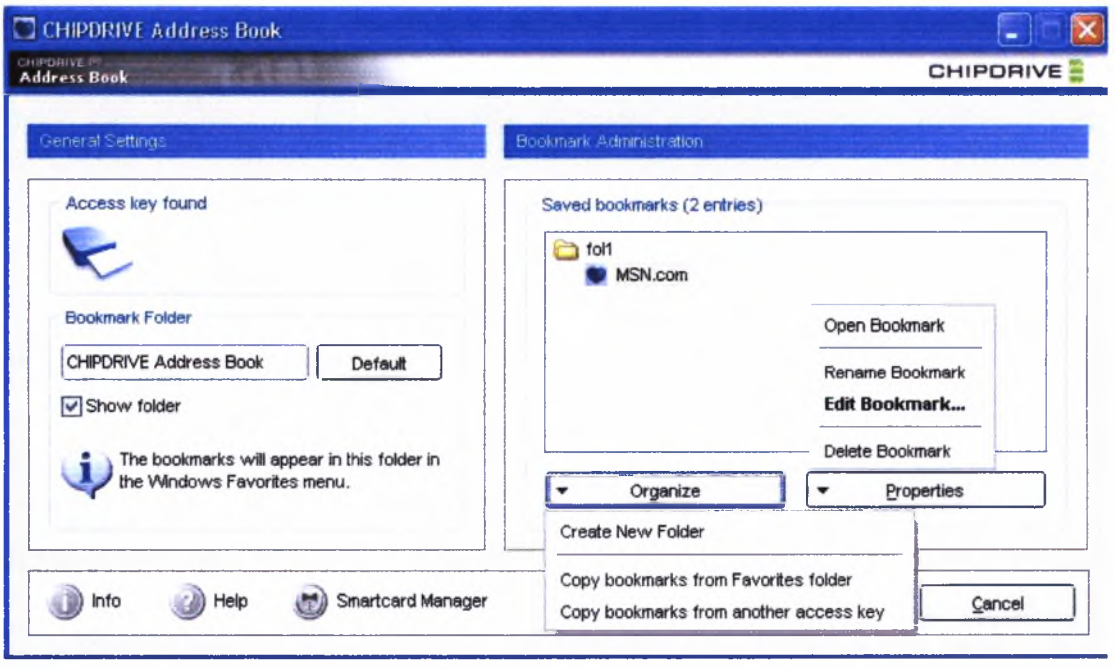
6.22 CHIPDRIVE Form Fill



6.23 Edit Profile

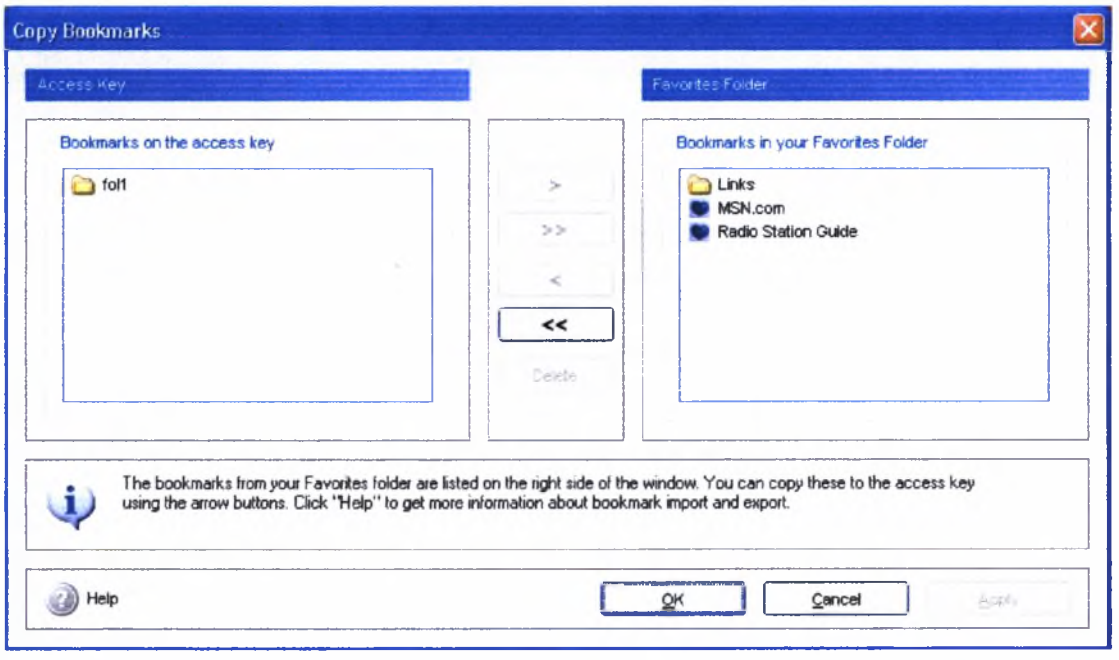
6.3.1.5. CHIPDRIVE Address Book

Το πρόγραμμα αυτό χρησιμοποιείται για την κρυπτογράφηση και αποθήκευση ιστοσελίδων. Οι δυνατότητες που παρέχει φαίνονται στην παρακάτω εικόνα. Ο χρήστης μπορεί να αποθηκεύσει τις ιστοσελίδες και να τις οργανώσει σε φακέλους ενώ επίσης παρέχονται και οι ενέργειες *open, rename, edit* και *delete* μιας ιστοσελίδας. Οι αποθηκευμένες ιστοσελίδες θα εμφανίζονται στον web browser που χρησιμοποιεί ο χρήστης στο menu Favorites στον φάκελο CHIPDRIVE Address Book



6.24 CHIPDRIVE Address Book

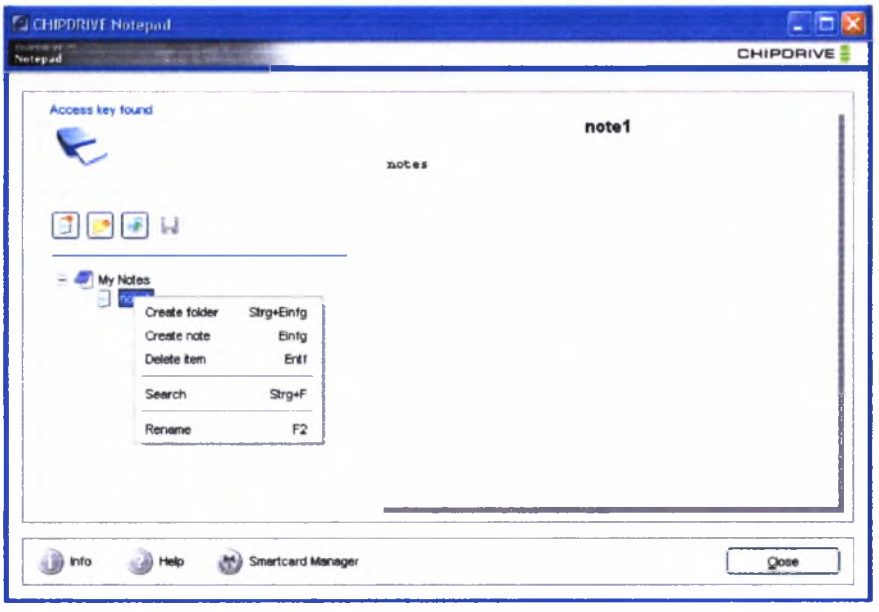
Με την επιλογή *Copy bookmarks from Favorites folder* μπορούμε να αντιγράψουμε τις σελίδες που είναι ήδη αποθηκευμένες στον φάκελο Favorites στην κάρτα. Η διαδικασία αυτή φαίνεται στην ακόλουθη εικόνα.



6.25 Copy Bookmarks

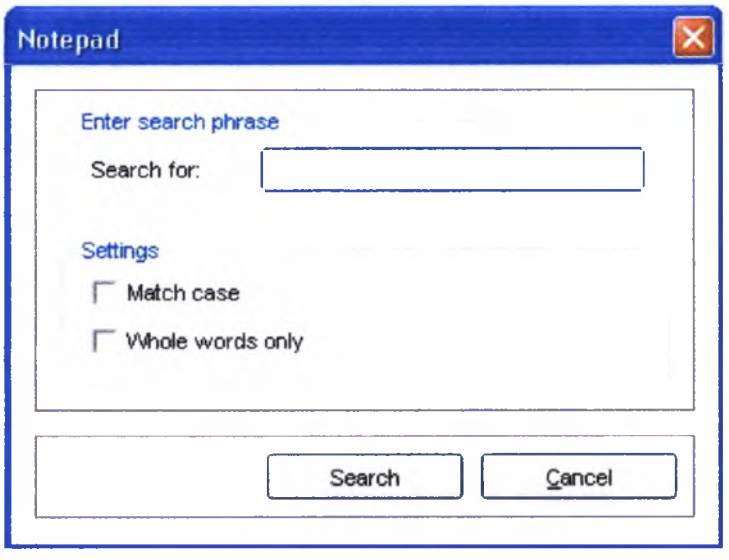
6.3.1.6. CHIPDRIVE Notepad

Το εργαλείο CHIPDRIVE Notepad δίνει τη δυνατότητα στο χρήστη να δημιουργεί τις σημειώσεις του και να τις αποθηκεύει στην έξυπνη κάρτα σε κρυπτογραφημένη μορφή ώστε να προστατεύονται από κάθε μη εξουσιοδοτημένη πρόσβαση.
Το κυρίως παράθυρο χωρίζεται σε δύο επιμέρους περιοχές . Στην αριστερή περιοχή εμφανίζεται ο φάκελος όπου θα περιέχονται οι σημειώσεις καθώς και τα εργαλεία για τη δημιουργία νέων φακέλων και σημειώσεων, τη διαγραφή και την αποθήκευση τους. Στην δεξιά περιοχή εμφανίζονται τα περιεχόμενα του τρέχοντος αρχείου που έχει επιλεχθεί και το οποίο μπορεί να τροποποιηθεί οποιαδήποτε στιγμή.



6.26 CHIPDRIVE Notepad

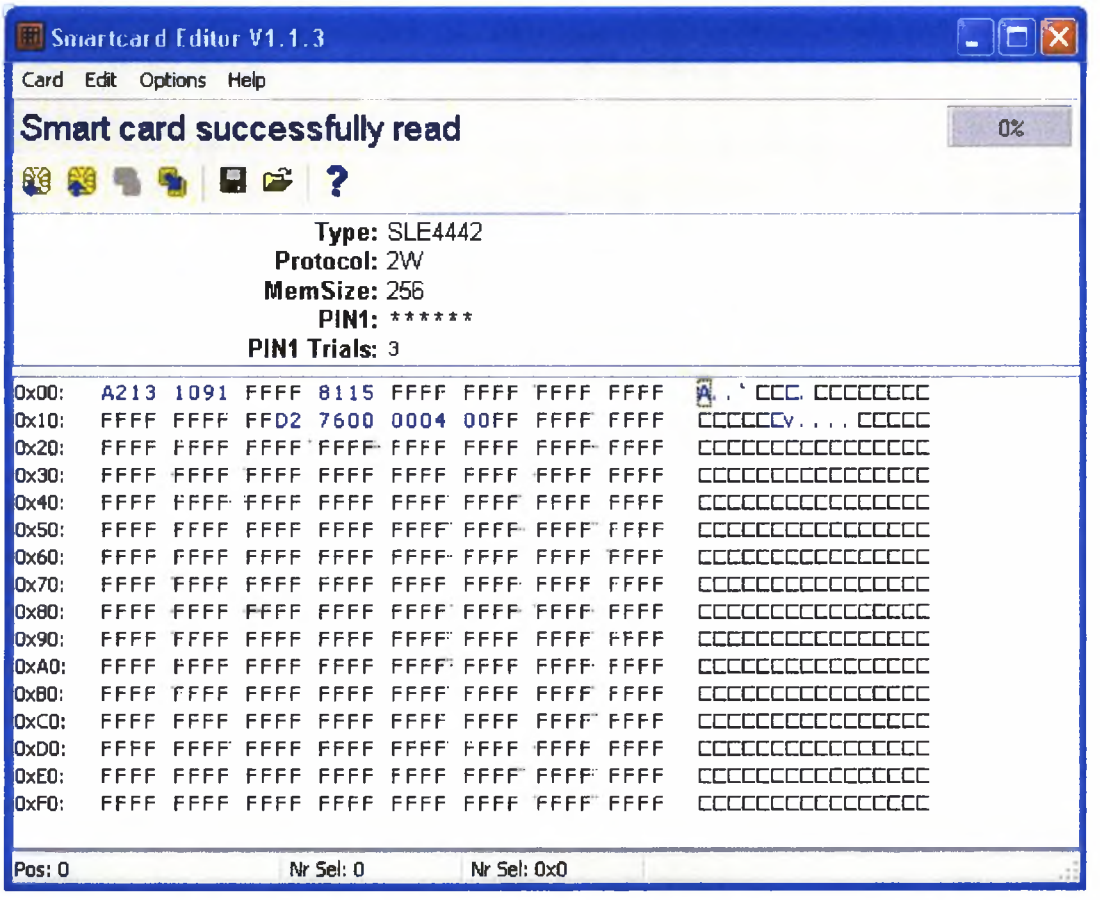
Τέλος υπάρχει η δυνατότητα αναζήτησης σε έναν συγκεκριμένο φάκελο ή στο φάκελο Notes. Το παράθυρο διαλόγου αναζήτησης, το οποίο φαίνεται αμέσως παρακάτω, εμφανίζεται με το συνδυασμό πλήκτρων CTRL+F.



6.27 Παράθυρο διαλόγου αναζήτησης

6.3.1.7. CHIPDRIVE Smart Card Editor

Είναι το εργαλείο το οποίο δείχνει τα περιεχόμενα της κάρτας, διαβάζει δηλαδή την κάρτα με την επιλογή read αλλά επίσης μπορεί να γράψει δεδομένα στην κάρτα εφόσον αυτό επιτρέπεται.



6.28 Smart Card Editor

6.3.2 Διαδικασία Αυθεντικοποίησης

Σύμφωνα με τον παραπάνω τρόπο αυθεντικοποίησης με έξυπνες κάρτες ο χρήστης μπορεί να κάνει login στον προσωπικό του υπολογιστή πληκτρολογώντας τον κωδικό του στο παράθυρο διαλόγου που εμφανίζεται στην οθόνη του.

Ο κάθε κωδικός συνδέεται με ένα συγκεκριμένο λογαριασμό χρήστη, σύμφωνα με τις ρυθμίσεις που έχουν γίνει με το εργαλείο CHIPDRIVE WinLogon που περιγράψαμε παραπάνω. Εάν ο κωδικός είναι σωστός η διαδικασία login έχει ολοκληρωθεί και εμφανίζεται το Windows XP Desktop με το profile του συγκεκριμένου χρήστη.

Κεφάλαιο 7

Συμπεράσματα και Μελλοντικές Επεκτάσεις

Σε προηγούμενα κεφάλαια είδαμε ότι η χρήση των νέων τεχνολογιών και συστημάτων ηλεκτρονικών συναλλαγών επιβάλουν την συνεχή μεταφορά και ανταλλαγή πληροφοριών οπουδήποτε και οποτεδήποτε. Γεγονός που εισάγει ζητήματα προστασίας των προσωπικών μας δεδομένων και ελέγχου στο ποιος αποκτά πρόσβαση σε αυτά. Επίσης είδαμε ότι τα συστήματα αυθεντικοποίησης με έξυπνες κάρτες παντρεύουν την αξιοπιστία, την ασφάλεια την ευκολία στη χρήση και την πολύ-εφαρμογικότητα. Μια έξυπνη κάρτα δεν έχει απλά τη δυνατότητα αποθήκευσης και επεξεργασίας των δεδομένων της αλλά μπορεί να περιλαμβάνει και εφαρμογές όπως συμβαίνει με τις java cards. Η ανάπτυξη αυτών των εφαρμογών μπορεί να γίνει με τη χρήση της Java Card Technology. Αντικείμενο αυτού του κεφαλαίου είναι ο σχεδιασμός ενός Java Card Applet το οποίο δίνει τη δυνατότητα διαχείρισης έξυπνων φοιτητικών καρτών, παρέχοντας λειτουργίες όπως αποθήκευση προσωπικών στοιχείων όπως όνομα του φοιτητή, το id του και ανάκτησης αυτών των δεδομένων. Η ανάπτυξη εφαρμογών με χρήση της Java Card Technology και του Java Card Framework αποτελεί την καλύτερη επιλογή δεδομένου της πληθώρας των πλεονεκτημάτων που αυτή παρέχει.

7.1 Πλεονεκτήματα Java Card Technology

Εύκολη ανάπτυξη εφαρμογών

Η ανάπτυξη των εφαρμογών γίνεται με χρήση της υψηλού επιπέδου γλώσσας προγραμματισμού java. Αυτό διευκολύνει το έργο των προγραμματιστών οι οποίοι δεν χρειάζεται να προγραμματίσουν σε κάποια γλώσσα χαμηλού επιπέδου όπως είναι η assembly. Η πολυπλοκότητα και οι λεπτομέρειες του συστήματος των έξυπνων καρτών αποκρύπτονται από τη java card πλατφόρμα η οποία παρέχει όλες τις απαραίτητες κλάσεις, πακέτα και διεπαφές που μπορούν να χρησιμοποιηθούν για τη διαχείριση των λειτουργιών των έξυπνων καρτών (ανάγνωση, εγγραφή, ανταλλαγή δεδομένων, επικοινωνία αναγνώστη έξυπνης κάρτας και υλοποίηση κρυπτογραφικών αλγορίθμων).

Ασφάλεια

Το κατεξοχήν και βασικότερο χαρακτηριστικό των έξυπνων καρτών είναι η το αυξημένο επίπεδο ασφάλειας που παρέχουν. Η java η οποία ενσωματώνει χαρακτηριστικά ασφάλειας είναι η καταλληλότερη γλώσσα και ταιριάζει σε περιβάλλοντα έξυπνων καρτών. Για παράδειγμα η πρόσβαση σε όλες τις μεθόδους και τις μεταβλητές είναι απολύτως ελεγχόμενη. Επιπλέον έχει σχεδιαστεί με τέτοιο τρόπο ώστε να παρέχει ασφάλεια εκτέλεσης του κώδικα σε δίκτυο. Δεν υπάρχουν

δείκτες (pointers), ώστε να μπορεί να γίνει αυθαίρετη προσπέλαση διευθύνσεων της μνήμης και παρέχει προστασία από τη δράση των ιών. Τέλος χρησιμοποιεί έναν ισχυρό μηχανισμό για τον έλεγχο των αναμενόμενων ή μη-αναμενόμενων σφαλμάτων (exception handling).

Ανεξαρτησία από το hardware που χρησιμοποιείται
Η java card τεχνολογία δεν εξαρτάται από τον τύπο του υλικού το οποίο χρησιμοποιείται. Όλες οι εφαρμογές μπορούν να τρέχουν σε οποιοδήποτε επεξεργαστή έξυπνων καρτών (8,16, ή 32 -bit). Με βάση την αρχιτεκτονική σχεδίαση της java card, τα java card applets βρίσκονται στην κορυφή αυτής και ως εκ τούτου είναι ανεξάρτητα από τις έξυπνες κάρτες (ως υλικό), ενώ μπορούν να φορτωθούν σε όλες τις java cards χωρίς να χρειάζεται επαναμεταγλώττιση.

Δυνατότητα συνύπαρξης και διαχείρισης πολλαπλών εφαρμογών
Μια java card μπορεί να "φιλοξενεί" πολλές εφαρμογές όπως για παράδειγμα εφαρμογές ηλεκτρονικού πορτοφολιού, αυθεντικοποίησης, υγείας κ.α. οι οποίες ενδέχεται να παρέχονται από διαφορετικούς παροχείς υπηρεσιών. Η java card διαθέτει έναν μηχανισμό για τον διαχωρισμό των διαφόρων applets ο οποίος επιτυγχάνεται με την ύπαρξη ενός firewall. Έτσι τα applets δεν μπορούν να αποκτήσουν πρόσβαση σε κάποιο άλλο applet εκτός και αν έχουν το δικαίωμα να το κάνουν. Επιπλέον μετά την έκδοση της κάρτας αν χρειαστεί να προστεθούν επιπλέον εφαρμογές δεν χρειάζεται να γίνει για δεύτερη φορά έκδοση της κάρτας ή να χρησιμοποιηθεί διαφορετική κάρτα. Αυτό που χρειάζεται είναι απλά να τοποθετηθούν τα καινούρια applets.

Συμβατότητα με τα ήδη υπάρχοντα πρότυπα έξυπνων καρτών
Η java card τεχνολογία είναι συμβατή με το διεθνές πρότυπο ISO 7816 και μπορεί εύκολα να υποστηρίξει εφαρμογές και συστήματα έξυπνων καρτών συμβατά με το πρότυπο αυτό.

7.2 Σχεδιασμός Java Card Applet

Στην παράγραφο αυτή περιγράφουμε τα βήματα που απαιτούνται για την ανάπτυξη ενός Java Card Applet το οποίο θα μπορεί να διαχειρίζεται έξυπνες φοιτητικές κάρτες.
Το πρώτο βήμα είναι η φάση σχεδίασης κατά την οποία πρέπει να καθοριστούν οι λειτουργίες –συναρτήσεις του applet, να οριστούν τα AIDs του applet και του πακέτου το οποίο θα περιέχει το applet και τέλος ο τρόπος με τον οποίο θα επικοινωνεί το applet με την συσκευή ανάγνωσης της κάρτας.

7.2.1 Λειτουργικότητα του Student applet

Βασικές λειτουργίες του Student applet θα είναι η αποθήκευση προσωπικών πληροφοριών όπως όνομα φοιτητή, ηλεκτρονική διεύθυνση κ.τ.λ. και η ανάκτηση αυτών και η υλοποίηση αυθεντικοποίησης ώστε να έχουν πρόσβαση στα δεδομένα της κάρτας εξουσιοδοτημένοι και μόνο χρήστες. Το τελευταίο, παρέχεται μέσω ενός αλγορίθμου ασφαλείας ο οποίος απαιτεί ένα PIN από το χρήστη πριν αυτός αποκτήσει πρόσβαση στα δεδομένα της κάρτας.

7.2.2 Ορισμός των AIDs του Student applet

Όταν η Java Card εισαχθεί στον αναγνώστη, αυτός επιλέγει το applet το οποίο βρίσκεται στην κάρτα και στέλνει μια σειρά από εντολές οι οποίες πρέπει να εκτελεστούν. Κάθε applet αναγνωρίζεται και επιλέγεται με βάση το μοναδικό αναγνωριστικό του το οποίο καλείται AID (application identifier). Έτσι σύμφωνα με το πρότυπο ISO 7816 ένα AID είναι μια ακολουθία από bytes το μήκος του οποίου κυμαίνεται από 5 έως 16 bytes (εικόνα 5.13). Ο παρακάτω πίνακας αποτελεί ένα ενδεικτικό παράδειγμα.

Field	**Value**	**Length**
RID	0xa0,0x00,0x00,0x00,0x62	5 bytes
Package PIX	0x03, 0x01, 0x0c, 0x06	4bytes
Applet PIX	0x03,0x01,0x0c,0x06, 0x01	5bytes

Ένα applet το οποίο τρέχει σε μια Java Card επικοινωνεί με την εφαρμογή δηλαδή με την εφαρμογή που τρέχει στον αναγνώστη στέλνοντας APDUs (application protocol data units εικόνες 3.14, 3.15). Έτσι η διεπαφή μεταξύ κάρτας και host εφαρμογής είναι ένα σύνολο APDU εντολών οι οποίες πρέπει να συμφωνηθούν εξαρχής και να υποστηρίζονται και από τις δύο πλευρές. Ένα Java Card applet θα πρέπει να υποστηρίζει μια εντολή SELECT APDU ώστε να μπορεί να επιλέγεται το συγκεκριμένο applet μέσω του AID που αντιστοιχεί σε αυτό καθώς και ένα σύνολο εντολών μέσω των οποίων θα γίνεται η επεξεργασία των APDUs και εξαρτώνται πάντα από την εφαρμογή του applet. Πριν περιγράψουμε αναλυτικά τις εντολές APDU και τις αντίστοιχες απαντήσεις τους θα πρέπει να υπενθυμίσουμε ότι μια εντολή Command APDU θα πρέπει να είναι της μορφής.

CLA	INS	P1	P2	Lc	Data Field	Le

Και η αντίστοιχη Response APDU της μορφής

Data Field	Status word

7.2.3 Εντολές commands & responses APDUs

Για την εφαρμογή μας Student Applet διακρίνουμε τις ακόλουθες εντολές

SELECT APDU command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
0x0	0xA4	0x04	0x00	0x0A	0xa0,0x00,0x00,0x00,0x62, 0x03,0x01,0x0c,0x06, 0x01	-

SELECT APDU response APDU

Data Field	Status word
No data	0x9000 (επιτυχημένη επεξεργασία εντολής)
	0x6999 (λάθος, η επιλογή του applet απέτυχε)

VERIFY APDU command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
0xB0	0x20	0x00	0x00	Μήκος του κωδικού PIN	Ο κωδικός PIN	-

VERIFY APDU command APDU

Data Field	Status word
No data	0x9000 (επιτυχημένη επεξεργασία εντολής)
	0x6300 (λάθος στη διαδικασία πιστοποίησης)

SET NAME command

CLA	INS	P1	P2	Lc	Data Field	Le
0xB0	0x30	0x00	0x00	Μήκος του name(0x0a)	Name	-

SET NAME response

Data Field	Status word
No Data	0x9000 (επιτυχημένη επεξεργασία εντολής)

SET ID command

CLA	INS	P1	P2	Lc	Data Field	Le
0xB0	0x40	0x00	0x00	Μήκος του id(0x05)	id	-

SET ID response

Data Field	Status word
No Data	0x9000 (επιτυχημένη επεξεργασία εντολής)

GET NAME command

CLA	INS	P1	P2	Lc	Data Field	Le
0xB0	0x50	0x00	0x00			(1byte)

GET NAME response

Data Field	Status word
Name	0x9000 (επιτυχημένη επεξεργασία εντολής)

GET ID command

CLA	INS	P1	P2	Lc	Data Field	Le
0xB0	0x60	0x00	0x00			(1byte)

GET ID response

Data Field	Status word
ID	0x9000 (επιτυχημένη επεξεργασία εντολής)

Το επόμενο βήμα μετά τον προσδιορισμό των εντολών Command και APDU response APDU ακολουθεί η υλοποίηση του κώδικα του Student Applet.

7.2.4 Ο κώδικας του Student Applet

Ο κώδικας είναι διαθέσιμος στο Παράρτημα Α

7.3 Διαδικασία ελέγχου του Student Applet με χρήση των Java Card Framework Tools και αναμενόμενα αποτελέσματα

Το τελευταίο βήμα προτού η εφαρμογή φορτωθεί σε μια java card είναι ο έλεγχος του applet ώστε να διαπιστωθεί ένα αυτό εκτελεί σωστά τις λειτουργίες για τις οποίες έχει σχεδιαστεί. Αυτό επιτυγχάνεται με τη χρήση των εργαλείων τα οποία παρέχει το Java Card Development Kit (JCDK).
Το πρώτο εργαλείο που χρησιμοποιείται είναι ο ***converter*** ο οποίος μετατρέπει το αρχείο class που παράγει η μεταγλώττιση του κώδικα σε ένα αρχείο τύπου CAP το οποίο στη συνέχει θα φορτωθεί στην κάρτα. Αρχικά πρέπει να δημιουργήσουμε ένα αρχείο το οποίο καλείται configuration file, έχει κατάληξη .opt και αποτελεί την είσοδο του converter. Το αρχείο αυτό θα πρέπει να είναι της ακόλουθης μορφής:

```
-out EXP JCA CAP
```

Όπου δηλώνουμε τους τύπους των αρχείων που θα παράγει ο converter ως έξοδο. Αυτά τα αρχεία είναι το αρχείο CAP,το αρχείο export file EXP, και τέλος το Java Card Assembly File.

```
-exportpath .
```

Εδώ δηλώνονται οι διευθύνσεις στις οποίες ο converter θα ψάξει για τα export files.

```
-applet 0xa0:0x00:0x00:0x00:0x62:0x03:0x01:0x0c:0x06:0x01
com.sun.javacard.samples.MyProject.StudentApplet
com.sun.javacard.samples.MyProject
0xa0:0x00:0x00:0x00:0x62:0x03:0x01:0x0c:0x06
```

Στο σημείο αυτό πρέπει να καθοριστούν τα AIDs του applet και του πακέτου στο οποίο ανήκει το applet όπως επίσης και οι ονομασίες τους .

Το αμέσως επόμενο βήμα είναι να τρέξουμε την εντολή converter -config file.opt σε command prompt όπου file η ονομασία του αρχείου.

Στη συνέχεια χρησιμοποιούμε το εργαλείο ***Java Card Workstation Development Environment tool***. Αυτό παίρνει ως είσοδο ένα αρχείο .app το οποίο θα πρέπει να περιλαμβάνει τα ακόλουθα αντικείμενα και ουσιαστικά είναι μια λίστα με τα applets που θα χρησιμοποιηθούν:

```
com.sun.javacard.installer.InstallerApplet
0xa0:0x0:0x0:0x0:0x0:0x62:0x3:0x1:0x8:0x1
```

Αυτό πρέπει πάντα να περιέχεται αρχικά σε ένα οποιοδήποτε java card WDE configuration file.

```
com.sun.javacard.samples.myProject.StudentApplet
0xa0:0x00:0x00:0x00:0x62:0x03:0x01:0x0c:0x06:0x01
```

Αποτελεί το δεύτερο applet μαζί με την ονομασία του και το AID που αντιστοιχεί σε αυτό.

Αφού λοιπόν έχουμε δημιουργήσει το αρχείο file.app τρέχουμε το συγκεκριμένο εργαλείο με την εντολή jcwde -p <port #> file.app όπου η παράμετρος –p μας επιτρέπουμε να καθορίσουμε τον αριθμό της TCP/IP θύρας ή μπορούμε να χρησιμοποιήσουμε τον default που είναι 9025.

Το τελευταίο εργαλείο που θα χρησιμοποιηθεί είναι το Apdutool το οποίο παίρνει ως είσοδο ένα αρχείο script .scr και πρέπει να περιλαμβάνει όλες τις εντολές command APDU και το οποίο ως έξοδο παράγει ένα αρχείο περιεχόμενα του οποίου είναι οι εντολές command APDU με τις αντίστοιχες responses APDU.
Η μορφή του αρχείου file.scr πρέπει να είναι η ακόλουθη:

```
powerup;

// Select the applet
0x00 0xA4 0x04 0x00 0x0A 0xa0 0x00 0x00 0x00 0x62 0x03 0x01 0x0c
0x06 0x01;

// send verify cmd
0xb0 0x20 0x00 0x00 0x08 0x31 0x32 0x33 0x34 0x35 0x36 0x037 0x38
0x00;

//Send SET_NAME cmd
0xb0 0x30 0x00 0x00 0x0a 0x4a 0x6f 0x68 0x6e 0x53 0x6d 0x69 0x74
0x68 0x00 0x00;

// send verify cmd
0xb0 0x20 0x00 0x00 0x08 0x31 0x32 0x33 0x34 0x35 0x36 0x037 0x38
0x00;

//Send SET_ID cmd
0xb0 0x40 0x00 0x00 0x05 0x54 0x31 0x30 0x32 0x34 0x00;

//Send GET_NAME cmd
0xb0 0x50 0x00 0x00 0x00 0x00 0x0a;

//Send GET_ID cmd
0xb0 0x60 0x00 0x00 0x00 0x00 0x05;
// *** SCRIPT END ***
powerdown;
```

Το παραπάνω αρχείο file.scr αποτελεί την είσοδο του εργαλείου apdutool και παράγει το αρχείο file.out μετά την εκτέλεση της εντολής `apdutool file.scr > file.out`. Τα αναμενόμενα αποτελέσματα τα οποία και υποδηλώνουν την ορθή λειτουργία του applet πρέπει να είναι τα παρακάτω:

```
Java Card 2.1.1 ApduTool (version 1.1)
Copyright (c) 2000 Sun Microsystems, Inc. All rights reserved.
Opening connection to localhost on port 9,025.
Connected.
CLA: 00, INS: a4, P1: 04, P2: 00, Lc: 0a, a0, 00, 00, 00, 62, 03,
01, 0c, 06, 01, Le: 00, SW1: 90, SW2: 00
CLA: b0, INS: 20, P1: 00, P2: 00, Lc: 08,31,32,33,34,35,36,37,38,
Le: 00, SW1: 90, SW2: 00
```

```
CLA: b0, INS: 30, P1: 00, P2: 00, Lc: 0a, 4a, 6f, 68, 6e, 53, 6d,
69, 74, 68, 00, 00, Le: 00, SW1: 90, SW2: 00
CLA: b0, INS: 20, P1: 00, P2: 00, Lc: 08, 31,32,33,34,35,36,37,38,
Le: 00, SW1: 90, SW2: 00
CLA: b0, INS: 40, P1: 00, P2: 00, Lc: 05, 54, 31, 30, 32, 34, Le:
00, SW1: 90, SW2: 00
CLA: b0, INS: 50, P1: 00, P2: 00, Lc: 00, Le: 0a,4a, 6f, 68, 6e,
53, 6d, 69,74, 68, 00,  SW1: 90, SW2: 00
CLA: b0, INS: 60, P1: 00, P2: 00, Lc: 00, Le: 05, 54, 31, 30, 32,
34, 00 SW1: 90, SW2: 00
```

Παράρτημα Α

```
package com.sun.javacard.samples.MyProject;
import javacard.framework.*;
public class StudentApplet extends Applet {
/* ορισμός των σταθερών μεταβλητών */
// το πεδίο CLA στην επικεφαλίδα της command APDU
final static byte StudentApplet_CLA =(byte)0xB0;
// τα πεδία INS στις επικεφαλίδες των command APDU
final static byte VERIFY = (byte) 0x20;
final static byte SET_NAME = (byte) 0x30;
final static byte SET_ID = (byte) 0x40;
final static byte GET_NAME = (byte) 0x50;
final static byte GET_ID= (byte) 0x60;
// ο μέγιστος αριθμός προσπαθειών εισαγωγής του PIN
//  προτού αυτός να κλειδωθεί
final static byte PIN_TRY_LIMIT =(byte)0x03;
// μέγιστο μέγεθος του κωδικού PIN
final static byte MAX_PIN_SIZE =(byte)0x08;
// στην περίπτωση που αποτύχει η πιστοποίηση επιστρέφεται
// η ακόλυθη status word
final static short SW_VERIFICATION_FAILED =0x6300;
// η παρακάτω status word σηματοδοτεί ότι απαιτείται
// αυθεντικοποίηση
final static short SW_PIN_VERIFICATION_REQUIRED =0x6301;
//μέγιστο μέγεθος του πεδίου ID
final static short MAX_ID_SIZE = ( short )5;
//μέγιστο μέγεθος του πεδίου name
final static short MAX_NAME_SIZE = ( short )10;

OwnerPIN pin;
byte [] StudentName=null;
byte [] StudentId=null;

private StudentApplet (byte[] bArray,short bOffset,byte bLength){
// αρχικά πρέπει να δεσμευτεί ή μνήμη την ποία θα χρειαστεί το applet
 pin = new OwnerPIN(PIN_TRY_LIMIT,   MAX_PIN_SIZE);
 byte iLen = bArray[bOffset]; //  aid length
 bOffset = (short) (bOffset+iLen+1);
 byte cLen = bArray[bOffset]; // info length
 bOffset = (short) (bOffset+cLen+1);
 byte aLen = bArray[bOffset]; // applet data length
  pin.update(bArray, (short)(bOffset+1), aLen);
  register();

  }

public static void install(byte[] bArray, short bOffset, byte bLength){

    // δημιουργία ενός στιγμιοτύπου του StudentApplet
```

```
    new StudentApplet(bArray, bOffset, bLength);
  } // τέλος της μεθόδου install

public boolean select() {

// Το applet μπορεί να επιλεχθεί εφόσον δοθεί το σωστόpin

    if ( pin.getTriesRemaining() == 0 )
      return false;

    return true;

  }// τέλος της μεθόδου select

public void deselect() {

    // reset the pin value
    pin.reset();

  }

public void process(APDU apdu) {

//Το αντικείμενο APDU χρησιμοποιεί έναν πίνακα byte (buffer) ώστε
//να μεταφέρει τις εισερχόμενες και εξερχόμενες επικεφαλίδες των APDU
//καθώς και τα δεδομένα μεταξύ κάρτας και αναγνώστη
//Μέχρι στιγμής μόνο τα 5 πρώτα bytes της επικεφαλίδας είναι διαθέσιμα
//στον APDU buffer

  byte[] buffer = apdu.getBuffer();
  buffer[ISO7816.OFFSET_CLA] = (byte)(buffer[ISO7816.OFFSET_CLA] &
(byte)0xFC);

  if ((buffer[ISO7816.OFFSET_CLA] == 0) &&
      (buffer[ISO7816.OFFSET_INS] == (byte)(0xA4)) )
     return;

 // Πρέπει να ελεγχθεί το πεδίο CLA και το πεδίο INS
 //ώστε να γνωρίζει το applet ποιά εντολή πρέπει να εκτελέσει

if (buffer[ISO7816.OFFSET_CLA] != StudentApplet_CLA)
  ISOException.throwIt(ISO7816.SW_CLA_NOT_SUPPORTED);
 switch (buffer[ISO7816.OFFSET_INS]) {

      case VERIFY:       verify(apdu);
                             return;
      case SET_NAME:     SetName(apdu);
                             return;
      case SET_ID:       SetId(apdu);
                            return;
      case GET_NAME:   GetName(apdu);
                           return;
      case GET_ID:       GetId(apdu);
```

```
                        return;
    default:    ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
   }

 }  // end of process method

 // Μέθοδος setId
private void SetId(APDU apdu) {

  // προηγείται η αυθεντικοποίηση
  if ( ! pin.isValidated() )
 ISOException.throwIt(SW_PIN_VERIFICATION_REQUIRED);

   byte[] buffer = apdu.getBuffer();
    //το πεδίο Lc ορίζει τον αριθμό των bytes
   //του πεδίου data της εντολής command APDU
   byte numBytes = buffer[ISO7816.OFFSET_LC];

   byte byteRead =(byte)(apdu.setIncomingAndReceive());

    if ((byteRead != 5 )|| (numBytes != 5))
    {
       ISOException.throwIt( ISO7816.SW_WRONG_LENGTH );
     }

   byte[] idData = new byte[ MAX_ID_SIZE ];
   Util.arrayCopy( buffer, ISO7816.OFFSET_CDATA, idData, ( short
)0,MAX_ID_SIZE );
   Util.arrayCopy( idData, (short)0, StudentId, ( short )0,MAX_ID_SIZE );

 } //τέλος της μεθόδου setId

// Μέθοδος SetName
private void SetName(APDU apdu) {

   if ( ! pin.isValidated() )
 ISOException.throwIt(SW_PIN_VERIFICATION_REQUIRED);

   byte[] buffer = apdu.getBuffer();
   byte numBytes = buffer[ISO7816.OFFSET_LC];
   byte byteRead =(byte)(apdu.setIncomingAndReceive());

    if ((byteRead != 10 )||(numBytes!=10))
    {
       ISOException.throwIt( ISO7816.SW_WRONG_LENGTH );
     }

   byte[] nameData = new byte[ MAX_NAME_SIZE ];
   Util.arrayCopy( buffer, ISO7816.OFFSET_CDATA, nameData, ( short
)0,MAX_NAME_SIZE );
   Util.arrayCopy( nameData, (short)0, StudentName, ( short )0,MAX_NAME_SIZE
);
```

```
 } //τέλος της μεθόδου SetName

//Μέθοδος GetName////
private void GetName( APDU apdu ) {
  byte[] buffer = apdu.getBuffer();
  short le = apdu.setOutgoing();

  if ( le < 10 )
      ISOException.throwIt(ISO7816.SW_WRONG_LENGTH);

  Util.arrayCopy( StudentName, ( short )0, buffer, ( short )0, MAX_NAME_SIZE);
  apdu.setOutgoingAndSend( ( short )0, MAX_NAME_SIZE );

   }

 //Μέθοδος getId
private void GetId( APDU apdu ){
  byte[] buffer = apdu.getBuffer();
  short le = apdu.setOutgoing();
  if ( le < 5 )
      ISOException.throwIt(ISO7816.SW_WRONG_LENGTH);

  Util.arrayCopy( StudentId, ( short )0, buffer, ( short )0, MAX_ID_SIZE);
  apdu.setOutgoingAndSend( ( short )0, MAX_ID_SIZE );

  }

private void verify(APDU apdu) {
   byte[] buffer = apdu.getBuffer();
   // ανάκτηση του PIN
   byte byteRead =(byte)(apdu.setIncomingAndReceive());
   //έλεγχος του pin
   if ( pin.check(buffer, ISO7816.OFFSET_CDATA,byteRead) == false )
    ISOException.throwIt(SW_VERIFICATION_FAILED);
}

}
```

Βιβλιογραφία

[1]. Ασφάλεια Πληροφοριακών Συστημάτων και Δικτύων, Γ.ΠΑΓΚΑΛΟΥ, Ι.ΜΑΥΡΙΔΗ Κεφάλαιο 3. Αναγνώριση και Αυθεντικοποίηση

[2]. iit.demokritos.gr/cip-conf/ presentations/4.2-Thomopoulos.pdf, Βιομετρικά Συστήματα

[3]. www.nextgenchina.com/kaiyuan/ presentations/MIPS_Smart_Card.pdf

[4]. www.redbooks.ibm.com/redbooks/pdfs/sg245239.pdf, Λειτουργικό σύστημα έξυπνων καρτών

[5]. http://www.maxking.com/smartcardbackground.htm , Smart Card Programming

[6]. http://www.maxking.com/smartcardcharacteristics.htm, Physical Characteristics of smart cards

[7]. http://www.maxking.com/smartcardstandards.htm, Some Basic Standards for smart cards

[8]. http://www.maxking.com/smartcardcommands.htm, smart card Commands

[9]. http://unix.be.eu.org/docs/smart-card-developer-kit/ewtoc.html

[10]. http://www.javaworld.com , java card

[11]. www.sspsolutions.com/files/ Introduction_to_Smart_Cards.pdf

[12].www.ebusinessforum.gr/index.php?op=modload&odname=Downloads&action=downloadsview&pageid=354, Βασικές έννοιες έξυπνων καρτών και πρότυπα

[13]. www.opencard.org/docs/gim/ocfgim.html, Open card framework

[14]. Ασφάλεια Πληροφοριακών Συστημάτων Σ.ΚΑΤΣΙΚΑΣ, Δ.ΓΚΡΙΤΖΑΛΗΣ, Σ.ΓΚΡΙΤΖΑΛΗΣ , Αρχιτεκτονική έξυπνων καρτών, Διαδικασία αυθεντικοποίησης

[15]. Δίκτυα Επικοινωνιών Jean Walrand , Πρότυπο OSI

[16].http://www.stat.sinica.edu.tw/library/cd_database/Dr_Dobbs/articles/2000/0002/0002f/0002f.htm#rf1 , OpenCard Framework Application Development

[17]. http://www.opencard.org/docs/gim/ocfgim.html, OpenCard Framework General Information Web Document

[18]. http://www.opencard.org/docs/pguide/PGuide.html, OpenCard Framework 1.2 Programmer's Guide

[19]. http://www.opencard.org/docs/1.2/index.html, Javadoc generated API documentation from the OpenCard Framework V 1.2

[20]. http://www.opencard.org/docs/ocfpcsc.pdf

[21]. http://www.gemplus.com/techno/opencard/whitepapers/ibm/node2.html

[22]. http://www-128.ibm.com/developerworks/java/library/j-opencard-framework/index.html#h0

[23]. http://www.gemplus.com/techno/opencard/whitepapers/ctst2000/paper/, OpenCard Introduction & Overview

[24].http://developers.sun.com/techtopics/mobility/javacard/articles/javacard1

[25]. java.sun.com/developer/Books/consumerproducts/javacard/ch03.pdf

[26].http://www.it.iitb.ac.in/~kirang/academic/Seminar/javacard/JavaCardSecurityWhitePaper.pdf

[27]. www.pochendorfer.com/macchiato/stuff/objectsharing.pdf