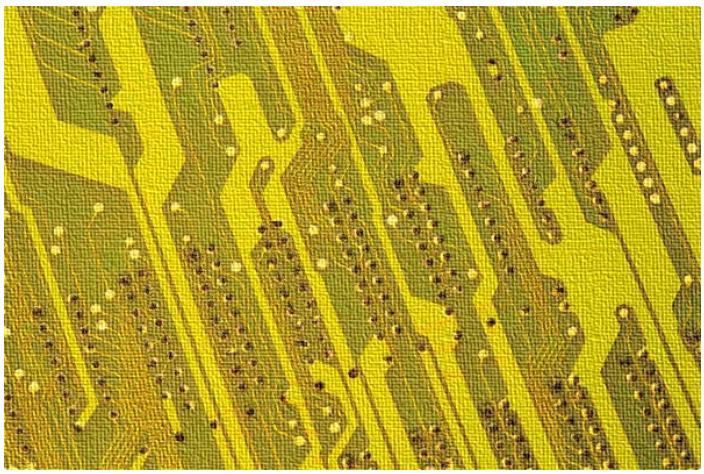
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη Εφαρμογών Σε Έξυπνες Κάρτες



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΑΘΑΝΑΣΙΑΣ Κ. ΔΗΜΗΤΡΑΚΗ

Επιβλέπων : Κιαμάλ Ζ. Πεκμεστζή
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2004

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη Εφαρμογών Σε Έξυπνες Κάρτες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΑΘΑΝΑΣΙΑΣ Κ. ΔΗΜΗΤΡΑΚΗ

Επιβλέπων : Κιαμάλ Ζ. Πεκμεστζή
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 2η Μαρτίου 2004.

.....	.....	.....
Κιαμάλ Ζ. Πεκμεστζή	Παναγιώτης Τσανάκας	Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επίκ. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2004

.....

ΑΘΑΝΑΣΙΑ Κ. ΔΗΜΗΤΡΑΚΗ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η διπλωματική αυτή εργασία έχει ως στόχο την γνωριμία και εξοικείωση με τις λεγόμενες "έξυπνες κάρτες", μία σύγχρονη και νέα σχετικά τεχνολογία που χρησιμοποιείται παγκοσμίως σε όλο και περισσότερες και πιο προηγμένες εφαρμογές. Κάποιες γνωστές εφαρμογές με έξυπνες κάρτες σχετίζονται με τις τηλεκάρτες, τις κάρτες SIM στην κινητή τηλεφωνία, κάρτες πρόσβασης και αναγνώρισης σε προστατευμένους χώρους, το ηλεκτρονικό πορτοφόλι και τις κάρτες διοδίων.
Οι έξυπνες κάρτες χωρίζονται σε συγκεκριμένες κατηγορίες, ανάλογα με το αν περιέχουν μικροεπεξεργαστή ή όχι και με το αν επικοινωνούν με ηλεκτρικές επαφές ή ασύρματα. Στις εφαρμογές που αναπτύχθηκαν στα πλαίσια αυτής της διπλωματικής, χρησιμοποιήθηκαν κάρτες με μικροεπεξεργαστή και ηλεκτρικές επαφές. Επίσης χρησιμοποιήθηκε ένας reader καρτών (συσκευή ανάγνωσης και εγγραφής καρτών) που συνδέεται με τον υπολογιστή στον οποίο τρέχει η εκάστοτε εφαρμογή.
Αρχικά έγινε μία επαφή με τα χαρακτηριστικά της τεχνολογίας των έξυπνων καρτών, όπως το λειτουργικό τους σύστημα (COS), το πρωτόκολλο επικοινωνίας T=0 μεταξύ καρτών και reader, τα πρότυπα ISO στα οποία βασίζονται, καθώς και με τη δομή και τις ιδιότητες των αρχείων των καρτών, τις εντολές που ανταλλάσσουν με το host σύστημα και τον reader (APDU Commands) και τις προηγμένες μεθόδους ασφαλείας που υποστηρίζουν όπως το Secure Messaging, τα παραγόμενα MAC κρυπτογράμματα με τη μέθοδο 3DES, τα Secret Keys και Secret Codes που προστατεύουν αρχεία και εντολές, τα Transaction proofs.
Επίσης υπήρξε εξοικείωση με συγκεκριμένα interfaces (Application Programming Interface – API) για την επικοινωνία host συστήματος με τους reader και μέσω αυτών και με τις κάρτες καθώς και εκτεταμένη χρήση εντολών και μεταβλητών των βιβλιοθηκών των interfaces αυτών.
Στα πλαίσια της εργασίας αναπτύχθηκαν και υλοποιήθηκαν 3 εφαρμογές, κάθε μία από τις οποίες ανταποκρίνεται σε ένα χώρο όπου μπορούν να χρησιμεύσουν ή και ήδη χρησιμοποιούνται έξυπνες κάρτες.
Έτσι στην πρώτη εφαρμογή, την εφαρμογής μισθοδοσίας, οι κάρτες χρησιμοποιούνται από τους εργαζόμενους κατά την είσοδο και έξοδο από το χώρο εργασίας τους για να μετρώνται οι ώρες εργασίας τους, οι οποίες χωρίζονται σε κανονικές ώρες και ώρες υπερωριακής εργασίας.
Η δεύτερη εφαρμογή αφορά το ηλεκτρονικό πορτοφόλι (e-purse) και ένα πρόγραμμα εμπιστοσύνης κατά το οποίο ο κάτοχος κερδίζει κάποιους πόντους για συγκεκριμένο ύψος αγορών.
Στην τρίτη εφαρμογή, γίνεται μία εξομοίωση της χρήσης έξυπνων καρτών ως "εισιτήρια" στα μέσα μαζικής μεταφοράς. Σε αυτή την περίπτωση ο κάτοχος απλά "επικυρώνει" ένα εισιτήριο και επιτρέπει την είσοδό του στο αντίστοιχο μέσο συγκοινωνίας.
Με τα αποτελέσματα των εξομοιώσεων των εφ
αρμογών αυτών έγινε σαφές ότι με σωστό σχεδιασμό, οι δυνατότητες που προσφέρουν οι έξυπνες κάρτες είναι πολύ μεγάλες.

Λέξεις Κλειδιά: έξυπνες κάρτες, reader καρτών, COS, API, APDU, MAC, e-purse, Secure Messaging, Secret Key, Secret Code, 3DES, Transaction Proofs.

Abstract

This thesis intends to make a first contact with the technology of the so-called “smart cards” and familiarize with its basic concepts. The technology of smart cards is modern and relatively new but still it is used worldwide in increasingly more advanced applications. Some well-known applications on smart cards involve public telephony, SIM cards in mobile telephony, access control and identification cards, the electronic purse and toll cards.
Smart cards are categorized according to whether they include a microprocessor or not and whether they communicate through electrical contacts or in a contactless way. In the applications that were developed in the terms of this thesis, the cards that were used are microprocessor cards with electrical contacts. A card reader (a device which reads from and writes data to a card) was also used and was connected to the computer on which the applications were running.
Initially, we became acquainted with basic characteristics of the smart card technology, like their operating system (COS), the communication protocol T=0 which handles communication between the cards and the reader, the ISO standards upon which smart cards are based, as well as the structure and properties of card files, the instructions exchanged between the host system and the reader (APDU Commands) and the advanced security methods that smart cards support like Secure Messaging, MAC cryptograms, 3DES cryptography, Secret Keys and Secret Codes that protect files and instructions, and Transaction Proofs.
Moreover, we became familiar with specific interfaces (Application Programming Interface – API) for the communication between the host system and the reader and through them with the cards as well, and also with the extended use of instructions and parameters of the interfaces’ libraries.
In the terms of this thesis, 3 applications were developed and implemented, each one corresponding to a field that can be or already is associated with smart cards.
In the first application, the salary – overtime application, cards are used by employees during their entrance and exit from their working place so that their working hours are counted. The calculations take into consideration both normal and overtime working hours.
The second application involves an electronic purse and a loyalty application in which the cardholder earns points for specific amounts of purchases.
Finally, in the third application there is a simulation of the smart cards’ use as tickets in transportation means. In this case, the cardholder just validates a “smart” ticket which allows him to enter the according transportation means.
Through the results of the simulations run on these applications it became clear that with the appropriate development, smart cards can offer a wide variety of important applications and have great potentials.

Keywords: smart cards, card readers, COS, API, APDU, MAC, e-purse, Secure Messaging, Secret Key, Secret Code, 3DES, Transaction Proofs.

ΠΡΟΛΟΓΟΣ

Η διπλωματική αυτή εργασία είναι το επιστέγασμα μίας πολυετούς πορείας στον ακαδημαϊκό χώρο του Εθνικού Μετσόβιου Πολυτεχνείου, μίας πορείας δύσκολης, με απαιτήσεις και ανάγκες, αλλά συνάμα ενδιαφέρουσας και γεμάτης γνώσεις, εμπειρίες και ελπίδες.

Στην εκτέλεση και ολοκλήρωση της εργασίας καθοριστικός παράγοντας υπήρξε η ηθική και πρακτική συμπαράσταση των ατόμων που βρέθηκαν κοντά μου και μπόρεσαν να κατανοήσουν το χρόνο και την αφοσίωση που χρειάζεται ένα τέτοιο εγχείρημα.

Επιθυμώ έτσι να τους ευχαριστήσω για την αμέριστη κατανόηση και βοήθειά τους, κάνοντας ιδιαίτερη μνεία στην οικογένειά μου, η οποία αποτέλεσε τον καλύτερο "συμφοιτητή" μου τους τελευταίους αυτούς μήνες.

Επίσης οφείλω να ευχαριστήσω τον Υπεύθυνο Καθηγητή της διπλωματικής εργασίας κ. Κιαμάλ Πεκμεστζή που μου έδωσε τη δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον και σύγχρονο θέμα όπως αυτό των έξυπνων καρτών. Το αντικείμενο της διπλωματικής αυτής αφορά μία νέα ακόμη για την ελληνική αγορά τεχνολογία, με πολλές δυνατότητες εξέλιξης και διείσδυσης στην καθημερινή πρακτική.

Ιδιαιτέρως θέλω τέλος να ευχαριστήσω τον Υποψήφιο Διδάκτορα Κωνσταντίνο Γκότση για την σημαντική του συνεισφορά στην πρόοδο της εργασίας, για τη συνεχή υποστήριξή του και κυρίως για την πολύτιμη βοήθειά του.

Ελπίζω η προσπάθεια και η αφοσίωση των τελευταίων μηνών αλλά και οι γνώσεις που συνέλεξα κατά τη διάρκεια των ετών φοίτησης στη Σχολή των Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών να γίνουν εμφανείς μέσα από τις σελίδες που ακολουθούν καθώς επίσης και να αποδώσουν σημαντικούς καρπούς στο μέλλον.

Αθανασία Κ. Δημητράκη

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1

Εισαγωγή

Ανατρέχοντας στην απώτερη ακόμη ιστορία, βλέπουμε ανθρώπους να υιοθετούν, να φέρουν και να χρησιμοποιούν αντικείμενα που συμβολίζουν ή υποδηλώνουν σημαντικά για τον φέροντα χαρακτηριστικά και στοιχεία. Τα μικρά συνήθως αυτά αντικείμενα, χωρίς ιδιαίτερη αξία αυτά καθ' εαυτά, μέσω της μοναδικότητάς τους αποτύπωναν και μετέφεραν αντιπροσωπευτικά χαρακτηριστικά του κατόχου τους, υποδηλώνοντας τη θέση του, τις ιδιότητες και τα προνόμιά του.
Μία σύγχρονη εκδοχή αυτών των αντικειμένων είναι και οι έξυπνες κάρτες, οι οποίες καλύπτουν τις ίδιες βασικές ανάγκες στηριγμένες σε αξιόπιστη τεχνολογία και ευρύ πεδίο εφαρμογών ενώ προσφέρουν ακόμη ευρύτερες προοπτικές. Μικρές σε μέγεθος και εύχρηστες δεν έχουν αξία αυτές καθ΄εαυτές, αποκτούν όμως μεγάλη αξία αν αναλογιστεί κανείς τις υπηρεσίες που μπορούν να προσφέρουν, τις λειτουργίες που υποστηρίζουν και τα στοιχεία που δύνανται να αποθηκεύουν και να μεταφέρουν.
Στη σύγχρονη κοινωνία όπου η ταχύτητα και η ασφάλεια ανταλλαγής πληροφοριών παίζουν κύριο ρόλο στην επιτυχή διεκπεραίωση καθημερινών και μη λειτουργιών και όπου η τεχνολογία έρχεται να αντικαταστήσει κλασικές και συνήθως χρονοβόρες διαδικασίες στις οποίες κυρίαρχη θέση κατείχε ο ανθρώπινος παράγοντας, οι έξυπνες κάρτες έρχονται να διασφαλίσουν την μετάβαση σε μία πραγματικότητα με μεγάλη λειτουργικότητα ποικίλων διεργασιών.

1.1 Αντικείμενο της Διπλωματικής

Η διπλωματική αυτή εργασία έχει ως σκοπό να παρουσιάσει την τεχνολογία των έξυπνων καρτών και να δώσει μία εικόνα βασικών λειτουργιών και χαρακτηριστικών τους. Επιπλέον παρουσιάζεται η επικρατούσα κατάσταση στην αγορά της συγκεκριμένης τεχνολογίας και συζητούνται οι προοπτικές των έξυπνων καρτών στον εμπορικό και τον επιστημονικό τομέα.
Κύριος στόχος της εργασίας είναι η γνωριμία και η εξοικείωση με το αντικείμενο αυτό και η απόκτηση βασικής τεχνογνωσίας σε θέματα έξυπνων καρτών, γεγονός που ανοίγει τον δρόμο για ανάπτυξη πραγματικών εφαρμογών και αναζήτηση προηγμένων λύσεων σε υπάρχοντα ή και μελλοντικά τεχνικά, εμπορικά, υπολογιστικά προβλήματα σε ένα ευρύ φάσμα δραστηριοτήτων.
Περαιτέρω, η εργασία αυτή δίνει την δυνατότητα πρακτικής εφαρμογής και διεύρυνσης όλων των γνώσεων και πληροφοριών που έχουν προσφερθεί μέσα στα πλαίσια του ακαδημαϊκού προγράμματος κατά τη διάρκεια των τελευταίων ετών.
Σαν πράξη πάνω στις έξυπνες κάρτες, σχεδιάστηκαν τρεις σύγχρονες εφαρμογές πάνω σε πλατφόρμα γνωστών προγραμματιστικών εργαλείων όπως η γλώσσα C αλλά και νέων σχεδιαστικών και επικοινωνιακών εργαλείων όπως οι βιβλιοθήκες διεπαφών των καρτών και του αναγνώστη καρτών με το host σύστημα.
Οι εφαρμογές αυτές αφορούν ένα πρόγραμμα μισθοδοσίας υπαλλήλων, τις λειτουργίες ηλεκτρονικού πορτοφολιού και προγραμμάτων εμπιστοσύνης μέσω έξυπνων καρτών καθώς και ένα πρόγραμμα για εξομοίωση της χρήσης έξυπνων καρτών σε μέσα μαζικής μεταφοράς.
Μέσω της σχεδίασης και υλοποίησης των εφαρμογών αυτών, στόχος ήταν η απόκτηση χρήσιμης γνώσης και εμπειρίας πάνω στα βασικά στοιχεία της τεχνολογίας των έξυπνων καρτών, δηλαδή σε στοιχεία όπως το λειτουργικό σύστημα των καρτών, το πρωτόκολλο

επικοινωνίας, τη δομή των αρχείων, τις διάφορες εντολές που υποστηρίζονται καθώς και σε έννοιες που αφορούν την ασφάλεια των ανταλλασσόμενων πληροφοριών, όπως η κρυπτογράφηση 3DES, η μέθοδος Secure Messaging, οι διάφορες συνθήκες πρόσβασης, τα Private και Public keys.
Όλα όσα αναφέρονται στην ενότητα αυτή περιγράφονται και περιλαμβάνονται στα επόμενα κεφάλαια.

1.2 Οργάνωση του τόμου

Για την καλύτερη κατανόηση του αντικείμενου της παρούσας διπλωματικής και για την ευκολία πρόσβασης στις επιμέρους πληροφορίες, το κείμενο έχει χωριστεί σε κεφάλαια και ενότητες, τα οποία θα αναφερθούν ακολούθως περιληπτικά.
Το πρώτο κεφάλαιο περιέχει την εισαγωγή της διπλωματικής εργασίας, δίνοντας πληροφορίες για το αντικείμενό της και το πλαίσιο στο οποίο εκπονήθηκε καθώς επίσης και μία περιγραφή της οργάνωσης των κεφαλαίων.
Το δεύτερο κεφάλαιο περικλείει εισαγωγικά στοιχεία για τις έξυπνες κάρτες, δίνοντας μία πρώτη περιγραφή της τεχνολογίας αυτής και παρέχοντας μια ιστορική αναδρομή πάνω στην παρουσία των έξυπνων καρτών και την εξέλιξή τους. Επίσης παρατίθενται στατιστικά στοιχεία για την αγορά των έξυπνων καρτών τα τελευταία χρόνια και την αναμενόμενη εξάπλωσή τους ενώ παρουσιάζονται και οι κυριότερες εφαρμογές που συναντάμε στην παγκόσμια αγορά και οι οποίες στηρίζονται στην νέα αυτή τεχνολογία.
Στο τρίτο κεφάλαιο παρέχεται λεπτομερής ανάλυση της τεχνολογίας των έξυπνων καρτών, με ανάλυση βασικών τεχνικών στοιχείων για την καλύτερη κατανόηση της δομής τους, πληροφορίες για τους τύπους των καρτών που υπάρχουν καθώς επίσης και εξέταση κύριων χαρακτηριστικών της τεχνολογίας όπως το λειτουργικό σύστημα των καρτών, και το πρωτόκολλο επικοινωνίας μεταξύ καρτών και reader. Επίσης γίνεται μία εισαγωγή στη τεχνολογία των reader – αναγνωστών καρτών.
Το τέταρτο κεφάλαιο αναλύει τα συγκεκριμένα εργαλεία έξυπνων καρτών που χρησιμοποιήθηκαν για την υλοποίηση των εφαρμογών. Εξηγεί δηλαδή χαρακτηριστικά του reader GemPC410 και των καρτών GemClub όπως τα αρχεία τους, οι εντολές που υποστηρίζουν καθώς και οι τεχνικές ασφαλείας που παρέχονται όπως το Secure Messaging.
Στο πέμπτο κεφάλαιο αναλύεται το περιεχόμενο και η σχεδίαση των τριών εφαρμογών της διπλωματικής αυτής με στοιχεία για τις τεχνικές και προγραμματιστικές προδιαγραφές τους και με χαρακτηριστικά της δομής των προγραμμάτων που τις αποτελούν.
Το έκτο κεφάλαιο περιγράφει την υλοποίηση των τριών εφαρμογών, εξηγώντας τις ακριβείς συναρτήσεις, εντολές και μεταβλητές που χρησιμοποιήθηκαν στα επιμέρους προγράμματα και όλα τα αρχεία των καρτών που υποστηρίζουν τις εφαρμογές αυτές.
Στο έβδομο κεφάλαιο παρατίθενται αποτελέσματα της εξομοίωσης των τριών εφαρμογών με χρήση των προγραμμάτων που αναλύθηκαν στο πέμπτο και έκτο κεφάλαιο και επισημαίνονται κύρια στοιχεία των προγραμμάτων αυτών.
Το όγδοο κεφάλαιο περιέχει τον επίλογο της διπλωματικής με τα συμπεράσματα και τις προοπτικές της τεχνολογίας των έξυπνων καρτών.
Τέλος, στο ένατο κεφάλαιο υπάρχει η απαραίτητη βιβλιογραφία στην οποία περιέχονται βιβλία, άρθρα και διευθύνσεις στο Διαδίκτυο με χρήσιμες πληροφορίες στο αντικείμενο των έξυπνων καρτών.

2

Εισαγωγή Στις Έξυπνες Κάρτες

Ο όρος και μόνο "έξυπνη κάρτα" εντυπωσιάζει και εξάπτει τη φαντασία τουλάχιστον αυτών που ενδιαφέρονται για τις τεχνολογίες αιχμής, χωρίς να είναι όμως επαρκής για να προσδιορίσει τις ιδιότητες και τις δυνατότητες μίας φαινομενικά απλής πλαστικής κάρτας.
Η έξυπνη κάρτα στην πραγματικότητα ορίζεται ως μία πλαστική κάρτα, συνήθως σε μέγεθος και σχήμα πιστωτικής κάρτας, η οποία όμως περιέχει μνήμη ή/και μικροεπεξεργαστή που της δίνουν τη δυνατότητα αποθήκευσης και επεξεργασίας μεγάλου όγκου δεδομένων και η οποία συμμορφώνεται με διεθνή πρότυπα.
Με απλούς όρους, η έξυπνη κάρτα είναι ένας μικροσκοπικός υπολογιστής με πολύ σημαντικές δυνατότητες και αποτελεί την πιο πρόσφατη εξέλιξη στο χώρο των πλαστικών καρτών, έχοντας ήδη ανοίξει το δρόμο σε σημαντικές και εκτεταμένες εφαρμογές παγκοσμίως. Ο μικροσκοπικός αυτός υπολογιστής, αλλιώς καλούμενος μικροτσίπ, είναι ένα ολοκληρωμένο κύκλωμα με ηλεκτρικές επαφές ή με δυνατότητες ασύρματης επικοινωνίας που συνδυαζόμενος με την κατάλληλη συσκευή υποδοχής καρτών έχει τη δυνατότητα αποθήκευσης και μεταφοράς χιλιάδων bit πληροφορίας καθώς και μεγάλη δύναμη επεξεργασίας αυτών των δεδομένων για την εξυπηρέτηση ποικίλων εφαρμογών.
Κύρια χαρακτηριστικά των έξυπνων καρτών είναι ότι παρέχουν ασφάλεια δεδομένων και συνδιαλλαγών, ταχύτητα και ευκολία χρήσης καθώς επίσης αντοχή στην καταπόνηση και κακή χρήση και μεγάλο διάστημα "ζωής".
Σε αντίθεση με τις γνωστές κάρτες με μαγνητική ταινία, οι έξυπνες κάρτες κατέχουν βασικές και απαραίτητες διεργασίες και πληροφορίες αποθηκευμένες στο σώμα τους προσφέροντας έτσι περισσότερη ασφάλεια καθώς και τη δυνατότητα μεταφοράς σημαντικών δεδομένων χωρίς την ανάγκη σύνδεσης με κεντρικές βάσεις δεδομένων για την άντληση ουσιωδών πληροφοριών. Για αυτό το λόγο η τάση στις σύγχρονες αγορές κυρίως της Ευρώπης, είναι η αντικατάσταση των καρτών μαγνητικής ταινίας από τις έξυπνες κάρτες και η ανάπτυξη όλο και πιο πολύπλοκων εφαρμογών, όλο και πιο αυτοματοποιημένων διαδικασιών.
Για να καταλάβουμε τη σπουδαιότητα της εξέλιξης αυτής, της μετάβασης δηλαδή από τις κάρτες μαγνητικής ταινίας στις έξυπνες κάρτες, είναι προτιμότερο να αναλύσουμε τα χαρακτηριστικά της χρήσης των πρώτων σε διάφορες διαδικασίες.

2.1 Κάρτες Μαγνητικής Ταινίας

Οι κάρτες μαγνητικής ταινίας είναι ευρέως διαδεδομένες και χρησιμοποιούνται από το μεγαλύτερο μέρος του πληθυσμού σε διάφορες καθημερινές και μη συναλλαγές και λειτουργ.

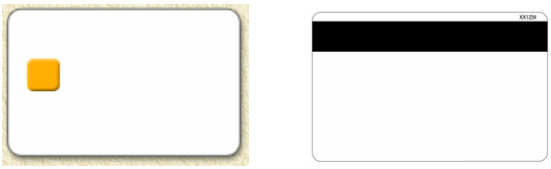
Σχήμα 2.1 - Δύο όψεις κάρτας μαγνητικής ταινίας

Από τις πιστωτικές κάρτες, στις κάρτες αυτόματης ανάληψης μετρητών και από τις κάρτες προγραμμάτων εμπιστοσύνης πολυκαταστημάτων ή αεροπορικών εταιριών στις κάρτες

ελέγχου πρόσβασης σε κτίρια, οι κάρτες μαγνητικής ταινίας χρησιμοποιούνται για να αποθηκεύουν πληροφορίες σε μορφή αναγνώσιμη από μηχανές και έτσι έχουν αυτοματοποιήσει καθημερινές συναλλαγές και διαδικασίες. Η εκτεταμένη τους χρήση έχει σαφώς διευκολύνει τον απλό χρήστη καθώς και διάφορους τραπεζικούς και εμπορικούς φορείς, έχει όμως ταυτόχρονα επιδείξει σημαντικά μειονεκτήματα τα οποία πλέον δεν μπορούν να παρακαμφθούν.
Εξετάζοντας αυτά τα μειονεκτήματα έχουμε να παρατηρήσουμε ότι η κύρια πηγή προβλημάτων έγκειται στο γεγονός ότι τα δεδομένα που αποθηκεύονται στη μαγνητική ταινία μιας κάρτας μπορούν εύκολα να διαβαστούν και να τροποποιηθούν από οποιονδήποτε έχει πρόσβαση στον κατάλληλο εξοπλισμό. Έτσι είναι σαφές ότι εμπιστευτικές και κρίσιμες πληροφορίες όπως ο κωδικός αναγνώρισης του κατόχου, δεν μπορούν να αποθηκεύονται στην ίδια τη κάρτα αλλά αναγκαστικά καταχωρούνται σε κάποια κεντρική βάση δεδομένων. Αυτό σημαίνει ότι για να εκτελεστεί οποιαδήποτε συναλλαγή πρέπει το τερματικό συναλλαγής (π.χ. ATM) να είναι online συνδεδεμένο με κάποιο κεντρικό υπολογιστή για να γίνει πιστοποίηση αυθεντικότητας, διαδικασία χρονοβόρα και με κόστος.
Έτσι, η χρήση των καρτών μαγνητικής ταινίας συνδυάζεται με την ύπαρξη και συντήρηση μεγάλων κεντρικών μονάδων για τη φύλαξη και επεξεργασία των ευαίσθητων δεδομένων, καθώς και με τη συντήρηση κυκλωμάτων για τις απαραίτητες online συνδέσεις μεταξύ κεντρικών βάσεων δεδομένων και σημείων πώλησης - συναλλαγής.
Επιπλέον, οι κάρτες μαγνητικής ταινίας παρουσιάζουν ευαισθησία σε παράγοντες όπως τα μαγνητικά πεδία, οι τυχόν επαφές με αιχμηρά αντικείμενα και η παρατεταμένη χρήση τους, οι οποίοι μπορούν να καταστρέψουν τη μαγνητική ταινία της κάρτας. Επίσης, οι κάρτες αυτές σχεδιάζονται για μία και μόνο εφαρμογή και οποιαδήποτε αλλαγή στα χαρακτηριστικά της εφαρμογής ή στα στοιχεία του κατόχου σημαίνει και αντικατάσταση της ίδιας της κάρτας.
Τα ανωτέρω στοιχεία, με κυριότερο το θέμα της ασφάλειας των δεδομένων και της εγκυρότητας των συναλλαγών, καθιστούν τις κάρτες μαγνητικής ταινίας ένα προϊόν που δεν δύναται να καλύψει πλήρως τις συνεχώς αυξανόμενες ανάγκες και απαιτήσεις της σύγχρονης αγοράς.

2.2 Σύγκριση Καρτών Μαγν/κής Ταινίας Με Έξυπνες Κάρτες

Οι έξυπνες κάρτες, όπως προαναφέρθηκε, έχουν σημαντικά πλεονεκτήματα σε σχέση με τις ευρέως χρησιμοποιούμενες κάρτες μαγνητικής ταινίας και τείνουν να αποτελέσουν τη κυρίαρχη τάση για ανεπτυγμένες, απλές ή και πιο σύνθετες εφαρμογές σε ποικίλους τομείς.



Σχήμα 2.2 - Έξυπνη κάρτα - Κάρτα μαγνητικής ταινίας

Κύρια χαρακτηριστικά των εξυπνων καρτων ειναι οι προηγμενες διαδικασίες ασφάλειας δεδομένων και συνδιαλλαγών, η δυνατότητα μεταφοράς σημαντικών δεδομένων καθώς και η εύκολη και γρήγορη πρόσβαση σε ευαίσθητες πληροφορίες εφόσον αποτελούν ένα κινητό ηλεκτρονικό αρχείο. Επίσης οι έξυπνες κάρτες προσφέρουν ευκολία χρήσης, ανθεκτικότητα, δυνατότητα επικοινωνίας και σύνδεσης με υπολογιστές, ταχύτητα επεξεργασίας και συνήθως υπολογιστική δύναμη.
Συγκρίνοντας τις δύο προαναφερθείσες μορφές πλαστικών καρτών, παρατηρούμε σημαντικές διαφορές σε τομείς που θα αναλυθούν ακολούθως.

- Αποθήκευση Δεδομένων: σε σχέση με τη περιορισμένη δυνατότητα αποθήκευσης πληροφοριών των καρτών μαγνητικής ταινίας (ως 140 byte πληροφορίας), οι έξυπνες

κάρτες έχουν μεγάλη χωρητικότητα, με δυνατότητα αποθήκευσης ως και 80 φορές περισσότερων ηλεκτρονικών δεδομένων (από 1Kbyte ως 32Kbytes πληροφορίας).

- Ασφάλεια: ενώ στις κάρτες μαγνητικής ταινίας οι εκάστοτε πληροφορίες μπορούν εύκολα να αλλοιωθούν ή να αναπαραχθούν από μη έγκυρους χρήστες, οι έξυπνες κάρτες παρέχουν αυξημένη ασφάλεια δεδομένων και συναλλαγών, με τη χρήση διαδικασιών όπως κρυπτογράφηση και κωδικοποίηση.
- Αντοχή / Διάρκεια: σε αντίθεση με την ευαισθησία των καρτών μαγνητικής ταινίας που συνίσταται στη πιθανότητα απομαγνητισμού της ταινίας λόγω χρήσης ή λόγω εξωτερικών μαγνητικών πεδίων, οι έξυπνες κάρτες παρουσιάζουν μεγάλη ανθεκτικότητα και έχουν μεγάλη συγκριτικά διάρκεια ζωής και αντοχή σε αλλεπάλληλες εισαγωγές σε μηχανήματα υποδοχής καρτών 100.000 φορές και πάνω.
- Χρήση: η σχεδίαση των καρτών μαγνητικής ταινίας γίνεται για μία εφαρμογή και η χρήση τους περιορίζεται σε απλά και επαναλαμβανόμενα καθήκοντα, ενώ οι έξυπνες κάρτες υποστηρίζουν πολλαπλές και πολύπλοκες εφαρμογές.
- Ευελιξία: τα δεδομένα μίας κάρτας μαγνητικής ταινίας είναι μόνο αναγνώσιμα με αποτέλεσμα οποιαδήποτε σημαντική αλλαγή στοιχείων να καθιστά αναγκαία την έκδοση νέας κάρτας, ενώ σε μία έξυπνη κάρτα διαδικασίες ανάγνωσης, εγγραφής και ανανέωσης δεδομένων γίνονται εύκολα και γρήγορα.
- Σύνδεση: η χρήση καρτών μαγνητικής ταινίας καθιστά αναγκαία την online σύνδεση με κεντρική βάση δεδομένων για κάθε συναλλαγή, γεγονός που συνεπάγεται συνήθως την ύπαρξη μισθωμένης γραμμής. Το κόστος που αντιστοιχεί στη μίσθωση γραμμής είναι ένα επιπλέον κόστος που δεν υπάρχει στην περίπτωση των έξυπνων καρτών, οι οποίες μπορούν να κάνουν offline ασφαλείς και έγκυρες συναλλαγές τα στοιχεία των οποίων θα περνάνε αν χρειάζεται σε κεντρικό σύστημα σε δεδομένη χρονική στιγμή, ανεξάρτητη της στιγμής συναλλαγής.

Το κόστος κατασκευής έξυπνων καρτών είναι μεγαλύτερο από το αντίστοιχο των καρτών μαγνητικής ταινίας, λόγω όμως της ανθεκτικότητάς τους, της χρησιμοποίησής τους σε ποικίλες εφαρμογές, τη μείωση των οικονομικών απατών και τη μείωση του κόστους τηλ/κής σύνδεσης, οι έξυπνες κάρτες είναι τελικά πιο αποδοτικές ως προς το κόστος.

Αν και οι παράγοντες που ευνοούν τη χρήση των έξυπνων καρτών στη θέση των καρτών μαγνητικής ταινίας είναι σημαντικοί, δεν έχουν το ίδιο βάρος σε όλες τις σύγχρονες αγορές με αποτέλεσμα να μην έχουν την ίδια απήχηση παγκοσμίως. Έτσι χρησιμοποιούνται ήδη ευρέως στις αγορές της Ευρώπης, της Ασίας και της Αφρικής, έχοντας γίνει ένα προϊόν εμπορικά επιτυχημένο. Οι εφαρμογές που στηρίζονται σε έξυπνες κάρτες καλύπτουν τομείς όπως η πρόσβαση και η αναγνώριση ταυτότητας σε διάφορους χώρους, οι ηλεκτρονικές αγορές μέσω του Διαδικτύου και οι τουριστικές επιχειρήσεις, δίνοντας εξελιγμένες δυνατότητες. Υπάρχουν όμως αγορές στις οποίες οι έξυπνες κάρτες, παρότι παρουσιάστηκαν επιτυχώς και ελπιδοφόρα, δεν έχουν καταφέρει ακόμα να καθιερωθούν ως κοινό μέσο συναλλαγών και εφαρμογών.

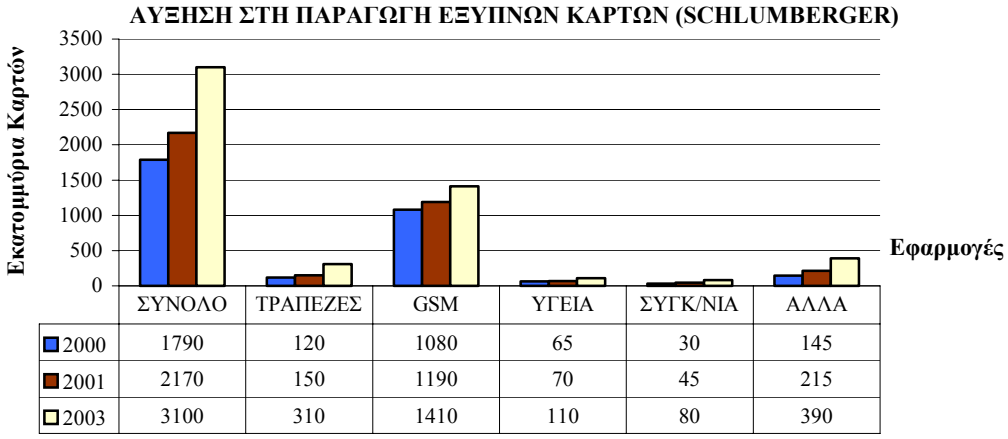
Ένα σημαντικό παράδειγμα είναι η αγορά της Αμερικής στην οποία η τεχνολογία των έξυπνων καρτών είναι ακόμα καινούρια. Το υψηλό κόστος των έξυπνων καρτών σε σχέση με τις κάρτες μαγνητικής ταινίας και η αναγκαιότητα ειδικών συσκευών ανάγνωσης καρτών (card readers) συνεπάγονται μία υψηλή επένδυση για τα Αμερικανικά οικονομικά ιδρύματα τα οποία ήδη έχουν επενδύσει στα συστήματα μαγνητικής ταινίας. Ο χρόνος και το κόστος μίας μεγάλης αλλαγής στη τεχνολογία αυτή αποτέλεσαν μέχρι τώρα ανασταλτικούς παράγοντες για μεγάλες και τολμηρές επενδύσεις. Κύριος όμως ανασταλτικός παράγοντας για την Αμερικανική αγορά αποτελεί το ότι η δομή διεξαγωγής οικονομικών και πληροφοριακών συναλλαγών έχει εξελιχθεί διαφορετικά από ότι στην Ευρώπη. Η Ευρώπη κατάφερε να αναπτύξει την τεχνολογία των έξυπνων καρτών ως ένα αποδοτικό, ως προς το κόστος, τρόπο διεξαγωγής συναλλαγών οι οποίες αποσυνδέθηκαν από τις online διαδικασίες πιστοποίησης που στην Ευρώπη συνεπάγονται μεγάλο τηλεπικοινωνιακό κόστος. Στην Αμερική αντιθέτως το τηλεπικοινωνιακό κόστος είναι χαμηλό και έτσι αποδυναμώνεται ένα σημαντικό προτέρημα της εισαγωγής των έξυπνων καρτών.

2.3 Ιστορική Αναδρομή

Πολλοί θεωρούν ότι οι έξυπνες κάρτες είναι μια πρόσφατη εφεύρεση. Αυτό όμως δε θα μπορούσε να απέχει περισσότερο από την αλήθεια. Στην πραγματικότητα, η ιστορική προέλευση των έξυπνων καρτών μας οδηγεί στη δεκαετία του 70. Η αρχική ιδέα της ενσωμάτωσης μικροτσίπ σε πλαστικές κάρτες γεννήθηκε το 1968 στη Γερμανία από τον Jurgen Dethloff και τον Helmut Grotrupp. Δύο χρόνια αργότερα, το 1970 στην Ιαπωνία, ο εφευρέτης Kunitaka Arimura διατύπωσε μία παρόμοια πατέντα στην ιδέα της έξυπνης κάρτας. Τα πραγματικά θεμέλια όμως για την υλοποίηση της τεχνολογίας των έξυπνων καρτών μπήκαν το 1974 στη Γαλλία από τον ανεξάρτητο εφευρέτη και ερευνητή Roland Moreno. Ο Moreno υλοποίησε πιλοτικά την ένωση πλαστικής κάρτας και μικροτσίπ, το παρουσίασε σε κάποιες τράπεζες στη Γαλλία και τον επόμενο χρόνο το κατοχύρωσε και ως πατέντα.
Η πρώτη έξυπνη κάρτα κατασκευάστηκε τελικά το 1977 από την Motorola και την Bull ενώ συγχρόνως 3 εμπορικοί κατασκευαστές, η Bull, η SGS Thomson και η Schlumberger ξεκίνησαν να αναπτύσσουν εφαρμογές πάνω στη νέα τεχνολογία. Η πρώτη αυτή κάρτα περιείχε δύο μικροτσίπ, δηλαδή ένα μικροελεγκτή και μία ξεχωριστή συσκευή μνήμης. Το 1980 η Motorola παρουσίασε την πρώτη ασφαλή έξυπνη κάρτα με ένα μικροτσίπ, για χρήση στο Γαλλικό τραπεζικό χώρο. Το 1982 έγινε στη Γαλλία το πρώτο εκτεταμένο και πραγματικό τεστ έξυπνων καρτών και συγκεκριμένα τηλεφωνικών καρτών σειριακής μνήμης. Ακολούθως το 1984 έγιναν τα πρώτα τεστ στην παραγωγή των έξυπνων καρτών αυτόματης ανάληψης.
Με την πάροδο των χρόνων, οι έξυπνες κάρτες εξελίσσονταν συνεχώς, και καινούριες εφαρμογές αναπτύσσονταν, κυρίως στην Ευρώπη. Η Γαλλία έχει πρωτοπορήσει όλα αυτά τα χρόνια στο σχεδιασμό και τη χρήση εφαρμογών έξυπνων καρτών και μαζί με τη Γερμανία αποτελούν τις κορυφαίες χώρες σε εισαγωγή ποικίλων εφαρμογών σε έξυπνες κάρτες.
Το 1987 εφαρμόστηκε το πρώτο μεγάλης κλίμακας έργο με έξυπνες κάρτες στην Αμερική ενώ το 1993 οι πρώτες εφαρμογές με κάρτες πολλαπλών διεργασιών δοκιμάστηκαν στην Γαλλία. Το ίδιο έτος ολοκληρώθηκε σχεδόν στη Γαλλία η αντικατάσταση των υπάρχουσων τραπεζικών καρτών με έξυπνες κάρτες και η τάση αυτή εξαπλώθηκε σε άλλες Ευρωπαϊκές και Ασιατικές χώρες.
Έκτοτε η βιομηχανία των έξυπνων καρτών εξαπλώνεται με πολύ μεγάλο ρυθμό και έχει φτάσει σε βαθμό παραγωγής και αποστολής καρτών σχεδόν ίσο με 1.000.000.000 το χρόνο ενώ πλέον οι έξυπνες κάρτες χρησιμοποιούνται σε διάφορες εφαρμογές σε περισσότερες από 90 χώρες παγκοσμίως. Το μεγαλύτερο μερίδιο της αγοράς των έξυπνων καρτών κατέχουν οι εφαρμογές τηλεφωνίας, οι τραπεζικές εφαρμογές, έργα που αφορούν το τομέα της Υγείας καθώς και άλλα ποικίλα σχέδια που θα αναπτύξουμε παρακάτω.

2.4 Στατιστικά Στοιχεία

Σύμφωνα με αναλύσεις της αγοράς έξυπνων καρτών από οικονομικούς παράγοντες και εταιρίες κατασκευής καρτών, έχουν προκύψει έγκυρες αναφορές της κίνησης της αγοράς των έξυπνων καρτών παγκοσμίως. Οι αναφορές αυτές προσδιορίζουν το μέγεθος της παραγωγής έξυπνων καρτών μέσα στο χρονικό διάστημα των τελευταίων ετών και παρουσιάζουν την εξάπλωση αυτής της τεχνολογίας ανά περιοχή. Συγχρόνως, λεπτομερή μοντέλα πρόβλεψης δείχνουν την πιθανή εικόνα της αγοράς των έξυπνων καρτών στα επόμενα χρόνια δίνοντας έτσι την ευκαιρία κυρίως στις εταιρίες να κατανοήσουν τις οικονομικές και επενδυτικές κινήσεις που μπορούν να ακολουθήσουν στην αγορά αυτή.
Με βάση το Σχήμα 2.4.1 που ακολουθεί (διάγραμμα της παραγωγής έξυπνων καρτών ανά τομέα εφαρμογής στα 4 τελευταία χρόνια), παρατηρούμε ότι το μεγάλο μερίδιο κατέχουν εφαρμογές κινητής τηλεφωνίας (GSM) που χρησιμοποιούν την έξυπνη κάρτα ως κάρτα SIM του κινητού τηλεφώνου. Δεύτερος τομέας σημαντικής χρήσης έξυπνων καρτών είναι ο τραπεζικός και ακολουθούν ο τομέας της Υγείας και των Συγκοινωνιών.



	ΣΥΝΟΛΟ	ΤΡΑΠΕΖΕΣ	GSM	ΥΓΕΙΑ	ΣΥΓΚ/ΝΙΑ	ΑΛΛΑ
2000	1790	120	1080	65	30	145
2001	2170	150	1190	70	45	215
2003	3100	310	1410	110	80	390

Σχήμα 2.4.1

Το σημαντικό στοιχείο, σύμφωνα και με το ακόλουθο διάγραμμα (η κόκκινη στήλη αποτελεί πρόβλεψη) είναι ότι κάθε τομέας και κυρίως το σύνολο της παραγωγής παρουσιάζει σταθερά αύξηση από το ένα έτος στο άλλο, δείχνοντας ότι η αγορά των έξυπνων καρτών είναι ακόμα σε ταχεία ανάπτυξη και εξάπλωση.

Σχήμα 2.4.2

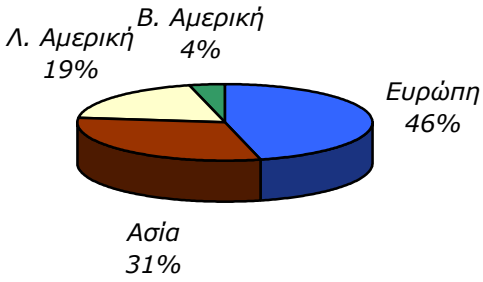
Σημαντικά συμπεράσματα βγάζουμε επίσης παρατηρώντας τη μορφή και το μέγεθος της αγοράς των έξυπνων καρτών ανά περιοχή.

	2001	2002	2003
ΕΥΡΩΠΗ	883	923	974
ΑΣΙΑ	558	597	656
Λ.ΑΜΕΡΙΚΗ	322	346	402
Β.ΑΜΕΡΙΚΗ	38	58	85

Σχήμα 2.4.3

Όπως έχει αναφερθεί και νωρίτερα, οι εφαρμογές έξυπνων καρτών είναι κυρίως διαδεδομένες στην Ευρώπη και την Ασία, με τον Ευρωπαϊκό χώρο να αποτελεί το μεγαλύτερο πεδίο χρήσης της τεχνολογίας αυτής κρατώντας το 46% της αγοράς. Η αγορά της Λατινικής Αμερικής ακολουθεί τρίτη, με μεγέθη κυκλοφορίας έξυπνων καρτών σχετικά κοντά με τα αντίστοιχα της Ασιατικής αγοράς, ενώ η Βόρεια Αμερική βρίσκεται στη τελευταία θέση, παρουσιάζοντας συγκριτικά πολύ μικρά μεγέθη χρήσης εφαρμογών που στηρίζονται σε έξυπνες κάρτες.

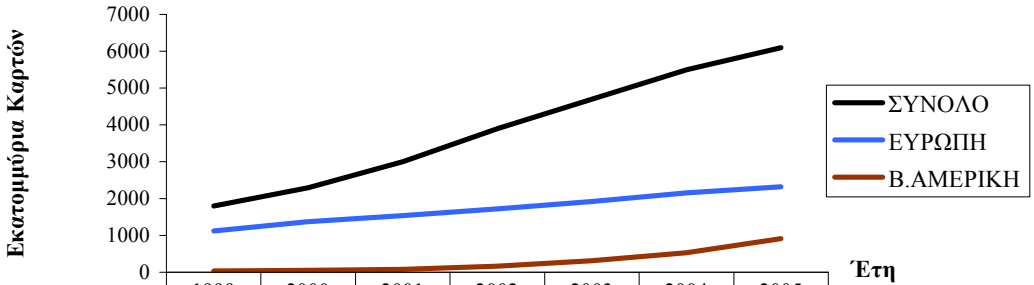
Χρήση Έξυπνων Καρτών Ανά Περιοχή



Σχήμα 2.4.4

Όπως παρατηρούμε και από την πίτα καταμερισμού της αγοράς, η Βόρεια Αμερική κατέχει μόνο το 4% της παγκόσμιας αγοράς. Το γεγονός αυτό οφείλεται σε παράγοντες που έχουν ήδη αναλυθεί, είναι όμως ελπιδοφόρο το γεγονός ότι με την πάροδο των ετών ο αριθμός των έξυπνων καρτών που κυκλοφορούν και χρησιμοποιούνται στην αγορά των Ηνωμένων Πολιτειών παρουσιάζει αύξηση. Σε αντίθεση με την Β. Αμερική, η Ευρώπη κατέχει μερίδιο αγοράς σχεδόν ίσο με το 50%. Η Λατινική Αμερική, η Κίνα, η Ιαπωνία και άλλες Ασιατικές χώρες, όπως είδαμε και προηγουμένως πρόκειται να αποτελέσουν δυνατούς διεκδικητές στη μάχη των μεριδίων της αγοράς.
Όπως τέλος παρατηρούμε από το διάγραμμα που ακολουθεί και παρουσιάζει μία πρόβλεψη για τη χρήση έξυπνων καρτών στην Ευρώπη και την Βόρεια Αμερική, μπορεί μεν ο αριθμός των καρτών που κυκλοφορούν ανά έτος στην αγορά να είναι πολύ μεγαλύτερος στον Ευρωπαϊκό χώρο και να προβλέπεται να παραμένει ανώτερος στα επόμενα χρόνια, ο ρυθμός ανάπτυξης όμως της αγοράς στη Βόρεια Αμερική παρουσιάζεται σημαντικά μεγαλύτερος από τον αντίστοιχο ευρωπαϊκό, ιδιαίτερα από το 2002 και έπειτα.

ΠΡΟΒΛΕΨΗ ΕΞΥΠΝΩΝ ΚΑΡΤΩΝ ΑΝΑ ΠΕΡΙΟΧΗ (TSI International)



	1999	2000	2001	2002	2003	2004	2005
ΣΥΝΟΛΟ	1800	2300	3000	3900	4700	5500	6100
ΕΥΡΩΠΗ	1124	1371	1535	1720	1926	2157	2318
Β.ΑΜΕΡΙΚΗ	36	50	76	164	310	530	915

Σχήμα 2.4.5

Συγκεκριμένα, σύμφωνα με αναλύσεις, ο ετήσιος ρυθμός ανάπτυξης της αγοράς έξυπνων καρτών στην Βόρεια Αμερική, εκτιμάται ότι θα αγγίξει το 50% στο 2004, γεγονός που δείχνει τη σημαντική δυναμική της τεχνολογίας έξυπνων καρτών στην αγορά αυτή.
Σε γενικότερη κλίμακα, παρατηρούμε ότι ο ρυθμός ανάπτυξης της παγκόσμιας αγοράς έξυπνων καρτών είναι έντονα θετικός, παρουσιάζοντας έτσι την εφαρμογή της τεχνολογίας των έξυπνων καρτών ως μία καλπάζουσα τάση που εδραιώνεται όλο και περισσότερο στη σύγχρονη πραγματικότητα, καθώς και στη συνείδηση των καταναλωτών και των επιχειρήσεων. Ενδεικτικό είναι το γεγονός ότι εκτιμάται πως οι παγκόσμιες πωλήσεις έξυπνων καρτών θα φτάσουν τα 8 δισεκατομμύρια δολάρια το 2004.
Υπάρχουν σαφώς αρκετά βήματα τα οποία πρέπει να γίνουν για να υπάρχει ακόμα μεγαλύτερη ανάπτυξη της αγοράς των έξυπνων καρτών, βήματα που έχουν σχέση με τη γνωριμία και εμπιστοσύνη του κοινού με τη νέα αυτή τεχνολογία, καθώς και με την ανάπτυξη κατάλληλης υποδομής σε πολλούς τομείς. Υπάρχει η βεβαιότητα όμως ότι η απόδοση των υπάρχουσων εφαρμογών θα αποτελέσουν τον καλύτερο "πωλητή" των έξυπνων καρτών στην αγορά.

2.5 Εφαρμογές Έξυπνων Καρτών

Οι έξυπνες κάρτες βοηθούν τις επιχειρήσεις να εξελιχθούν και να διευρύνουν τα προϊόντα και τις υπηρεσίες τους σε μία συνεχώς μεταβαλλόμενη παγκόσμια αγορά. Λόγω της επεξεργαστικής δυνατότητας που έχουν μέσω του ενσωματωμένου μικροτσίπ, χρησιμοποιούνται παγκοσμίως για ένα μεγάλο εύρος καθημερινών εργασιών αλλά και προηγμένων εφαρμογών, την πλειονότητα των οποίων θα αναπτύξουμε παρακάτω.
Οι εκάστοτε εταιρίες, σχεδιάζοντας εφαρμογές και προγράμματα, μπορούν να δουν και να χρησιμοποιήσουν τις έξυπνες κάρτες ως:

- Μέσα Πληρωμής: οι έξυπνες κάρτες εξασφαλίζουν ασφαλείς χρεωστικές και πιστωτικές συναλλαγές, με μηχανισμούς που να προστατεύουν από κακόβουλες επιθέσεις. Συγχρόνως, αποτελούν για τις εταιρίες μία νέα καθαρή πηγή εσόδων αφού τις απαλλάσσουν από το πάγιο κόστος συναλλαγής το οποίο συνόδευε κάθε συναλλαγή με τις γνωστές τραπεζικές κάρτες (credit/debit cards) όπως και από τις πιθανές απώλειες εσόδων λόγω χαμένων / κλεμμένων καρτών.
- Εργαλεία Πρόσβασης: οι έξυπνες κάρτες υποστηρίζουν λειτουργίες κρυπτογράφησης, πιστοποίησης, εξουσιοδότησης, επεξεργασίας και αποθήκευσης πληροφοριών οι οποίες καθιστούν δυνατή την ασφαλή διεξαγωγή οικονομικών συναλλαγών και ανταλλαγή πληροφορίας σε on-line/off-line περιβάλλοντα. Έτσι γίνονται ιδανικές για τον έλεγχο πρόσβασης στο Διαδίκτυο και για εφαρμογές όπως το home banking.
- Διαχειριστές Πληροφοριών: λόγω της επεξεργαστικής και αποθηκευτικής τους δύναμης όσο αφορά πληροφορίες, οι έξυπνες κάρτες μπορούν να χρησιμοποιηθούν ως ένα κινητό ηλεκτρονικό αρχείο που μπορεί να μεταφέρει δεδομένα όπως χρήσιμα τηλέφωνα, στοιχεία του λογαριασμού του κατόχου, πόντους προγραμμάτων εμπιστοσύνης λιανικής πώλησης ή ακόμα και τον ιατρικό φάκελο του χρήστη.
- Εργαλεία Προώθησης: οι έξυπνες κάρτες μπορούν να λειτουργήσουν ως προϊόντα προώθησης μίας εταιρίας αφού υπηρεσίες όπως εκπτωτικές προσφορές, προγράμματα εμπιστοσύνης, ηλεκτρονικά κουπόνια και δωροεπιταγές μπορούν κάλλιστα να αποθηκεύουν και να επεξεργάζονται με ασφάλεια τα εκάστοτε στοιχεία τους στις έξυπνες κάρτες.
- Συστήματα Προσωποποιημένων Υπηρεσιών: με τις δυνατότητες αποθήκευσης, επεξεργασίας και κωδικοποίησης δεδομένων που υποστηρίζουν οι έξυπνες κάρτες, μπορούν να κρατούν σημαντικά στοιχεία για το κάτοχό τους και να χρησιμεύουν για την παροχή προσωποποιημένων υπηρεσιών από διάφορες εταιρίες.

Με μία ή περισσότερες από τις μορφές που αναφέρθηκαν παραπάνω, οι έξυπνες κάρτες έχουν χρησιμοποιηθεί σε ποικίλες εφαρμογές τις οποίες θα παραθέσουμε ακολούθως πιο διεξοδικά.

2.5.1 Τηλεφωνικές Κάρτες

Οι τηλεφωνικές κάρτες προπληρωμένης αξίας αποτελούν μία από τις πρώτες εφαρμογές έξυπνων καρτών. Διαδεδομένη χρήση τους ξεκίνησε το 1986 από τη Γαλλία και έκτοτε επεκτάθηκε ραγδαία και σε άλλες χώρες. Σε περισσότερες από 100 χώρες παγκοσμίως οι τηλεφωνικοί κερματοδέκτες σε δημόσιους και κοινόχρηστους χώρους, έχουν αντικατασταθεί από καρτοτηλέφωνα και τα κέρματα, ως μέσο πληρωμής των τηλεφωνικών υπηρεσιών, από τις τηλεφωνικές έξυπνες κάρτες.
Αγοράζονται από τους καταναλωτές έναντι συγκεκριμένου αντιτίμου (3€, 6€ και 18€ για την Ελληνική αγορά) και περιέχουν συγκεκριμένο αριθμό μονάδων (ανάλογα με το ποσό αγοράς τους), οι οποίες μειώνονται με κάθε κλήση. Οι τηλεκάρτες είναι έξυπνες κάρτες που ανήκουν στην κατηγορία των καρτών μνήμης (memory cards).
Μεγάλης κλίμακας προγράμματα εφαρμόζονται σε χώρες όπως η Γερμανία, η Γαλλία, η Αγγλία, η Βραζιλία, το Μεξικό και η Κίνα, ενώ στην Ελλάδα το δίκτυο καρτοτηλεφωνίας περιλαμβάνει 70.000 καρτοτηλέφωνα σε όλη τη χώρα.

2.5.2 Κινητή Τηλεφωνία (GSM)

Οι έξυπνες κάρτες χρησιμοποιούνται ευρέως ως κάρτες SIM (Security Identity Module) στην κινητή τηλεφωνία GSM (Global System for Mobile communications). Η κάρτα SIM περιέχει πληροφορίες ασφαλείας και συνδρομητικά στοιχεία. Μπορεί είτε να εισάγεται στη συσκευή είτε να βρίσκεται ενσωματωμένη σε αυτή και με την ενεργοποίησή της το τηλέφωνο προσωποποιείται ως προς το χρήστη και φορτώνει στοιχεία όπως το νούμερό του στο δίκτυο, πληροφορίες κοστολόγησης και πρόσφατα κληθέντες αριθμούς. Η κάρτα μπορεί να μεταφέρεται από συσκευή σε συσκευή αφού περιέχει τα στοιχεία του συνδρομητή τα οποία προστατεύονται από ειδικό κωδικό (PIN).
Οι παροχείς κινητής τηλεφωνίας κερδίζουν από τη μείωση των περιπτώσεων απάτης και μη έγκυρης χρήσης λόγω της αυξημένης ασφάλειας που προσφέρουν οι έξυπνες κάρτες. Με την έλευση προηγμένων υπηρεσιών κινητής τηλεφωνίας όπως η πρόσβαση στο Διαδίκτυο (web browsing), το ηλεκτρονικό ταχυδρομείο και άλλες υπηρεσίες πληροφοριών, οι παροχείς βασίζονται στις έξυπνες κάρτες να δράσουν ως μηχανισμοί ασφαλείας για τις υπηρεσίες αυτές.
Στη παγκόσμια αγορά, το 1994 πωλήθηκαν περισσότερες από 9.000.000 έξυπνες κάρτες κινητής τηλεφωνίας ενώ πλέον τα κινητά τηλέφωνα που χρησιμοποιούν τις έξυπνες κάρτες ως κάρτες SIM ξεπερνούν τα 300.000.000.

2.5.3 Συνδρομητική Τηλεόραση

Σχεδόν κάθε μικρό πιάτο δορυφορικής τηλεόρασης στις Ηνωμένες Πολιτείες χρησιμοποιεί μία έξυπνη κάρτα ως αφαιρέσιμο στοιχείο ασφαλείας και πληροφοριών για το συνδρομητή. Οι έξυπνες κάρτες λειτουργούν ως μία προπληρωμένη εφαρμογή, όπως και οι τηλεκάρτες που αναφέρθηκαν παραπάνω, και περιέχουν πληροφορίες εξουσιοδότησης και κοστολόγησης που αντιστοιχούν στον συνδρομητή-κάτοχο. Κυρίως περιέχουν ειδικά "κλειδιά" (keys) τα οποία χρειάζονται για να μπορεί ο συνδρομητής να δει την κωδικοποιημένη μετάδοση.
Η κάρτα συνδρομητικής τηλεόρασης μπορεί να χρησιμοποιηθεί σε οποιοδήποτε χώρο έχει την κατάλληλη υποδομή και δε συνδέεται αποκλειστικά με τη συσκευή αλλά με το συνδρομητή. Έτσι ένας συνδρομητής μπορεί με την κάρτα του να παρακολουθήσει το πρόγραμμα της συνδρομητικής τηλεόρασης στο σπίτι του αλλά και σε ένα ξενοδοχείο. Ένα μεγάλο προτέρημα της χρήσης έξυπνων καρτών σε αυτή την εφαρμογή είναι τα στοιχεία προσωποποίησης που περιέχουν και προσδιορίζουν-φιλτράρουν το μέρος της μετάδοσης που θα λαμβάνει ο συνδρομητής. Έτσι οι γονείς μπορούν να παρέχουν στα παιδιά κάρτες

συνδρομητικής τηλεόρασης που αποκλείουν την πρόσβαση των παιδιών σε προγράμματα ακατάλληλα.
Στην Αμερική, πάνω από 4.000.000 κάρτες συνδρομητικής τηλεόρασης χρησιμοποιούνται και εκατομμύρια ακόμα διατίθενται στην Ευρώπη και την Ασία.

2.5.4 Συγκοινωνίες

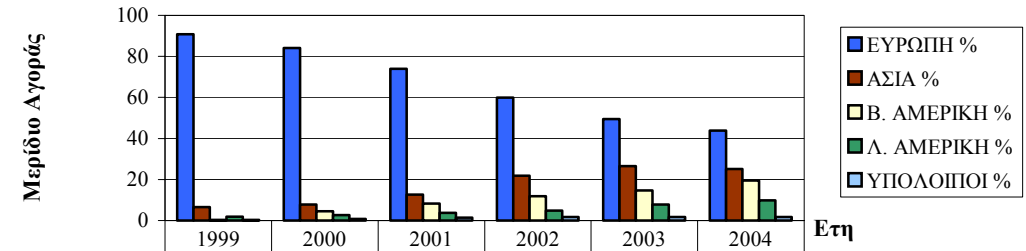
Οι έξυπνες κάρτες χρησιμοποιούνται σε μεγάλο βαθμό ως "εισιτήρια", στα μέσα μαζικής μεταφοράς, στα πάρκινγκ και τα διόδια. Συνήθως χρησιμοποιούνται contactless (χωρίς επαφή) κάρτες, που διευκολύνουν και επιταχύνουν τη διαδικασία. Πωλούνται ως κάρτες προπληρωμένης αξίας, όπως και οι τηλεφωνικές. Σχεδιάζονται για χρήση σε μέσα μαζικής μεταφοράς όπως λεωφορεία και τρένα, όπως επίσης και στα διόδια, κάνοντας τη διαδικασία έκδοσης εισιτηρίων πολύ πιο γρήγορη και εύκολη.
Η αξία του εισιτηρίου, ανάλογα με το πεδίο εφαρμογής, αφαιρείται από το ποσό που είναι αποθηκευμένο στη κάρτα κάθε φορά που ο κάτοχος περνάει από την ειδική συσκευή ανάγνωσης / γραφής και μπορεί να επαναφορτώνεται στο κατάλληλο σημείο πώλησης. Αυτός ο τρόπος παρέχει ευκολία στους καταναλωτές και με τη χρήση ειδικών "επιβραβευτικών" προγραμμάτων στις συγκοινωνίες, αυξάνει τη χρήση των μέσων μαζικής μεταφοράς, προσελκύοντας περισσότερους καταναλωτές.
Ειδικά στην περίπτωση των διοδίων, η χρήση των έξυπνων καρτών συμβάλει πολύ στην εξυπηρέτηση του κοινού αφού επιτρέπει την συλλογή διοδίων χωρίς τη παρεμπόδιση της κυκλοφορίας. Στις περιοχές συλλογής των διοδίων υπάρχουν ειδικές συσκευές ανάγνωσης και πάνω στα διερχόμενα οχήματα υπάρχουν ειδικές συσκευές για τις έξυπνες κάρτες ούτως ώστε με τη διέλευση του οχήματος από το σημείο συλλογής, η χρέωση να γίνεται αυτόματα χωρίς να χρειάζεται να δημιουργούνται ουρές.

2.5.5 Banking / E-purse

Ο οικονομικός και τραπεζικός χώρος ήταν από τους πρώτους που υιοθέτησαν την τεχνολογία των έξυπνων καρτών σε πολλές χώρες παγκοσμίως. Κάθε Γαλλική χρεωστική κάρτα VISA έχει πλέον μικροτσίπ. Χώρες όπως η Πορτογαλία και η Σιγκαπούρη έχουν εισάγει προγράμματα ηλεκτρονικού πορτοφολιού στα εθνικά τραπεζικά δίκτυά τους.
Οι έξυπνες κάρτες χρησιμοποιούνται από τις τράπεζες είτε ως πιστωτικές είτε ως χρεωστικές, εις αντικατάσταση των υπάρχουσων καρτών μαγνητικής ταινίας. Οι πιστωτικές κάρτες δίνουν πληροφορίες για το πιστωτικό λογαριασμό του κατόχου ο οποίος θα χρεωθεί μετά από μία αγορά και είναι ένας τρόπος να δοθεί μία "πίστωση χρόνου" στον κάτοχό της για την πληρωμή, ένας τρόπος άτοκου (μέχρι ενός ορισμένου χρονικού διαστήματος) δανεισμού. Οι χρεωστικές κάρτες δίνουν πληροφορίες για τον καταθετικό λογαριασμό του κατόχου της κάρτας και η οποιαδήποτε αγορά χρεώνεται κατευθείαν στο λογαριασμό, είναι δηλαδή μία άμεση πληρωμή χωρίς μετρητά.
Η πιστοποίηση της ταυτότητας του κατόχου μίας κλασικής πιστωτικής κάρτας γίνεται με παρατήρηση της υπογραφής του και της ταυτότητας του, ενώ στην περίπτωση των συνηθισμένων χρεωστικών καρτών (debit cards) υπάρχει ένας κωδικός (PIN) που επαληθεύεται όμως μόνο on-line. Οι έξυπνες κάρτες έρχονται να αλλάξουν αυτό το τοπίο αφού ο κωδικός του κατόχου είναι αποθηκευμένος στην ίδια την κάρτα και προστατεύεται όπως και επαληθεύεται με ασφαλείς διαδικασίες που παρέχει η κάρτα.
Έτσι οι κάρτες αυτές γίνονται πιο ασφαλείς και για τις καινούριες τραπεζικές υπηρεσίες που παρέχονται στους πελάτες, όπως το web-banking, αυξάνοντας την ποιότητα εξυπηρέτησης των πελατών. Συγχρόνως, μειώνεται το λειτουργικό κόστος των πιστωτικών ιδρυμάτων αφού εργασίες που θα απαιτούσαν καθημερινή ανθρώπινη εργασία γίνονται με ηλεκτρονικό τρόπο.
Η τεχνολογία των έξυπνων καρτών ευνοεί και τους πωλητές λιανικής αφού με την ασφάλεια που παρέχει μειώνει το κόστος από απώλειες λόγω απατών ή λαθών.
Στο διάγραμμα που ακολουθεί στο Σχήμα 2.5.1 βλέπουμε πώς κινείται η αγορά των έξυπνων τραπεζικών καρτών παγκοσμίως και παρατηρούμε πως το μεγαλύτερο μερίδιο κατέχει

σταθερά η Ευρώπη, το μέγεθος όμως του μεριδίου ελαττώνεται με την πάροδο των ετών, καθώς η τεχνολογία αυτή εξαπλώνεται στις άλλες ηπείρους.

ΑΝΑΛΥΣΗ ΜΕΡΙΔΙΩΝ ΑΓΟΡΑΣ ΤΡΑΠΕΖΙΚΩΝ ΕΞΥΠΝΩΝ ΚΑΡΤΩΝ



	1999	2000	2001	2002	2003	2004
ΕΥΡΩΠΗ %	90,82	84,09	73,92	59,85	49,49	43,82
ΑΣΙΑ %	6,61	7,78	12,63	21,85	26,46	25,17
Β. ΑΜΕΡΙΚΗ %	0,37	4,59	8,31	11,79	14,64	19,5
Λ. ΑΜΕΡΙΚΗ %	1,84	2,69	3,67	4,79	7,79	9,81
ΥΠΟΛΟΙΠΟΙ %	0,37	0,85	1,47	1,72	1,65	1,7

Σχήμα 2.5.1

Στο διάγραμμα που ακολουθεί στο Σχήμα 2.5.1 βλέπουμε πώς κινείται η αγορά των έξυπνων τραπεζικών καρτών παγκοσμίως και παρατηρούμε πως το μεγαλύτερο μερίδιο κατέχει σταθερά η Ευρώπη, το μέγεθος όμως του μεριδίου ελαττώνεται με την πάροδο των ετών, καθώς η τεχνολογία αυτή εξαπλώνεται στις άλλες ηπείρους.
Τέλος, το ηλεκτρονικό πορτοφόλι (e-purse/e-wallet) είναι ένας ακόμα τρόπος κατοχής ηλεκτρονικού χρήματος (κάτι αντίστοιχο με τις κάρτες προπληρωμένης αξίας όπως οι τηλεφωνικές κάρτες), με τη διαφορά ότι μπορεί να γίνει και πίστωση και χρέωση στην κάρτα, δίνοντας έτσι μεγαλύτερες δυνατότητες στο κάτοχο. Το ηλεκτρονικό πορτοφόλι προσφέρει στους κατόχους ευκολία χρήσης και ασφάλεια και προτείνεται κυρίως σε εφαρμογές που έχουν σχέση με το Διαδίκτυο.

2.5.6 Προγράμματα Εμπιστοσύνης

Πολλές εταιρίες χρησιμοποιούν έξυπνες κάρτες σε προγράμματα εμπιστοσύνης για να εντοπίζουν και να δίνουν κίνητρα αγοράς στους τακτικούς πελάτες.
Οι κάρτες αυτές είναι συνήθως κάρτες επαφής (contact cards) που μαζεύουν πόντους από αγορά προϊόντων ή υπηρεσιών από συγκεκριμένο πωλητή λιανικής. Οι πόντοι αυτοί ανταλλάσσονται με πιστώσεις, με βραβεία ή και άλλα στοιχεία. Μέσω του συστήματος αυτού, οι εταιρίες λιανικής πώλησης μπορούν για πρώτη φορά να έχουν λεπτομερή στοιχεία για τις προτιμήσεις των πελατών.
Ειδικά για μεγάλες αλυσίδες πωλήσεων που διαχειρίζονται προγράμματα εμπιστοσύνης σε διαφορετικά αντικείμενα (όπως τα πολυκαταστήματα), πληροφορίες για τον πελάτη και τις προτιμήσεις του διαχειρίζονται και αποθηκεύονται κεντρικά σε μία έξυπνη κάρτα που κατέχει όλες τις πληροφορίες και δίνει την δυνατότητα στις εταιρίες λιανικής πώλησης να κάνουν σωστό σχεδιασμό της πολιτικής προσέγγισης των πελατών. Έτσι παρέχεται μεγαλύτερη ποιότητα στην εξυπηρέτηση των πελατών και σαφώς τα έσοδα για τις εταιρίες είναι μεγαλύτερα.

2.5.7 Έλεγχος Πρόσβασης

Σημαντική δραστηριότητα έχει παρουσιαστεί από μεγάλες εταιρίες και οργανισμούς, καθώς και από κυβερνήσεις για την εισαγωγή καινούριων συστημάτων ελέγχου πρόσβασης, τα οποία ελέγχουν την ταυτότητα και τα επίπεδα εξουσιοδότησης κάποιου πριν του δοθεί πρόσβαση φυσική (σε κάποιο κτίριο για παράδειγμα) ή λογική (π.χ. σε εμπιστευτικές πληροφορίες σε δίκτυα).
Όσο περισσότερο οι ανωτέρω φορείς χρησιμοποιούν δίκτυα τοπικά και μη, και το Διαδίκτυο για να αποθηκεύουν και να κοινοποιούν σημαντικές πληροφορίες σε αυτούς που τις χρειάζονται, τόσο περισσότερο επεκτείνεται η χρήση των έξυπνων καρτών σε αυτό το τομέα. Μεγάλες εμπορικές επιχειρήσεις όπως η Sun και η Microsoft, εφαρμόζουν συστήματα ελέγχου πρόσβασης που βασίζονται στη τεχνολογία των έξυπνων καρτών για να διαχειριστούν καθολικά την πρόσβαση εργαζομένων σε συγκεκριμένες πηγές.
Σε αυτή την κατεύθυνση, οι έξυπνες κάρτες προσφέρουν ταχύτητα πρόσβασης και μειωμένα κόστη συντήρησης (ειδικά στην περίπτωση του ασύρματου ελέγχου πρόσβασης), πολλαπλά επίπεδα ταυτοποίησης και πλήθος μεθόδων κρυπτογράφησης και πιστοποίησης, καθώς και ευελιξία στη χρησιμοποίηση διαφορετικών καρτών λόγω σταθερών προτύπων που ακολουθούνται.

2.5.8 Υγεία

Οι ιατρικές έξυπνες κάρτες χρησιμοποιούνται κατά κόρον σε πολλές χώρες παγκοσμίως. Η τάση των τελευταίων ετών είναι η μεταφορά από συστήματα πληροφοριών ιατρικής φροντίδας που βασίζονται σε χαρτιά και έγγραφα σε ηλεκτρονικά συστήματα τα οποία προστατεύουν τα προσωπικά δεδομένα των κατόχων των καρτών.
Οι έξυπνες ιατρικές κάρτες αποθηκεύουν πολλών ειδών ιατρικές πληροφορίες που αφορούν τον κάτοχο, όπως λεπτομέρειες για αλλεργίες και χρόνιες ασθένειες. Μπορούν να έχουν αποθηκευμένες παλιές, επαναλαμβανόμενες ή και νέες συνταγές ιατρών καθώς και διάφορες θεραπείες στις οποίες ο κάτοχος έχει υποβληθεί. Για τους ασθενείς, αυτός ο τρόπος αυξάνει την ποιότητα της παρεχόμενης ιατρικής φροντίδας, ενώ για τους παροχείς της ιατρικής βοήθειας, μειώνονται τα λειτουργικά κόστη και αυξάνεται η αποτελεσματικότητα της δράσης τους. Το κυριότερο είναι ότι με αυτή τη μέθοδο, σώζονται πραγματικά ζωές, αφού το ηλεκτρονικό ιατρικό ιστορικό του ασθενή είναι εύκολα προσβάσιμο και μπορεί να μεταφέρεται. Πολλές χώρες με εθνικά προγράμματα ιατρικής φροντίδας χρησιμοποιούν συστήματα έξυπνων καρτών, το μεγαλύτερο των οποίων λειτουργεί στη Γερμανία όπου πάνω από 80.000.000 κάρτες έχουν μοιραστεί σε κάθε άτομο στη Γερμανία και την Αυστρία.

2.5.9 Πανεπιστημιακοί χώροι

Πανεπιστήμια και σχολές σε πολλές χώρες χρειάζονται ένα τρόπο αναγνώρισης της ταυτότητας των εργαζομένων και των φοιτητών και χρησιμοποιούν τη τεχνολογία των έξυπνων καρτών για αυτό το σκοπό. Οι περισσότεροι από τους κατόχους αυτών των καρτών έχουν πρόσβαση σε συγκεκριμένες πληροφορίες, εξοπλισμό και τμήματα, ανάλογα με τις συνθήκες και τα χαρακτηριστικά της θέσης τους.
Έξυπνες κάρτες πολλαπλών διεργασιών περιέχουν τα στοιχεία ταυτότητας με χαρακτηριστικά πρόσβασης ενώ επίσης μπορούν να αποθηκεύουν αξία (χρήματα) για χρήση σε διάφορους χώρους εντός των πανεπιστημίων, όπως τα κυλικεία ή κάποια καταστήματα. Γίνονται έτσι ένα εύκολο εργαλείο για τον εργαζόμενο και τον φοιτητή ο οποίος με μία κάρτα μπορεί να κινηθεί όπου επιθυμεί και να καλύψει τις ανάγκες του στο συγκεκριμένο χώρο.
Για παράδειγμα, το Πανεπιστήμια της Florida, έχει εκδώσει 40.000 κάρτες οι οποίες εξυπηρετούν λειτουργίες προσωπικής ταυτοποίησης, τραπεζικών συναλλαγών και πρόσβασης σε σπουδαστικούς χώρους για τους φοιτητές ενώ ταυτόχρονα λειτουργούν ως κάρτες προπληρωμένης αξίας για υπηρεσίες σίτισης, τηλεφωνίας και μετακίνησης μέσα στο Πανεπιστήμιο.

3

Τεχνολογία Έξυπνων Καρτών

Για να κατανοήσουμε τα τεχνικά χαρακτηριστικά των έξυπνων καρτών χρειάζεται να εξηγήσουμε περιληπτικά κάποιες βασικές ηλεκτρονικές έννοιες.

3.1 Βασικές Έννοιες

3.1.1 Μικροτσίπ

Το μικροτσίπ είναι ένα σύνολο πολύπλοκων και πολύ μικρών στοιχείων που μπορούν να αποθηκεύσουν υπολογιστική μνήμη ή να παρέχουν το λογικό κύκλωμα για μικροεπεξεργαστές. Κατασκευάζεται από λεπτά κυκλικά επίπεδα δισκία πυριτίου τα οποία επεξεργάζονται ως προς το μέγεθος και συνδέονται με κυκλώματα και ηλεκτρονικές συσκευές. Οι συσκευές αυτές χρησιμοποιούν τεχνολογία ημιαγωγών μετάλλου-οξειδίου. Το τρέχον στάδιο της ολοκλήρωσης των μικροτσίπ είναι το γνωστό VLSI (Very Large-Scale Integration). Το μικροτσίπ καλείται αλλιώς και ολοκληρωμένο κύκλωμα (IC).

3.1.2 VLSI

VLSI είναι όπως είπαμε, το παρόν επίπεδο "μικρογραφίας" μικροτσίπ υπολογιστών και αναφέρεται σε μικροτσίπ που περιέχουν εκατοντάδες χιλιάδες τρανζίστορ, δίνοντας έτσι τη δυνατότητα να παραχθούν μνήμες (RAM, ROM) ή μονάδες επεξεργασίας (CPU) σε ένα και μόνο τσιπ.

3.1.3 Μνήμη

Μνήμη είναι το ηλεκτρονικό μέρος αποθήκευσης εντολών και πληροφοριών στις οποίες ένας επεξεργαστής μπορεί εύκολα να έχει πρόσβαση. Σε έναν υπολογιστή που βρίσκεται σε κανονική λειτουργία, η μνήμη περιέχει κύρια μέρη του λειτουργικού συστήματος του υπολογιστή και πολλά ή όλα τα προγράμματα εφαρμογών και τα δεδομένα που αυτά χρησιμοποιούν.

3.1.3.1 RAM

RAM (Random Access Memory - Μνήμη Τυχαίας Προσπέλασης) είναι μνήμη που περιέχεται σε ένα ή περισσότερα μικροτσίπ κοντά στον μικροεπεξεργαστή, έχει μικρό φυσικό μέγεθος και μικρή γενικά χωρητικότητα σε σχέση με άλλα αποθηκευτικά μέσα όπως ο σκληρός δίσκος και το CD-ROM. Είναι όμως πολύ πιο γρήγορη και άμεση η πρόσβαση στα δεδομένα της και οι διαδικασίες ανάγνωσης και εγγραφής γίνονται ταχύτερα (χρόνος πρόσβασης σε τάξη nanoseconds). Τα όποια δεδομένα και στοιχεία συστήματος αποθηκεύονται στη RAM, βρίσκονται εκεί μόνο όσο ο υπολογιστής λειτουργεί και χάνονται όταν το ρεύμα αφαιρεθεί.
Ο όρος "τυχαία προσπέλαση" αναφέρεται στο ότι η πρόσβαση σε αποθηκευμένη πληροφορία δεν γίνεται ακολουθιακά αλλά άμεσα.

3.1.3.2 ROM

Η ROM (Read Only Memory - Μνήμη Μόνο Ανάγνωσης) είναι μνήμη στην οποία δεν μπορούν να γίνουν εγγραφές, αλλά μόνο ανάγνωση. Η ROM περιέχει τα στοιχεία προγραμματισμού που επιτρέπουν σε ένα υπολογιστή να ξεκινήσει και δεν χάνει τα δεδομένα της όταν ο υπολογιστής κλείσει. Όταν η ROM συντηρείται με μπαταρία, αυτή είναι μία μικρή μεγάλης διάρκειας μπαταρία. Υπάρχουν και ROM που "χτίζονται" μία φορά κατά τη κατασκευή τους (firmware). Το κόστος μίας ROM μνήμης είναι μεγαλύτερο από αυτό της RAM.

3.1.3.3 EEPROM

Η EEPROM (Electrically Erasable Programmable Read-Only Memory) είναι μνήμη ROM που μπορεί να μεταβληθεί από τον χρήστη, δηλαδή μπορεί να σβηστεί και να επαναπρογραμματιστεί με την εφαρμογή υψηλότερης από την κανονική τάσης. Το κύριο χαρακτηριστικό της είναι ότι δεν μπορεί να σβηστεί και να προγραμματιστεί σε κομμάτια αλλά μόνο στην ολότητά της, και αυτό μπορεί να γίνει χωρίς να μετακινηθεί από τον υπολογιστή. Έχει περιορισμένη διάρκεια ζωής αφού επιτρέπει περιορισμένο αριθμό επαναπρογραμματισμών ο οποίος φτάνει σε δεκάδες ή εκατοντάδες χιλιάδες φορές.

3.1.3.4 FLASH

Η μνήμη FLASH είναι ένας τύπος μνήμης EEPROM με τη βασική διαφορά να έγκειται στη ταχύτητα της διαγραφής και επαναπρογραμματισμού του περιεχομένου της. Συγκεκριμένα, ενώ στην EEPROM μπορεί να διαγραφεί ένα byte τη φορά, στη μνήμη FLASH μπορούν να σβηστούν ολόκληρα μπλοκ από byte, αυξάνοντας έτσι την ταχύτητα διαγραφής και προγραμματισμού της μνήμης αυτής.

3.1.4 Επεξεργαστής

Ο επεξεργαστής είναι η λογική κυκλωματική συνδεσμολογία που επεξεργάζεται και ανταποκρίνεται σε βασικές εντολές που οδηγούν ένα υπολογιστή. Ο όρος "επεξεργαστής" έχει γενικά αντικαταστήσει τον όρο Κεντρική Μονάδα Επεξεργασίας (CPU). Ο επεξεργαστής σε ένα προσωπικό υπολογιστή ή ενσωματωμένος σε μικρές συσκευές καλείται συχνά "μικροεπεξεργαστής".

3.1.5 Μικροεπεξεργαστής

Ο μικροεπεξεργαστής είναι ένας επεξεργαστής υπολογιστή σε μικροτσίπ. Μερικές φορές καλείται "λογικό τσιπ". Είναι σχεδιασμένος για να εκτελεί αριθμητικές και λογικές διεργασίες που χρησιμοποιούν μικρές περιοχές καταγραφής αριθμών που καλούνται καταχωρητές. Τυπικές τέτοιες διεργασίες είναι η πρόσθεση, αφαίρεση ή σύγκριση δύο αριθμών ή η μεταφορά αριθμών από μία περιοχή σε άλλη. Αυτές οι διεργασίες είναι αποτέλεσμα ενός συνόλου εντολών που αποτελούν μέρος του σχεδιασμού του μικροεπεξεργαστή.

3.1.6 Εντολές

Μία εντολή είναι μία διαταγή που δίνεται σε ένα επεξεργαστή από ένα υπολογιστικό πρόγραμμα. Στο πιο χαμηλό επίπεδο (επίπεδο μηχανής), κάθε εντολή είναι μία ακολουθία από μηδενικά και άσσους που περιγράφουν μία φυσική διεργασία που θα εκτελέσει ο υπολογιστής ή η ταυτότητα των καταχωρητών που θα χρησιμοποιηθούν για την εντολή. Στη γλώσσα assembly ενός υπολογιστή μία δήλωση αντιστοιχεί σε μία εντολή μικροεπεξεργαστή ενώ σε γλώσσες υψηλότερου επιπέδου, αντιστοιχεί σε σετ εντολών.

3.1.7 Private Key

Ένα private (secret) key είναι μία κλείδα (ένας κωδικός) που χρησιμοποιείται για κρυπτογράφηση και αποκρυπτογράφηση μηνυμάτων μέσα στα πλαίσια της επικοινωνίας δύο πλευρών. Το κλειδί αυτό προφανώς είναι κοινώς γνωστό στις δύο πλευρές που επικοινωνούν και συνεπακόλουθα αν μία από τις δύο πλευρές το χάσει ή το κλειδί κλαπεί, η ασφάλεια και η μυστικότητα της επικοινωνίας θα χαθεί.

3.1.8 Public Key

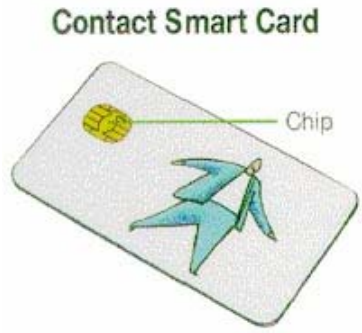
Ένα public key είναι πάλι ένας κωδικός, ο οποίος όμως σε συνδυασμό με ένα private key το οποίο παράγεται από το public key, χρησιμοποιείται για ασφαλή κρυπτογράφηση μηνυμάτων και δημιουργία ψηφιακών υπογραφών. Η συνδυαστική χρήση public και private keys είναι γνωστή ως ασύμμετρη κρυπτογράφηση.

3.2 Τύποι καρτών

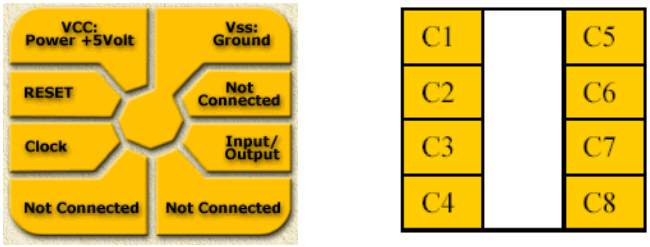
Υπάρχουν διάφορες κατηγορίες στις οποίες χωρίζονται οι έξυπνες κάρτες, ανάλογα με το τύπο της διεπαφής τους (interface) με τον έξω κόσμο ή ανάλογα με το τύπο του μικροτσίπ.

3.2.1 Contact Cards

Οι κάρτες με επαφή χρειάζεται να εισαχθούν μέσα σε ένα card reader (αναγνώστη καρτών) ο οποίος θα έχει άμεση επαφή με το λεπτό μεταλλικό πιάτο που βρίσκεται στην επιφάνεια της κάρτας (κάτω από το οποίο βρίσκεται το μικροτσίπ), για να επιτευχθεί η επικοινωνία μεταξύ τους μέσω αυτών των ηλεκτρικών επαφών και για να πάρουν ρεύμα.



Η επικοινωνία συνίσταται στην ανταλλαγή εντολών, δεδομένων και πληροφοριών κατάστασης.. Οι επαφές πρέπει να βρίσκονται σε αυστηρά καθορισμένο επίπεδο και να μην υπάρχει μεταξύ τους μη αγώγιμο υλικό.



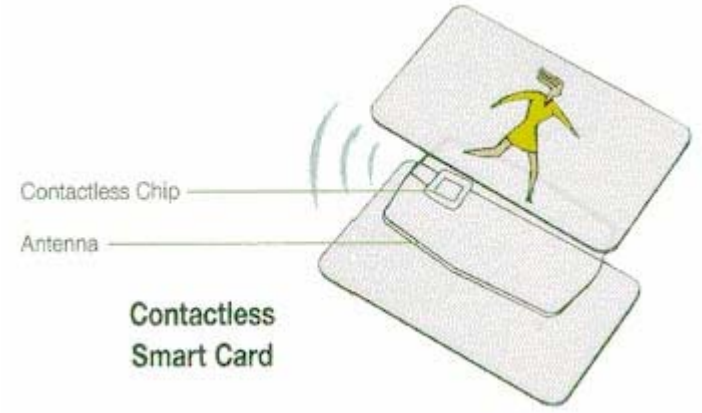
Σχήμα 3.2.1 - Επαφές κάρτας

Στο παραπάνω σχήμα φαίνονται οι ηλεκτρικές επαφές στην επιφάνεια της κάρτας, των οποίων η θέση καθορίζεται από παγκόσμια πρότυπα (ISO 7816-1 και ISO 7816-2).

Οι λειτουργίες των επαφών αναλύονται ακολούθως. Η επαφή C1 αντιστοιχεί στη Vcc που είναι η τάση τροφοδοσίας και συνήθως είναι στα 5V. Η επαφή C2 αντιστοιχεί στο Reset που είναι η γραμμή σήματος η οποία χρησιμεύει για να αρχικοποιήσει την κατάσταση του ολοκληρωμένου κυκλώματος μετά τη τροφοδοσία της κάρτας. C3 είναι η επαφή που συνδέεται με το σήμα ρολογιού (Clock) το οποίο οδηγεί την λογική του ολοκληρωμένου και συγχρόνως χρησιμοποιείται ως σημείο αναφοράς για τη σειριακή σύνδεση επικοινωνίας. Οι επαφές C4 και C8 δεν χρησιμοποιούνται. Η επαφή C5 αντιστοιχεί στο GND (γείωση) που είναι σημείο μηδενικού δυναμικού και βάση του οποίου μετριέται η τάση τροφοδοσίας. C6 είναι η επαφή που αντιστοιχεί στη Vpp, στην υψηλή δηλαδή τάση που χρησιμοποιείται για να προγραμματιστεί η EEPROM μνήμη. Τέλος, η επαφή C7 αντιστοιχεί στη σειριακή θύρα εισόδου-εξόδου η οποία χρησιμοποιείται για την ανταλλαγή και τη λήψη εντολών και πληροφοριών από τον εξωτερικό κόσμο.
Οι κάρτες με επαφή μειονεκτούν στο ότι έχουν περιορισμένη διάρκεια ζωής λόγω φθοράς. Τα κυκλώματα στη κάρτα μπορεί να καταστραφούν από παράγοντες όπως οι ηλεκτροστατικές εκκενώσεις ή η κακή χρήση των καρτών από τους κατόχους.

3.2.2 Contactless Cards

Οι ασύρματες κάρτες χρειάζεται μόνο να βρίσκονται κοντά σε ένα reader και δεν απαιτείται φυσική επαφή. Η κάρτα έχει εσωτερικά ενσωματωμένη κεραία όπως και ο reader και επικοινωνούν μέσω αυτού του ασύρματου συνδέσμου. Οι περισσότερες ασύρματες κάρτες παίρνουν και το ρεύμα για τη λειτουργία του τσιπ τους από το ηλεκτρομαγνητικό σήμα μεταξύ κάρτας και reader.

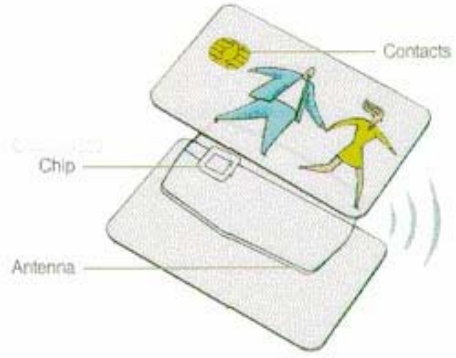


Στη παραπάνω εικόνα φαίνονται τα τρία στρώματα που στοιχειοθετούν μία ασύρματη κάρτα. Το πάνω και το κάτω στρώμα (εξωτερικά στρώματα) κλείνουν εσωτερικά το επίπεδο με την κεραία και το μικροτσίπ. Η κεραία είναι συνήθως 3 - 5 στροφές από πολύ λεπτό σύρμα (ή αγώγιμο μελάνι) που συνδέεται με το μικροτσίπ.
Οι ασύρματες κάρτες χρησιμοποιούνται κυρίως σε εφαρμογές και χώρους όπου οι συναλλαγές πρέπει να γίνονται πολύ γρήγορα, όπως για παράδειγμα στα μέσα συγκοινωνίας και στους σταθμούς διοδίων. Είναι πιο ακριβές από τις κάρτες επαφής αλλά έχουν μεγαλύτερη διάρκεια ζωής και είναι πιο αξιόπιστες.

3.2.3 Combi – Hybrid Cards

Από τις παραπάνω κατηγορίες καρτών που ορίστηκαν ως προς τον τύπο του interface τους, προκύπτουν και δύο ακόμα τύποι, οι Combi και οι Hybrid κάρτες. Οι κάρτες Hybrid, οι οποίες ήδη κυκλοφορούν στην αγορά, έχουν δύο τσιπ, ένα με επαφές και ένα για ασύρματη επικοινωνία. Τα δύο τσιπ δεν επικοινωνούν μεταξύ τους. Υπάρχουν ήδη εφαρμογές στις οποίες αυτός ο τύπος καρτών εξυπηρετεί τους καταναλωτές αλλά και τους παροχείς των καρτών.

Οι κάρτες Combi από την άλλη, ενσωματώνουν και τους δύο τύπους interface σε μία κάρτα με ένα τσιπ. Δηλαδή μπορεί να υπάρχει πρόσβαση στο ίδιο μικροτσίπ και μέσω ηλεκτρικών επαφών στην επιφάνεια της κάρτας και μέσω ασύρματης επικοινωνίας με τη χρήση της κεραίας. Στις κάρτες αυτές το επίπεδο ασφαλείας είναι πολύ υψηλό.

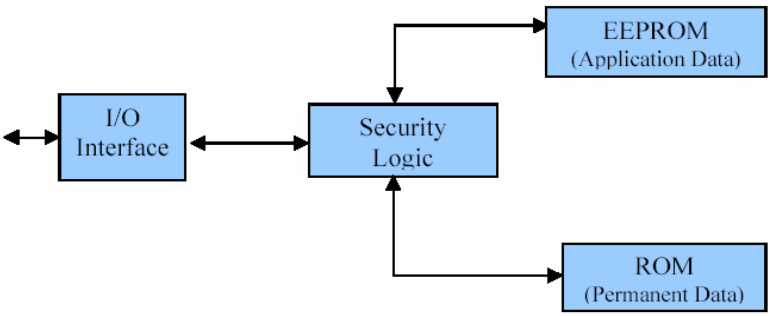


Σχήμα 3.2.3 – Combi Card

Όπως βλέπουμε και στην παραπάνω εικόνα, στην ίδια κάρτα το μικροτσίπ μπορεί να έχει επικοινωνία με τον εξωτερικό κόσμο μέσω των ηλεκτρικών επαφών στην επιφάνεια της κάρτας και μέσω της κεραίας που το περιβάλλει στο εσωτερικό στρώμα της κάρτας. Το ασύρματο τσιπ χρησιμοποιείται για εφαρμογές που χρειάζονται γρήγορες συναλλαγές και το τσιπ με τις ηλεκτρικές επαφές για εφαρμογές που απαιτούν μεγαλύτερη ασφάλεια.
Οι κάρτες αυτές αναμένεται να έχουν μεγάλη απορρόφηση στο χώρο των μέσων μαζικής μεταφοράς και στο τραπεζικό τομέα.

3.2.4 Memory Cards

Εξετάζοντας τις κάρτες ως προς τον τύπο του μικροτσίπ που περιέχουν, προκύπτουν δύο κατηγορίες καρτών: οι κάρτες μνήμης (memory cards) και οι κάρτες με μικροεπεξεργαστή (microprocessor cards).
Οι κάρτες μνήμης δεν έχουν επεξεργαστική δύναμη και δεν μπορούν να χειριστούν αρχεία δυναμικά. Ένα τσιπ μνήμης μπορεί να θεωρηθεί ως μία δισκέτα με διάφορες χωρητικότητες και με προαιρετική ασφάλεια. Οι κάρτες αυτές επικοινωνούν με το reader με σύγχρονα πρωτόκολλα τα οποία θα εξηγήσουμε παρακάτω. Είναι πιο κοινές και φτηνές από τις microprocessor κάρτες αλλά μειονεκτούν στα θέματα προστασίας και διαχείρισης δεδομένων. Μπορούν να αποθηκεύσουν από μερικές εκατοντάδες bit ως συνήθως 16Kbyte πληροφορίας.
Γενικά οι κάρτες μνήμης περιέχουν δύο είδη μνήμης, μνήμη EEPROM και μνήμη ROM. Η μνήμη EEPROM χρησιμοποιείται για την αποθήκευση των δεδομένων της εκάστοτε εφαρμογής και μπορεί να προστατεύεται τμηματικά ή στην ολότητά της από κάποιο κωδικό. Ο κωδικός αυτός μπορεί να παρέχεται από το reader ή από τον κάτοχο της κάρτας κατά τη χρήση της. Η μνήμη ROM χρησιμοποιείται για την αποθήκευση δεδομένων που δεν αλλάζουν στη διάρκεια ζωής της κάρτας, όπως ο αναγνωριστικός αριθμός της κάρτας, τα στοιχεία του κατόχου της κάρτας κ.α.



Σχήμα 3.2.4 – Δομή Κάρτας Μνήμης

Επίσης οι κάρτες μνήμης μπορούν να περιέχουν στη μνήμη τους μία εφαρμογή η οποία δεν εκτελείται από τις ίδιες τις κάρτες που δεν έχουν δύναμη επεξεργασίας, αλλά από τις συσκευές υποδοχής των καρτών με τις οποίες επικοινωνούν.
Αναλύοντας λίγο παραπάνω τη τεχνολογία των καρτών μνήμης, μπορούμε να τις χωρίσουμε σε 3 υποκατηγορίες.

3.2.4.1 Straight Memory Cards

Οι κάρτες αυτές χρησιμεύουν μόνο για την αποθήκευση δεδομένων και δεν μπορούν να προσφέρουν καμία ασφάλεια. Είναι οι πιο φτηνές κάρτες μνήμης. Δεν μπορούν να δηλώσουν τη ταυτότητά τους στο reader, ο οποίος για να έχει επικοινωνία μαζί τους πρέπει εκ των προτέρων να γνωρίζει τι τύπου κάρτες είναι..

3.2.4.2 Protected / Segmented Memory Cards

Αυτές οι κάρτες έχουν ενσωματωμένη λογική για να ελέγχουν την πρόσβαση στη μνήμη. Μπορούν να προγραμματιστούν έτσι ώστε να προστατεύουν κομμάτια ή και ολόκληρη τη μνήμη από διαδικασίες εγγραφής / ανάγνωσης ή και τις δύο. Συνήθως αυτό γίνεται με τη χρήση κάποιου κωδικού ή κλειδιού συστήματος. Οι κάρτες αυτές έχουν τη δυνατότητα να χωριστούν σε λογικές ενότητες με στόχο την χρήση τους σε διαφορετικές ταυτόχρονες εφαρμογές (multi-functionality).

3.2.4.3 Stored Value Memory Cards

Οι κάρτες μνήμης αποθηκευμένης αξίας σχεδιάζονται μόνο για αποθήκευση αξίας ή "αποδείξεων". Μπορούν να είναι και επαναφορτιζόμενες, έχουν δηλαδή τη δυνατότητα με την εξάντληση της αποθηκευμένης αξίας να την ανανεώνουν αποθηκεύοντας καινούρια. Οι περισσότερες από αυτές τις κάρτες συσσωματώνουν στοιχεία ασφαλείας κατά την κατασκευή τους. Οι περιοχές μνήμης σχεδιάζονται και λειτουργούν είτε ως αφαιρέτες είτε ως μετρητές. Ελάχιστη ή καθόλου μνήμη περισσεύει για να χρησιμοποιηθεί σε κάποια άλλη λειτουργία.

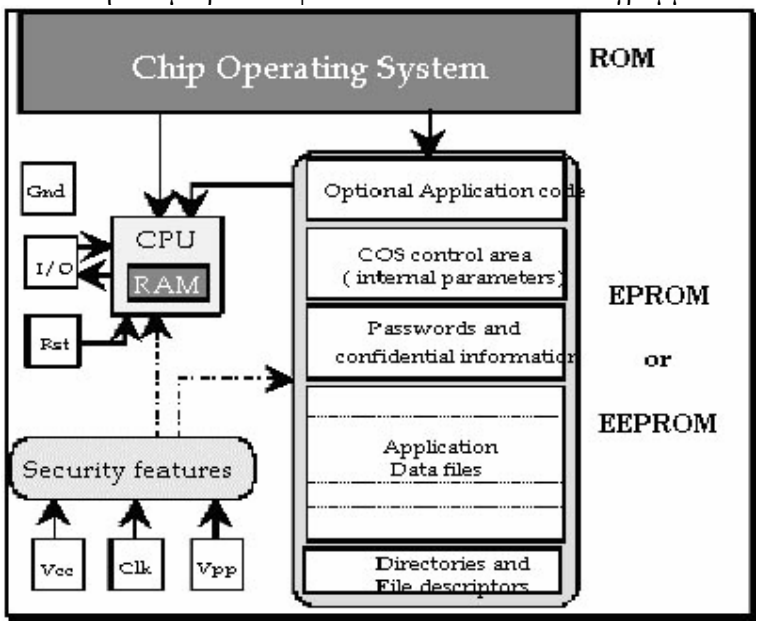
3.2.5 Microprocessor Cards

Σε εφαρμογές που η ασφάλεια παίζει σημαντικό ρόλο χρησιμοποιούνται κάρτες με μικροεπεξεργαστή. Αυτές οι κάρτες είναι οι μόνες που μπορούν να χαρακτηριστούν τεχνικά ως έξυπνες κάρτες.
Οι μικροεπεξεργαστές λειτουργούν όπως ένας υπολογιστής με θύρα εισόδου / εξόδου, λειτουργικό σύστημα και σκληρό δίσκο. Μπορούν να αποθηκεύσουν και να επεξεργαστούν δεδομένα, κυρίως όμως ξεχωρίζουν λόγω της δυνατότητάς τους για δυναμική κρυπτογράφηση και για ενημερώσεις στις λογισμικές εφαρμογές τους. Αυτό σημαίνει ότι μπορεί να προσθέσει ή να αφαιρέσει εφαρμογές ή ακόμα να βελτιώσει μία υπάρχουσα εφαρμογή, γεγονός που κάνει τις microprocessor κάρτες πολύ ευέλικτες.
Ο επεξεργαστής μπορεί να υποστηρίξει διαδικασίες εγγραφής, ανάγνωσης και ενημέρωσης πληροφορίας καθώς και κρυπτογράφησης / αποκρυπτογράφησης δεδομένων αποθηκευμένων στην EEPROM. Χειρίζεται την κατανομή της μνήμης και τη πρόσβαση σε αρχεία και οργανώνει την πληροφορία σε συγκεκριμένες δομές αρχείων μέσω ενός λειτουργικού συστήματος της κάρτας (Card Operating System – COS).
Οι κάρτες με μικροεπεξεργαστή αποτελούνται κυρίως από τα ακόλουθα στοιχεία:

- ROM: η μνήμη ROM περιέχει το λειτουργικό σύστημα της κάρτας και καλείται αλλιώς "μάσκα" (mask) της κάρτας. Οι διάφορες εντολές γράφονται μόνιμα στη μνήμη από τον κατασκευαστή της κάρτας κατά την κατασκευή της. Το μέγεθός της κινείται από μερικά Kbyte μέχρι τα 32Kbyte, ανάλογα με το λειτουργικό σύστημα που χρησιμοποιείται.

- EEPROM: η μνήμη αυτή περιέχει τα προγράμματα εφαρμογών της κάρτας και τα αντίστοιχα δεδομένα των εφαρμογών. Τα περιεχόμενά της δεν είναι μόνιμα, μπορούν δηλαδή να διαγραφούν και να επανεγγραφούν.
- RAM: η μνήμη αυτή χρησιμεύει για την προσωρινή αποθήκευση αποτελεσμάτων από υπολογισμούς ή στοιχείων της επικοινωνίας με τον έξω κόσμο. Τα περιεχόμενά της διαγράφονται όποτε η κάρτα αποσυνδέεται από το ρεύμα.
- CPU: η κεντρική μονάδα επεξεργασίας είναι η καρδιά της microprocessor κάρτας. Η μονάδα επεξεργασίας έχει την ευθύνη της εκτέλεσης διαφόρων εντολών.

Η γενική αρχιτεκτονική του μικροτσίπ φαίνεται στο ακόλουθο διάγραμμα.



Σχήμα 3.2.5 – Αρχιτεκτονική Μικροτσίπ

Όπως αναφέραμε και προηγουμένως, οι κάρτες αυτές έχουν τη δυνατότητα να τρέχουν και να ανανεώνουν τις εφαρμογές τους. Ας αναλύσουμε όμως λίγο τι εννοούμε με τον όρο "εφαρμογή".
Στο χώρο των έξυπνων καρτών ο όρος εφαρμογή χρησιμοποιείται για να περιγράψει το λογισμικό ή το πρόγραμμα που η κάρτα εφαρμόζει. Στην πιο απλή περίπτωση, εφαρμογή μπορεί να είναι ένας διαχειριστής αρχείων για την οργάνωση της αποθήκευσης και της ανάκτησης πληροφοριών. Μία τέτοια εφαρμογή μπορεί να σχεδιασθεί και να πραγματοποιηθεί κατευθείαν στη λογική του μικροτσίπ. Αντίστοιχα το τσιπ πρέπει να έχει τη κατάλληλη λογική σχεδίαση για την επίτευξη επικοινωνίας, μέσω της οποίας θα δέχεται εντολές από το reader καθώς και θα λαμβάνει και θα μεταδίδει τα δεδομένα της εφαρμογής. Οι κάρτες με μικροεπεξεργαστή μπορούν να υποστηρίξουν και πιο προηγμένες εφαρμογές αφού η CPU μπορεί πέρα από την επεξεργασία δεδομένων να πάρει και αποφάσεις πάνω σε θέματα - πράξεις που ανακύπτουν.

3.3 Mask - Λειτουργικό Σύστημα Καρτών (COS)

Η μάσκα των έξυπνων καρτών είναι βασικά μία ακολουθία εντολών οι οποίες έχουν ενσωματωθεί στη μνήμη ROM. Όπως και τα υπολογιστικά προγράμματα Dos και Windows, η μάσκα δεν εξαρτάται από κάποια συγκεκριμένη εφαρμογή αλλά μπορεί συχνά να χρησιμοποιείται από τις περισσότερες εφαρμογές.
Όσο αφορά το COS, υπάρχει το General Purpose COS (Γενικού Σκοπού Λειτουργικό Σύστημα Καρτών) το οποίο είναι ένα σύνολο γενικών εντολών που με διαφορετικές ακολουθίες μπορούν και καλύπτουν τις περισσότερες εφαρμογές.

Από την άλλη υπάρχει το Dedicated COS (Αφιερωμένο Λειτουργικό Σύστημα Καρτών) το οποίο αποτελείται από εντολές σχεδιασμένες για ειδικές εφαρμογές και περιέχει την ίδια την εφαρμογή (όπως στην περίπτωση του e-purse).
Οι βασικές και πιο κοινές λειτουργίες ενός Λειτουργικού Συστήματος Καρτών είναι:

- Διαχείριση των ανταλλαγών μεταξύ του εξωτερικού κόσμου και της κάρτας (στα πλαίσια του πρωτοκόλλου επικοινωνίας)
- Διαχείριση αρχείων και πληροφοριών στη μνήμη
- Διατήρηση αξιοπιστίας, ειδικά στα πλαίσια της συνοχής δεδομένων, των διακοπών μίας ακολουθίας και της ανάκτησης λειτουργίας μετά από λάθος
- Έλεγχος πρόσβασης σε πληροφορίες και λειτουργίες (ανάγνωση, εγγραφή, ενημέρωση δεδομένων και επιλογή αρχείου)
- Διαχείριση διαφόρων φάσεων στη διάρκεια ζωής μίας κάρτας (κατασκευή και προσωποποίηση μίας κάρτας, ενεργός ζωή, τερματισμός ζωής της κάρτας)
- Διαχείριση διαδικασιών ασφαλείας και κρυπτογραφικών αλγορίθμων

Η συνήθης τακτική μέχρι τώρα είναι τα στοιχεία μίας εφαρμογής να ενσωματώνονται στη μάσκα κατά τη διάρκεια της κατασκευής του μικροτσίπ.
Σημαντικό στοιχείο στο τομέα που συζητάμε είναι ότι το COS μίας κάρτας δεν δεσμεύεται από κάποιο συγκεκριμένο πρότυπο, με αποτέλεσμα να υπάρχουν διαφορετικά λειτουργικά συστήματα από κατασκευαστή σε κατασκευαστή ή ακόμα και από τύπο κάρτας σε άλλο τύπο κάρτας του ίδιου κατασκευαστή. Γι' αυτό και είναι απαραίτητη μία βαθμίδα η οποία θα μεσολαβεί για την επικοινωνία μίας εφαρμογής με διαφορετικού τύπου κάρτες ή card readers. Τέτοια βαθμίδα είναι η Διεπαφή Προγραμματισμού της Εφαρμογής (API).

3.4 Application Programming Interface (API)

Το API είναι ένα πρόγραμμα λογισμικού που μεταφράζει τις εντολές και τις λειτουργίες μίας εφαρμογής σε ειδική γλώσσα την οποία αντιλαμβάνεται μία έξυπνη κάρτα ή ένας αναγνώστης καρτών. Το λογισμικό αυτό βοηθάει και την επικοινωνία μεταξύ μίας συσκευής ανάγνωσης - υποδοχής καρτών και διαφόρων τύπων καρτών. Αυτό εξυπηρετεί κυρίως μεγάλες εφαρμογές οι οποίες συνεργάζονται με διαφορετικούς εκδότες - παροχείς έξυπνων καρτών και χρειάζονται μία κοινή βάση επικοινωνίας.
Ένα API δεν μπορεί να σχεδιαστεί για να προβλέψει την επικοινωνία μίας εφαρμογής με όλους τους τύπους καρτών που κυκλοφορούν αλλά έχει τη δυνατότητα να προσφέρει συμβατότητα της εφαρμογής με αρκετά διαφορετικά COS καρτών.

3.5 Card Reader - Terminal

Οι έξυπνες κάρτες χρειάζονται για να λειτουργήσουν συσκευές υποδοχής που ονομάζονται CADs (Card Acceptance Devices) και έχουν δυνατότητες ανάγνωσης / εγγραφής. Οι συσκευές αυτές μπορούν να είναι είτε Readers είτε Terminals. Στο χώρο των έξυπνων καρτών, readers ονομάζουμε αυτές τις συσκευές που λειτουργούν συνδεδεμένες με ένα υπολογιστή για τη πλειοψηφία των επεξεργαστικών τους διεργασιών ενώ τα terminals είναι ανεξάρτητα και "αυτάρκη".
Οι readers διαφοροποιούνται μεταξύ τους σύμφωνα με το τύπο της διεπαφής τους με τον υπολογιστή, με τον οποίο συνδέονται για παράδειγμα μέσω σειριακής θύρας, παράλληλης θύρας, θύρας USB ή θύρας υπερύθρων. Επίσης διαφοροποιούνται από το τύπο καρτών και τα πρωτόκολλα που μπορούν να υποστηρίξουν.
Αντίστοιχα τα terminals διαφοροποιούνται για τους ίδιους λόγους αλλά και από τα αναπτυξιακά εργαλεία που υποστηρίζουν. Πολλά terminal μπορούν να διαβάσουν δεδομένα από μαγνητική ταινία (magstripe image) και να τυπώσουν στοιχεία συναλλαγών.

3.6 Πρότυπα ISO

Τα πρότυπα χρησιμεύουν για να εξασφαλίσουν τη διαλειτουργικότητα των έξυπνων καρτών, που αποτελεί σημαντικότατο παράγοντα για την εξάπλωση της τεχνολογίας αυτής. Ο Παγκόσμιος Οργανισμός Προτύπων (International Standards Organization - ISO) έχει αναπτύξει το πρότυπο ISO 7816 (αποτελείται από τέσσερα μέρη - part 1, 2, 3 and 4) για τις έξυπνες κάρτες, τις οποίες ονομάζει ως κάρτες ολοκληρωμένων κυκλωμάτων (ICC-Integrated Circuit Card ή IC).
Το πρότυπο ISO 7816-1 καθορίζει τα φυσικά χαρακτηριστικά τα οποία πρέπει να έχουν οι έξυπνες κάρτες. Συγκεκριμένα ορίζει χαρακτηριστικά όπως το υλικό και η μέθοδος κατασκευής, οι διαστάσεις των καρτών, η θέση της μαγνητικής ταινίας στη κάρτα (αν υπάρχει) και η μέθοδος χάραξής της, τα όρια έκθεσης σε διάφορα ηλεκτρομαγνητικά φαινόμενα όπως οι ακτίνες X, η ακτινοβολία UV, ηλεκτρομαγνητικά πεδία, στατικά ηλεκτρικά πεδία, θερμοκρασία περιβάλλοντος. Επίσης εξετάζονται οι ιδιότητες των καρτών όταν υπόκεινται σε μηχανική πίεση αφού πρέπει η σύνδεση μεταξύ των ηλεκτρικών επαφών της επιφάνειας και των pin του μικροτσίπ να διατηρούνται όταν για παράδειγμα η κάρτα κάμπτεται. Το κομμάτι αυτό του προτύπου είναι σημαντικό για τους κατασκευαστές καρτών οι οποίοι επιλέγουν τα υλικά και τη διαδικασία της κατασκευής των καρτών.
Το ISO 7816-2 πρότυπο αναφέρεται σε δύο άλλα βασικά στοιχεία, τη θέση των ηλεκτρικών επαφών στην επιφάνεια της κάρτας για τις κάρτες με επαφή και το ελάχιστο μέγεθός τους. Περιέχει πρότυπα για τον αριθμό, τη λειτουργία και τη θέση των ηλεκτρικών επαφών. Οι επαφές του μικροτσίπ είναι 8 (αναφέρονται ως C1-C8) αλλά όπως είδαμε και σε προηγούμενο κεφάλαιο δεν χρησιμοποιούνται όλες.
Ακολούθως, το πρότυπο 7816-3 ορίζει τις ηλεκτρονικές ιδιότητες και τα χαρακτηριστικά του πρωτοκόλλου μετάδοσης των έξυπνων καρτών, στοιχεία που έχουν κύριο ρόλο στη διαλειτουργικότητα των καρτών αυτών. Συγκεκριμένα, αναφέρεται σε ηλεκτρικά χαρακτηριστικά όπως η τάση τροφοδοσίας και το σήμα μηδενισμού (reset signal) και στα στοιχεία της ασύγχρονης ημι-διμερούς μετάδοσης πληροφορίας από τις κάρτες, κάτω από τα πρωτόκολλα T=0 (μετάδοση χαρακτήρων) και T=1 (μετάδοση πακέτων).
Το κομμάτι αυτό είναι σημαντικό για τους κατασκευαστές των αναγνωστών καρτών (card readers) καθώς και για όσους ενδιαφέρονται να έχουν low-level επικοινωνία με τις κάρτες, δηλαδή επικοινωνία σε επίπεδο σημάτων. Περιγράφει αναλυτικά τι περιλαμβάνεται στην ανάπτυξη ενός λογισμικού Εισόδου-Εξόδου (I/O).
Τέλος, το πρότυπο 7816-4 ορίζει:

- Το περιεχόμενο των μηνυμάτων, εντολών και απαντήσεων που μεταδίδονται μεταξύ της συσκευής διεπαφής των καρτών και των ίδιων των καρτών
- Τη δομή και το περιεχόμενο των ιστορικών χαρακτήρων που στέλνονται από τη κάρτα με το ATR (Answer To Reset) σήμα τους
- Τη δομή των αρχείων και των δεδομένων από την οπτική της διεπαφής όταν επεξεργάζεται διαλειτουργικές εντολές προς ανταλλαγή
- Μεθόδους πρόσβασης σε αρχεία και πληροφορίες στη κάρτα
- Μεθόδους ασφαλούς μετάδοσης μηνυμάτων (Secure Messaging)
- Μεθόδους πρόσβασης στους αλγόριθμους που επεξεργάζεται η κάρτα (δεν περιγράφονται οι ίδιοι οι αλγόριθμοι)

Το πρότυπο αυτό δεν καλύπτει την εσωτερική δομή και μορφή του μικροτσίπ της κάρτας. Επιτρέπει την ανάπτυξη περισσοτέρων προτύπων για άλλες διαλειτουργικές εντολές και καινούριους μηχανισμούς ασφαλείας που μπορεί να παρουσιαστούν.

3.7 Πρωτόκολλο Επικοινωνίας T=0

Το πρωτόκολλο αυτό αναφέρεται σε ασύγχρονη ημι-διμερή μετάδοση χαρακτήρων. Η σειριακή επικοινωνία γίνεται με τη χρήση μίας απλής σύνδεσης με το μικροτσίπ της κάρτας και η κατεύθυνση της επικοινωνίας αλλάζει ανάλογα με το αν το ολοκληρωμένο κύκλωμα της κάρτας ή η διεπαφή μεταδίδει δεδομένα. Αυτό χαρακτηρίζεται ως ημι-διμερής μετάδοση ενώ η πλήρης-διμερής μετάδοση χρειάζεται δύο επαφές Εισόδου / Εξόδου (Input / Output connectors) για να μπορεί η μετάδοση να λαμβάνει χώρο και προς τις δύο κατευθύνσεις ταυτόχρονα.
Στην ασύγχρονη μετάδοση, η μετάδοση ενός χαρακτήρα (8 bits) συνοδεύεται πάντα με συνοδευτικά bit τα οποία βοηθούν στην ανίχνευση της μετάδοσης και στον έλεγχο της ορθότητάς της (start bit, parity bit). Η διάρκεια ενός bit ορίζεται ως ένα etu (Elementary Time Unit) και για τις κάρτες με εσωτερικό ρολόι ισούται συνήθως με 1/9600 δευτερόλεπτα. Η γραμμή I/O δειγματοληπτείται με ρυθμό μικρότερο των 0.2etu για να ανιχνευθεί το start bit που δηλώνει την αρχή μετάδοσης πληροφορίας. Η ισοτιμία είναι σωστή (έλεγχος ορθότητας πληροφορίας) όταν ο αριθμός των άσων που μεταδίδονται (το "1" αντιστοιχεί σε ηλεκτρικό σήμα συγκεκριμένης τάσης - high state) είναι άρτιος. Οι συμβάσεις που αφορούν την κωδικοποίηση των επιπέδων τάσης και την αντιστοιχία με άσους και μηδενικά, καθορίζονται από ένα αρχικό χαρακτήρα TS που μεταδίδεται στο ATR που μεταδίδει η κάρτα στην αρχή οποιασδήποτε επικοινωνίας με την διεπαφή.
Η επικοινωνία μεταξύ της κάρτας και της συσκευής διεπαφής γίνεται με την ανταλλαγή εντολών, οι οποίες πάντα εισάγονται από τη μεριά της διεπαφής. Οι εντολές αυτές είτε περιέχουν δεδομένα για το κύκλωμα της κάρτας είτε ζητούν κάποια πληροφορία από το κύκλωμα της κάρτας και η πληροφορία αυτή περιέχεται έπειτα στην απάντηση της κάρτας, με δεδομένη πάντα τη μονή ροή δεδομένων κάποια δεδομένη στιγμή (είτε προς είτε από την κάρτα). Η κατεύθυνση της πληροφορίας εξαρτάται από το τύπο της εντολής και προφανώς πρέπει να δηλώνεται στις δύο πλευρές της επικοινωνίας πριν γίνει η διοχέτευση της πληροφορίας.
Κάθε εντολή περιέχει ένα κομμάτι header, το οποίο αποτελείται από 5 χαρακτήρες (bytes) και στέλνεται από τη συσκευή διεπαφής (συνήθως από το reader) προς το ολοκληρωμένο κύκλωμα της κάρτας. Τα 5 αυτά byte αντιστοιχούν στην τάξη της εντολής CLA (Instruction Class), στον κωδικό της INS (Instruction Code), στις συμπληρωματικές παραμέτρους της εντολής P1 και P2 (δηλώνουν π.χ. μία διεύθυνση στη μνήμη που θα χρησιμοποιηθεί) και στη παράμετρο P3 που δηλώνει τον αριθμό των χαρακτήρων πληροφορίας που θα μεταδοθούν με την εντολή. Αν η εντολή δηλώνει μεταφορά δεδομένων από την κάρτα και η παράμετρος P3 είναι 0, τότε η κάρτα θα μεταδώσει 256 byte πληροφορίας προς τη διεπαφή, ενώ αν η εντολή αναφέρεται σε μετάδοση πληροφορίας προς την κάρτα, τότε όταν η παράμετρος P3 είναι 0 συνεπάγεται μεταφορά μηδενικής πληροφορίας.
Η κάρτα εις απάντηση μίας εντολής από τη συσκευή διεπαφής, στέλνει ένα procedure byte (διαδικαστικός χαρακτήρας), μετά το οποίο δεδομένα στέλνονται προς την κάρτα ή από αυτήν, ανάλογα με την εντολή που προηγήθηκε. Το procedure byte ορίζει στην συσκευή διεπαφής τη δράση που θα ακολουθήσει. Μπορεί να είναι είτε ACK (Acknowledgement byte), είτε NULL, είτε τέλος SW (Status byte). Στην πρώτη περίπτωση η συσκευή διεπαφής ορίζει την κατάσταση της τάσης προγραμματισμού Vpp και ανταλλάσσει δεδομένα με την κάρτα ανάλογα με τη τιμή του ACK byte. Στη δεύτερη περίπτωση η κάρτα ζητάει απλά περισσότερο χρόνο για την επεξεργασία της εντολής και δηλώνει στη συσκευή διεπαφής να περιμένει το επόμενο procedure byte. Τέλος όταν το procedure byte είναι SW δηλώνει στη συσκευή την κατάσταση της κάρτας μετά την εκτέλεση μίας εντολής.
Αναλυτική παρουσίαση διαφόρων εντολών / απαντήσεων και των status bytes θα ακολουθήσει στο επόμενο κεφάλαιο, όπου αναφερόμαστε στο τύπο καρτών και reader που χρησιμοποιήσαμε για την υλοποίηση των εφαρμογών της διπλωματικής αυτής.

4

Εφαρμογή με GemClub

Για μία πρώτη εξοικείωση με τη τεχνολογία των έξυπνων καρτών και την υλοποίηση των τριών εφαρμογών της διπλωματικής, χρησιμοποιήσαμε τα προϊόντα της GEMPLUS, μίας από τις μεγαλύτερες εταιρίες στο χώρο των έξυπνων καρτών η οποία παρέχει από συμβουλευτικές υπηρεσίες και σχεδιασμό εφαρμογών, στην παροχή λογισμικού και εξοπλισμού και την πλήρη υλοποίηση των εφαρμογών. Η GEMPLUS κατέχει τα 5 τελευταία χρόνια την πρώτη θέση στις παραγγελίες καρτών και διατηρεί σχεδόν το 32% της αγοράς των έξυπνων καρτών.
Τα προϊόντα που χρησιμοποιήσαμε είναι ο reader GemPC410 και οι έξυπνες κάρτες GemClub Micro 1K.

4.1 GemPC410 Reader

Ο μικρός αυτός αναγνώστης καρτών είναι σχεδιασμένος για να συνδέεται με περιβάλλον υπολογιστή και αποτελεί ένα από το πιο σύγχρονα περιφερειακά για ένα υπολογιστή. Είναι απλός στη χρήση και την εγκατάσταση και δεν απαιτεί από το χρήστη ιδιαίτερη τεχνογνωσία.
Ο reader διαχειρίζεται την διεπαφή με τις έξυπνες κάρτες ενώ ο υπολογιστής υποστηρίζει και διαχειρίζεται τις αντίστοιχες εφαρμογές. Είναι συμβατός με όλους τους βασικούς υπολογιστές και τα κύρια λειτουργικά συστήματα και παίρνει ρεύμα από τη θύρα του πληκτρολογίου, παρακάμπτοντας έτσι περιορισμούς που προκύπτουν από άλλες επιλογές τροφοδοσίας.

Σχήμα 4.1 – GemPC410 reader

Ο GemPC410 βασίζεται στη τεχνολογία GemCore της Gemplus, το οποίο σημαίνει ότι μπορεί να χειριστεί όλους τους τύπους καρτών που συμμορφώνονται με το πρότυπο ISO 7816-1/2/3/4 χωρίς προβλήματα συμβατότητας. Λειτουργεί σε όλα τα κύρια περιβάλλοντα (DOS, Windows 3.x / 95 / 98 / Me / 2000 / XP, OS2 κ.ά.) και με όλους τους τύπους καρτών. Επίσης μπορεί να χρησιμοποιηθεί σε καινούριες υπηρεσίες έξυπνων καρτών μόλις αυτές παρουσιαστούν.
Περιέχει ένα LED, μία σχισμή – υποδοχή για τις έξυπνες κάρτες και ένα καλώδιο για σύνδεση σε σειριακή θύρα.
Το LED αναβοσβήνει όσο ο reader είναι συνδεδεμένος στη τροφοδοσία και δεν έχει εισαχθεί κάρτα, ενώ μένει αναμμένο όσο υπάρχει κάρτα στην κατάλληλη σχισμή.
Τα κύρια χαρακτηριστικά του GCR410 αναλύονται στον πίνακα που ακολουθεί:

ΣΤΟΙΧΕΙΟ	ΠΕΡΙΓΡΑΦΗ
Διεπαφή με έξυπνες κάρτες	• Διαβάζει από και γράφει σε όλες τις κάρτες μνήμης και μικροεπεξεργαστή που είναι συμβατές με το πρότυπο ISO 7816-1/2/3/4 • Υποστηρίζει κάρτες των 3V και των 5V • Έχει εγγύηση για 100.000 κύκλους εισαγωγής καρτών
Επικοινωνία	• Προγραμματίζεται στα 9,600 ή στα 115,200 baud με την κάρτα • Είναι μέχρι και 38.400 με το PC
Κατανάλωση Ρεύματος	Κατά μέσο όρο στα 20 mA σε συνθήκες λειτουργίας
Modes Διεπαφής	• Σειριακή επικοινωνία με το PC με σύνδεση RS232 • TLP224 και GBP (Gemplus Block Protocol)
Τροφοδοσία	5V το μέγιστο

Στην ουσία ένας reader μπορεί να καλείται "αναγνώστης" καρτών αλλά χρησιμοποιείται και για λειτουργίες εγγραφής και για λειτουργίες ανάγνωσης με έξυπνες κάρτες.
Ακολούθως θα εξηγήσουμε τη βασική δομή ενός GemCore-based reader όπως ο GemPC410 και θα εξηγήσουμε τις διαδικασίες επικοινωνίας μεταξύ Host, reader και καρτών.
Ένας GemCore-based reader αποτελείται από ένα προγραμματισμένο ελεγκτή και μέχρι 9 Gemplus IC100 τσιπ διεπαφής έξυπνων καρτών, έχει σχεδιασθεί για να απλοποιήσει την ενσωμάτωση διεπαφών με έξυπνες κάρτες σε ηλεκτρονικές συσκευές και διαχειρίζεται την επικοινωνία με έξυπνες κάρτες που είναι συμβατές με το πρότυπο 7816-1/2/3/4.
Το λογισμικό μέσα στο reader είναι συμβατό με το λειτουργικό σύστημα των reader της Gemplus (OROS) και εισάγει πρωτόκολλα επικοινωνίας για το host σύστημα (TLP / GBP protocol) καθώς και για σύγχρονες ή ασύγχρονες κάρτες.
Όλες οι μεταδόσεις με τον reader ελέγχονται από 3 στρώματα πρωτοκόλλου:

- Στρώμα Εντολών: χειρίζεται και μεταφράζει τις εντολές του reader. Κάθε εντολή αποτελείται από ένα κωδικό, δεδομένα και παραμέτρους.
- Στρώμα Μεταφοράς: το στρώμα αυτό χειρίζεται τη διευθυνσοδότηση μηνυμάτων, καθορίζει το τύπο της μετάδοσης και επικυρώνει κάθε μετάδοση. Το στρώμα αυτό μπορεί να χρησιμοποιήσει ένα από τα δύο πρωτόκολλα TLP224 ή GBP (Gemplus Block Protocol).
- Φυσικό Στρώμα: χειρίζεται την ίδια την μετάδοση πληροφορίας.

Ο host επικοινωνεί με τον reader μέσω της σειριακής θύρας με ένα από τα δύο πρωτόκολλα και αυτός με τη σειρά του επικοινωνεί με την κάρτα με μία σειρά ηλεκτρικών σημάτων.

4.1.1 GemCore – Based Reader Commands

Όπως είπαμε προηγουμένως, μία εντολή προς το reader αποτελείται από 3 τμήματα και έχει την ακόλουθη μορφή:

| CommCode | Parameters | Data |

όπου CommCode είναι ο κωδικός της εντολής, Parameters οι παράμετροι που στέλνονται με την εντολή και Data τα δεδομένα που συνοδεύουν την εντολή, όταν χρειάζεται.
Ο reader απαντάει σε κάθε εντολή που λαμβάνει με μία απάντηση της μορφής:

| S | Data |

όπου S είναι το αναγνωριστικό του κωδικού κατάστασης και Data τα δεδομένα που επιστρέφονται μαζί με τον κωδικό κατάστασης, αν χρειάζεται. Όταν S=0 σημαίνει ότι η εντολή εκτελέστηκε σωστά από τον reader.
Οι κυριότερες εντολές που υποστηρίζονται από ένα GemCore - Based reader είναι οι εξής:

Εντολές διαμόρφωσης του reader:

- Configure SIO line: θέτει την ισοτιμία, το ρυθμό μετάδοσης και τον αριθμό των bit ανά χαρακτήρα για τη γραμμή I/O.
- Set mode: επιτρέπει την απενεργοποίηση της συμβατότητας με τις εντολές του OROS και ορίζει το mode λειτουργίας του reader (TLP ή κανονικό).
- Set delay: καθυστερεί τις απαντήσεις από τον reader (για αργούς υπολογιστές)
- Read firmware version: διαβάζει την έκδοση του λογισμικού του reader
- Restart: μηδενίζει το λειτουργικό σύστημα του reader και έτσι επαναφέρει όλες τις παραμέτρους στις προκαθορισμένες αρχικές τιμές τους.

Εντολές διεπαφής με τις κάρτες:

- Power down: αφαιρεί τη τροφοδοσία της κάρτας.
- Power up: τροφοδοτεί και μηδενίζει την κάρτα.
- ISO output: στέλνει ISO OUT εντολές, δηλαδή εντολές που ζητούν και λαμβάνουν δεδομένα από την κάρτα.
- ISO input: στέλνει ISO IN εντολές, δηλαδή εντολές που στέλνουν δεδομένα στην κάρτα.
- Exchange APDU: στέλνει μία Μονάδα Πρωτοκόλλου Δεδομένων Εφαρμογής (Application Data Protocol Unit) στη κάρτα και λαμβάνει μία APDU απάντηση (χρησιμοποιείται μόνο για κάρτες που λειτουργούν με το πρωτόκολλο T=1).
- Define main card type: δηλώνει το τύπο της κάρτας που χρησιμοποιείται.
- Card status: παίρνει πληροφορίες για την κατάσταση της κάρτας, όπως ο τύπος που χρησιμοποιείται εκείνη τη στιγμή, η παρουσία της κάρτας, η τιμή της τάσης τροφοδοσίας, το πρωτόκολλο επικοινωνίας που χρησιμοποιείται (T=0, T=1), τις παραμέτρους ταχύτητας μετάδοσης μεταξύ της κάρτας και του reader.
- Directory: παρέχει πληροφορίες για τους τύπους καρτών που υποστηρίζει ο reader και για τα χαρακτηριστικά του προγράμματος οδήγησης κάθε κάρτας.

Για παράδειγμα, η εντολή set delay προς το reader είναι η εξής: 23h 01h 00h 67h 01h *Delay* όπου *Delay* είναι η καθυστέρηση της απάντησης από τον reader σε ms (0-255).
Υπάρχει επίσης εντολή διαχείρισης του LED του reader, όπως και εντολές διαχείρισης της LCD οθόνης, του buzzer, και του keypad, για εκείνους του GemCore-Based readers που έχουν αυτά τα στοιχεία.

4.1.2 GemCore – Based Reader Interface Library

Η βιβλιοθήκη της διεπαφής των reader που βασίζονται στη τεχνολογία GemCore περιέχει ένα σύνολο εντολών που επικοινωνούν με έξυπνες κάρτες μέσω των GemCore-Based reader. Μπορούν να κλιθούν από εφαρμογές γραμμένες σε C ή Visual Basic. Η βιβλιοθήκη αυτή δίνει την δυνατότητα στο προγραμματιστή να σχεδιάσει εφαρμογές για έξυπνες κάρτες οι οποίες θα χρησιμοποιούν ως μέσο επικοινωνίας με τις κάρτες τους συγκεκριμένους αυτούς reader (GemCore-Based readers).
Η όλη σχεδίαση και υλοποίηση της βιβλιοθήκης έχει βασιστεί στην προσπάθεια να είναι η βιβλιοθήκη ανεξάρτητη από το λειτουργικό σύστημα της κάρτας (COS). Με τη χρήση των εντολών της, μπορεί να επιτευχθεί επικοινωνία με κάρτες με μικροεπεξεργαστές καθώς και με κάθε τύπο σύγχρονων και συνήθων καρτών.
Είναι σημαντικό σε αυτό το σημείο να γίνει κατανοητή η διαφορά μεταξύ των εντολών reader (εντολές που αναφέρονται στους reader μόνο) και των εντολών έξυπνων καρτών.
Οι εντολές reader (του λειτουργικού συστήματος του reader) αφορούν κυρίως το reader και στέλνονται από το host σύστημα άμεσα και κατευθείαν στο reader.
Οι εντολές έξυπνων καρτών σχετίζονται με λειτουργίες των καρτών και δεν μπορούν να σταλθούν κατευθείαν από το host σύστημα και την εφαρμογή στη κάρτα. Αντίθετα πρέπει να μεσολαβήσει ο reader για να "μεταφράσει" και να μεταβιβάσει την εντολή στη κάρτα. Το ίδιο συμβαίνει και με τις απαντήσεις της κάρτας προς την εφαρμογή και το host σύστημα που πρέπει να περάσουν πρώτα "μέσα από" το reader. Έτσι για να στείλει μια εφαρμογή μία εντολή στην κάρτα πρέπει να χρησιμοποιήσει τις ISO IN και ISO OUT εντολές του reader.

4.1.2.1 Ανάπτυξη Εφαρμογής

Μία εφαρμογή που έχει αναπτυχθεί με βάση τη βιβλιοθήκη αυτή πρέπει να ακολουθεί για τη σωστή υλοποίησή της την ακόλουθη διαδικασία.

- Να διαμορφώσει την θύρα στην οποία είναι συνδεδεμένος ο reader και να καθορίσει τον τύπο του reader. Για αυτό το σκοπό χρησιμοποιεί την εντολή G4_OpenChannel.
- Να καθορίσει τον τύπο των καρτών για τις οποίες αναπτύχθηκε η εφαρμογή και να μηδενίσει την κάρτα (card reset) με την εντολή G4_OpenSession.
- Να στείλει και να λάβει δεδομένα από την κάρτα χρησιμοποιώντας την εντολή G4_ExchangeApdu.
- Να κλείσει στο τέλος της εφαρμογής τις συνόδους επικοινωνίας με την κάρτα χρησιμοποιώντας την εντολή G4_CloseSession .
- Να τερματίσει χρησιμοποιώντας την εντολή G4_CloseChannel, κλείνοντας έτσι το ανοιχτό κανάλι επικοινωνίας εφαρμογής - reader.

4.1.2.2 Κωδικοί Κατάστασης (Status Codes)

Όταν μία εντολή εκτελείται, η βιβλιοθήκη επιστρέφει ένα κωδικό κατάστασης ο οποίος επισημαίνει την έκβαση της εντολής. Κωδικοί λαθών στέλνονται όταν:

- Υπάρχει πρόβλημα με τον Host υπολογιστή. Για παράδειγμα, αν η ορισμένη θύρα επικοινωνίας είναι ανοιχτή ή ένα λάθος συστήματος έχει λάβει θέση. Σε αυτή την περίπτωση επιστρέφεται ένα host status code.
- Υπάρχει πρόβλημα μεταξύ του Host υπολογιστή και του GemCore reader. Για παράδειγμα, αν υπάρχει λάθος επικοινωνίας μεταξύ των δύο μελών ή αν κάποιο μήνυμα έχει πολύ μεγάλο μήκος. Στη συγκεκριμένη περίπτωση επιστρέφεται ένα host/reader status code.

- Υπάρχει πρόβλημα με τον reader. Για παράδειγμα, αν ο reader δεν είναι σωστά συνδεδεμένος ή αν μία εντολή του δεν μπορεί να πραγματοποιηθεί. Τότε επιστρέφεται ένα reader status code.
- Υπάρχει πρόβλημα μεταξύ του reader και της κάρτας. Για παράδειγμα, αν το ATR (Answer To Reset) που επιστρέφεται από την κάρτα δεν είναι σωστό ή αν μία λάθος εντολή APDU έχει σταλεί στη κάρτα. Τότε επιστρέφεται ένα reader/card status code.
- Υπάρχει πρόβλημα με την ίδια την κάρτα. Για παράδειγμα, αν δεν υπάρχει κάρτα μέσα στο reader ή αν έχει αφαιρεθεί χωρίς τη σωστή διαδικασία. Στην περίπτωση αυτή επιστρέφεται ένα card status code.

Αν μία εντολή εκτελεστεί σωστά, τότε η βιβλιοθήκη επιστρέφει στην εφαρμογή ένα θετικό ή μηδενικό status code. Όλες τα status codes που μπορεί να επιστραφούν, υπάρχουν στο Reference Manual της βιβλιοθήκης.

4.1.2.3 Εντολές βιβλιοθήκης

Οι ακόλουθες εντολές διαλειτουργούν με έξυπνες κάρτες μέσω όλων των τύπων reader της Gemplus. Υπάρχουν και κάποιες ακόμα για τη διαχείριση LCD οθόνης, buzzer, και του keypad για τους reader που έχουν αυτά τα στοιχεία, οι οποίες δεν αναφέρονται εδώ.

- G4_OpenChannel: ανοίγει ένα λογικό κανάλι επικοινωνίας με τη συσκευή και αντιστοιχεί ένα λογικό νούμερο σε αυτό. Μπορούν να ανοιχτούν πολλά κανάλια για την ίδια εφαρμογή, αλλά πρέπει όλα να κλειστούν πριν τερματίσει η εφαρμογή.
- G4_LockChannel: κλειδώνει την πρόσβαση στη θύρα που σχετίζεται με το λογικό νούμερο του καναλιού επικοινωνίας. Όλες οι άλλες εφαρμογές θα εμποδιστούν να επικοινωνήσουν με τον reader αυτό.
- G4_UnlockChannel: ξεκλειδώνει την πρόσβαση σε μία θύρα που σχετίζεται με ένα λογικό νούμερο καναλιού επικοινωνίας.
- G4_CloseChannel: κλείνει ένα κανάλι που είχε ανοίξει νωρίτερα. Αυτή η εντολή πρέπει να εκτελείται για να απελευθερώνει την θύρα για κάποια άλλη εφαρμογή.
- G4_OpenSession: ανοίγει μία σύνοδο με ένα συγκεκριμένο τύπο καρτών και ενημερώνει μία δομή συνόδου (μία δομή παραμέτρων που χαρακτηρίζουν την εκάστοτε σύνοδο, κρατώντας στοιχεία για το τύπο της κάρτας, το μήκος της APDU που υποστηρίζει η κάρτα, το μήκος του ATR της κάρτας και άλλα) με τα στοιχεία του ATR της κάρτας. Πριν το άνοιγμα της συνόδου, τερματίζει και μετά επαναφέρει τη τροφοδοσία της κάρτας.
- G4_CloseSession: κλείνει μία υπάρχουσα σύνοδο και τερματίζει τη τροφοδοσία της κάρτας.
- G4_SwitchSession: ανοίγει μία "warm" όπως καλείται σύνοδο όπως και πριν χωρίς όμως να τερματίζει τη τροφοδοσία της κάρτας. Αυτό γίνεται μόνο όταν υπάρχει ήδη ανοιχτή κάποια άλλη σύνοδος.
- G4_ICCSetVoltage: αλλάζει τη τάση τροφοδοσίας του κυκλώματος της κάρτας.
- G4_ICCGetVoltage: διαβάζει τη τάση τροφοδοσίας του κυκλώματος της κάρτας.
- G4_ICCSetPTS: αλλάζει τις παραμέτρους της διαχείρισης επιλογής τύπου πρωτοκόλλου (PTS) που αναφέρεται στα πρωτόκολλα T=0 και T=1.
- G4_ICCGetPTS: διαβάζει τις παραμέτρους της διαχείρισης επιλογής τύπου πρωτοκόλλου (PTS) που αναφέρεται στα πρωτόκολλα T=0 και T=1.
- G4_ExchangeApdu: στέλνει μία APDU εντολή στη κάρτα μέσω ενός ανοιχτού καναλιού και επιστρέφει την απάντηση της κάρτας.
- G4_CmdGetTimeout: διαβάζει τη τιμή του timeout που χρησιμοποιείται. Timeout είναι ο χρόνος που χρειάζεται ο reader για να αντιδράσει σε μία εντολή και εξαρτάται από το τύπο της κάρτας που χρησιμοποιείται.
- G4_CmdSetTimeout: ορίζει τη τιμή του timeout για τις επόμενες εντολές.

- G4_ICCStatus: διαβάζει το πρωτόκολλο που χρησιμοποιείται μεταξύ reader και κάρτας (T=0 ή T=1).
- G4_IFDStatus: επιστρέφει πληροφορίες για τον reader όπως το πρωτόκολλο επικοινωνίας του με το host σύστημα και τις παραμέτρους επικοινωνίας που χρησιμοποιούνται για την εντολή G4_OpenChannel.
- G4_IFDExchange: επιτρέπει ανταλλαγές με τον reader. Με αυτή την εντολή μπορούν να σταλθούν οι εντολές reader που αναφέραμε στη προηγούμενη ενότητα, όπως η εντολή για τη διαχείριση του LED.
- G4_IFDGetPowerTimeout: διαβάζει τη τιμή του power timeout για τον επιλεγμένο reader. Το power timeout είναι ο χρόνος που χρειάζεται ο reader για να τερματίσει τη τροφοδοσία της κάρτας κατά τη διαδικασία ανοίγματος μίας συνόδου.
- G4_IFDSetPowerTimeout: αλλάζει τη τιμή του power timeout για τον reader.

4.1.2.4 Παραδείγματα χρήσης εντολών βιβλιοθήκης

Για να δούμε ένα παράδειγμα, η εντολή G4_ICCGetVoltage έχει στη γλώσσα C τη δήλωση

```
INT16 G_DECL G4_ICCGetVoltage (const WORD16 ChannelNb,
                               WORD16 G_FAR *Voltage
                              )
```

όπου ChannelNb είναι το λογικό νούμερο που έχει αντιστοιχηθεί στο επιλεγμένο κανάλι όταν αυτό άνοιξε και Voltage είναι η τιμή της τάσης τροφοδοσίας που διαβάζεται.

Επίσης, για να ανοίξει μία εφαρμογή κανάλι επικοινωνίας με τον reader, μπορεί να χρησιμοποιήσει τον κώδικα:

```
G4_CHANNEL_PARAM Channel; /* δομή με πληροφορίες για το κανάλι */
WORD16 ChanNb;            /* παράμετρος που παίρνει το λογικό  */
                          /* νούμερο του καναλιού που ανοίγει  */


Channel.IFDMode=G_SERIAL; /* τύπος σύνδεσης reader=σειριακός   */
Channel.IFDType=9;        /* τύπος reader=9=GCR410             */
Channel.Comm.Serial.Port=G_COM1; /* θύρα σύνδεσης reader=COM1  */
Channel.Comm.Serial.ITNumber=255; /* τιμή διακοπής για τη θύρα */
Channel.Comm.Serial.BaudRate=9600; /* ρυθμός baud για τη θύρα  */
Channel.IFDBaudRate=9600;         /* ρυθμός baud για το reader */
ChanNb=G4_OpenChannel(&Channel); /*ανοίγει κανάλι με παραμέτρους*/
                                 /*αυτές που ορίστηκαν παραπάνω  */
                                 /*και το λογικό νούμερο που θα  */
                                 /*του αποδοθεί αποθηκεύεται στη */
                                 /*παράμετρο ChanNb              */


if (ChanNb < 0) /* αν επιστραφεί λάθος, τυπώνει τον κωδικό του  */
{ printf("G4_OpenChannel Error %d!\n", ChanNb);
  return(ChanNb); }
else return(ChanNb);
```

Όπως ειπώθηκε και ανωτέρω, για να επικοινωνήσει το host σύστημα με την κάρτα και να της στείλει εντολές, πρέπει να το πραγματοποιήσει μέσω του reader, χρησιμοποιώντας τις ISO in και ISO out εντολές του reader.
Οι εντολές αυτές έχουν τη μορφή | CLA | INS | P1 | P2 | P3 | που αναφέραμε στην ενότητα 3.7 (Πρωτόκολλο Επικοινωνίας T=0) και με διάφορους συνδυασμούς των παραμέτρων CLA, INS, P1, P2 και P3 περιέχουν με τη σειρά τους εντολές συγκεκριμένες για τις κάρτες που μπορεί να είναι διαφορετικές για διάφορους τύπους καρτών.
Με τη χρήση της βιβλιοθήκης GemCore–Based Reader Interface Library οι δύο αυτές εντολές συνενώνονται στην εντολή G4_ExchangeApdu, όπου η δομή G4_APDU_COMM που χρησιμοποιείται ως είσοδος στη συνάρτηση, περιέχει τις παραμέτρους CLA, INS, P1, P2 και P3 εκτός άλλων. Προφανώς με την εντολή αυτή μία εφαρμογή μπορεί να μιλήσει μέσω του GemCore-Based reader σε όλες τις κάρτες που είναι συμβατές με το πρότυπο ISO 7816.
Οι συγκεκριμένες εντολές για τον GemClub τύπο κάρτας περιέχονται σε μία άλλη βιβλιοθήκη την οποία θα εξετάσουμε σε επόμενο κεφάλαιο και απλουστεύουν τη διαδικασία επικοινωνίας της εφαρμογής με την κάρτα.

4.2 GemClub Cards

Η GemClub Micro είναι μία εύκολη στη χρήση κάρτα με μικροεπεξεργαστή που έχει σχεδιασθεί για ανεπτυγμένες εφαρμογές προγραμμάτων εμπιστοσύνης αλλά μπορεί ταυτόχρονα να χρησιμοποιηθεί σε πολλούς διαφορετικούς τύπους εφαρμογών όπως ιδιωτικό ηλεκτρονικό πορτοφόλι, κάρτες αποθηκευμένης αξίας, έλεγχος ταυτότητας πελάτη, μέτρηση μεγεθών (π.χ. μέτρηση κατανάλωσης ρεύματος) κ.ά. Η GemClub Micro 1K περιέχει 1Kilobyte (1000 byte) μνήμης EEPROM.

Σχήμα 4.2 – GemClub card

Οι GemClub Micro κάρτες στέλνουν και λαμβάνουν δεδομένα σύμφωνα με το πρωτόκολλο T=0 σε συμφωνία με το πρότυπο 7816-3 (ενότητα 3.6). Προσφέρουν υψηλής ταχύτητας επικοινωνία στα 9,600 ή στα 115,200 baud.
Οι κάρτες αυτές έχουν flat-file δομή, το οποίο σημαίνει ότι όλα τα αρχεία που χρειάζονται για την εφαρμογή του προγράμματος εμπιστοσύνης βρίσκονται στο ίδιο επίπεδο στη μνήμη.
Τα αρχεία χωρίζονται στους εξής τύπους:

- Αρχείο Συστήματος (System File)
- Αρχεία Εμπιστοσύνης (Loyalty Files)
- Αρχεία Εγγραφής (Record Files)
- Αρχεία Ασφαλείας (Security Files)

Το λειτουργικό σύστημα των καρτών GemClub περιλαμβάνει εντολές διαχείρισης (Administrative commands) και εντολές εφαρμογής (Application commands) που θα εξετάσουμε αργότερα. Επίσης υποστηρίζει διαχείριση πρόσβασης σε δεδομένα με συνθήκες πρόσβασης για την προστασία αρχείων και στοιχείων πληροφορίας καθώς επίσης και προστασία στη χρήση κάθε εντολής. Γενικά, τα αρχεία της κάρτας προστατεύονται με μυστικούς κωδικούς και 3DES κλειδιά.
Τέλος, οι κάρτες GemClub–EMV είναι συμβατές με το πρότυπο EMV (της Europay, MasterCard, και Visa για τις τραπεζικές συναλλαγές) και μπορούν να γίνουν δεκτές σε EMV τερματικά.

4.2.1 Ηλεκτρικά χαρακτηριστικά

Τα DC χαρακτηριστικά της κάρτας στα 5V φαίνονται στον ακόλουθο πίνακα:

Παράμετρος	Περιγραφή	Μονάδα	Ελάχιστο	Μέγιστο
Vcc	Supply voltage	V	4.5	5.5
ICC	Supply current consumption	mA		15
ICC sleep	Supply current consumption (idle)	μA		200
Frequency	Clock frequency	MHz	1	5
T^o	Chip operating temperature	^oC	-20	55
I/O VIH	Input high-level voltage	V	0.7*Vcc	Vcc
I/O VIL	Input low-level voltage	V	0.0	0.2*Vcc
I/O IIL	*where* 0<Vil<0.2*Vcc	mA		1
I/O IIH	*where* 0.7*Vil<Vih<Vcc	μA		500
I/O VOH	Output high-level voltage (where IOH=-20μA)	V	3.8	Vcc
I/O VOL	Output low-level voltage (where IOL=1.6mA)	V	0.0	0.4
CLOCK VIH	Input high-level voltage	V	0.7*Vcc	Vcc
CLOCK VIL	Input low-level voltage	V	0.0	0.2*Vcc
CLOCK IIH/IIL	*where* 0<Vil<0.2*Vcc 0.7*Vil<Vih<Vcc	μA	-20	20
RESET VIH	Input high-level voltage	V	0.7*Vcc	Vcc
RESET VIL	Input low-level voltage	V	0.0	0.2*Vcc
RESET IIH/IIL	*where* 0<Vil<0.2*Vcc 0.7*Vil<Vih<Vcc	μA	-20	20

Η κάρτα GemClub περιλαμβάνει και ένα mode αναμονής όπου μειώνεται η κατανάλωση ρεύματος όταν η κάρτα είναι ανενεργή. Η κάρτα μπαίνει αυτόματα σε αυτή την κατάσταση μετά την εκτέλεση κάθε εντολής.

4.2.2 Φυσικά χαρακτηριστικά

Οι διαστάσεις της κάρτας είναι:

- Μήκος = 85,6 mm
- Πλάτος = 53,97 mm
- Πάχος = 0,78 mm.

Το εύρος της θερμοκρασίας λειτουργίας του μικροτσίπ είναι από τους –20°C ως τους 85°C. Ο πίνακας που ακολουθεί δίνει τα χαρακτηριστικά αξιοπιστίας της GemClub κάρτας.

	Τεστ	Πρότυπο Αναφοράς	Μεθοδολογία	Κριτήρια Επιτυχίας Τεστ
1	Card dimensions	ISO 7810	85.47<length<85.72mm 53.92<width<54.03mm 0.76<thickness<0.84mm	Nominal functionality
2	Contact location	ISO 7816-2		Nominal functionality
3	Level difference between contacts and card	ISO 7816-1	-100 μm; +100 μm	Nominal functionality
4	Dynamic bending torsional stress	ISO 10373	1 ISO cycle = 500 bends (width) 500 bends (length) 500 torsions	Nominal functionality Visual
5	Salt atmosphere	CEI68211	48 h, 35°C, 45% RH, 5% NaCl	Nominal functionality
6	Chip Assembly Humidity	CECC 90 000	168 h at 85°C / 85% RH	Nominal functionality
7	Vibration	ISO 10373	3 axes, 1hr each direction. 10g acceleration, 10Hz - 500Hz	Nominal functionality
8	Reader insertion for contacts		10 000 insertions	Nominal functionality
9	Card stability with temperature		48 h at 70°C (dry °T) (depending on the card body)	Nominal functionality Dimensions
10	Cold storage temperature		24 h at -25°C	Nominal functionality Dimensions
11	Card stability with humidity (and temperature)	Based on 7810 / 8-1-5	168 h at 50°C / 93% RH	Nominal functionality Dimensions
12	Data retention	Semi-conductor standard	10 years / 70°C	Nominal functionality
13	ESD protection	MIL STD - 883 Method 3015-6	Class A: 4Kv	Nominal functionality
14	EEPROM		100.000 write / erase cycles	Nominal functionality

4.2.3 Δομή πληροφοριών

Όπως ειπώθηκε οι κάρτες GemClub υποστηρίζουν τους εξής τύπους αρχείων:

- Αρχείο Συστήματος (System File): χρησιμεύει για τη διαχείριση της συνολικής ασφάλειας της κάρτας. Υπάρχει μόνο ένα τέτοιο αρχείο ανά κάρτα.
- Αρχεία Εμπιστοσύνης (Loyalty Files): χρησιμοποιούνται για την αποθήκευση πόντων, προφίλ πελατών και άλλων πληροφοριών σε αρχεία Μετρητών (Counter files) και κανόνων σε αρχεία Κανόνων (Rule Files).
- Αρχεία Εγγραφής (Record Files): χρησιμεύουν για την αποθήκευση δεδομένων. Τα αρχεία αυτά συμμορφώνονται με τα πρότυπα ISO/IEC 7816-4 για τη διαχείριση εγγραφών και ονομάζονται σε αντιστοιχία με τα πρότυπα αυτά.
- Αρχεία Ασφαλείας (Security Files): χρησιμοποιούνται για την αποθήκευση μυστικών κωδικών (Secret Codes) και μυστικών κλειδιών (Secret Keys) για τις ανάγκες ασφαλείας των καρτών.

Για να επιλεγεί ένα αρχείο πρέπει να χρησιμοποιηθούν οι ακόλουθες μέθοδοι επιλογής:

- Τύπος αρχείου: ταυτοποιεί το σκοπό του αρχείου δεδομένων.
- Short File Identifier (SFI): είναι ένας αριθμός των 5 bit που παίρνει τιμές μεταξύ του 01h και του 1Eh (1 ως 30 στο δεκαδικό σύστημα) και διανέμεται σε ένα αρχείο κατά τη δημιουργία του. Το SFI του system file είναι πάντα 01h αφού είναι και μοναδικό.

Το λειτουργικό σύστημα των καρτών GemClub υποστηρίζει τους τύπους αρχείων που φαίνονται στον πίνακα που ακολουθεί:

Τύπος Αρχείου	Τύπος αντικειμένου δεδομένων	Περιγραφή
System File	01h	Στοιχεία δεδομένων που χειρίζονται εσωτερικά από το λειτουργικό σύστημα
Record File	02h	Εγγραφές που χειρίζονται από το τερματικό
Counter	03h	Αποθηκεύει πόντους που έχουν δοθεί στον κάτοχο της κάρτας ως επιβράβευση για την εμπιστοσύνη του
Rule	04h	Μακροεντολές προς εκτέλεση πάνω σε μετρητές
Secret Code	05h	Μυστικός κωδικός
Secret Key	06h	Μυστική τιμή που χρησιμοποιείται από κρυπτογραφικούς αλγόριθμους

Πρέπει να σημειωθεί ότι για ένα συγκεκριμένο τύπο αρχείου, το SFI είναι μοναδικό. Δηλαδή για τα αρχεία μετρητών για παράδειγμα δεν μπορούν να υπάρχουν δύο μετρητές με SFI=02h. Δεν υπάρχει όμως κανένα πρόβλημα να υπάρχει ένας μετρητής και ένα secret code με το ίδιο SFI. Επίσης, ο μέγιστος αριθμός αρχείων του ίδιου τύπου που μπορούν να αποθηκευτούν στην κάρτα είναι 30 (εκτός από το αρχείο συστήματος που είναι μοναδικό). Γι' αυτό και το SFI παίρνει τιμές όπως είπαμε από το 01h ως το 1Eh (1 ως 30 στο δεκαδικό σύστημα).
Για κάθε αρχείο υπάρχουν κάποιες πληροφορίες προσωποποίησης (personalization data) που αποθηκεύονται στο ίδιο το αρχείο και χρησιμοποιούνται για τη διαχείρισή του. Τα στοιχεία τους θα αναλυθούν παρακάτω.

4.2.3.1 Αρχείο Συστήματος (System File)

Το αρχείο αυτό μπορεί να αποθηκεύσει μέχρι 12 byte πληροφορίας και περιλαμβάνει ένα μετρητή CTC (Card Transaction Counter), πληροφορίες του λειτουργικού συστήματος και πληροφορίες προσωποποίησης. Είναι το πρώτο αρχείο που δημιουργείται σε μία κάρτα. Τα στοιχεία του που παρατίθενται ακολούθως, πρέπει να καθοριστούν για τη σωστή και αποδοτική λειτουργία της κάρτας.

Μέγεθος	Περιεχόμενο	Λεπτομέρειες
1 byte	Access condition for Update / Delete	Ορίζει το δικαίωμα να ανανεωθεί ή να διαγραφεί το αρχείο συστήματος
1 byte	Access condition for Read	Ορίζει το δικαίωμα να διαβαστεί το αρχείο συστήματος
1 byte	Access condition for Create	Ορίζει το δικαίωμα να δημιουργηθεί ένα καινούριο αρχείο στην κάρτα
1 byte	PIN Code File Reference information	Ορίζει το SFI του Secret Code αρχείου που χρησιμοποιείται ως PIN
1 byte	EMV-DIR File Reference information	Ορίζει το SFI του αρχείου που χρησιμοποιείται για εξομοίωση EMV-DIR
1 byte	Card Transaction Counter	Χρησιμοποιείται για λειτουργίες πιστοποίησης και για ασφαλή μετάδοση μηνυμάτων (Secure Messaging)

Ένα δικαίωμα (access condition) μπορεί να ορίσει ότι το συγκεκριμένο αρχείο ή η συγκεκριμένη λειτουργία που σκοπεύει να προστατεύσει θα προστατεύεται από PIN, από Secret Code, ή από Secret Key (με Secure Messaging) ή θα είναι κλειδωμένο. Περισσότερες λεπτομέρειες για τα δικαιώματα αυτά θα εξηγηθούν σε επόμενη ενότητα.
Έτσι, το πρώτο στοιχείο του αρχείου συστήματος μπορεί να διαμορφωθεί ούτως ώστε να ορίζει ότι για να διαγραφεί ή να ανανεωθεί το αρχείο του συστήματος πρέπει η εφαρμογή να παρουσιάσει το secret code που βρίσκεται στο secret code file με SFI=02h (κοινώς το secret code 02).
Σε κάθε ένα από τα στοιχεία ενός αρχείου, όπως και στο αρχείο συστήματος, αντιστοιχεί μία μοναδική ετικέτα (tag) που βοηθάει το σύστημα να εντοπίσει και να προσδιορίσει τη πληροφορία τους αν αυτό ζητηθεί από την εφαρμογή. Τα στοιχεία αυτά μπορούν να μεταβληθούν από κάποια εφαρμογή με εντολές βιβλιοθήκης που χρησιμοποιούν το εκάστοτε tag για να προσδιορίσουν το στοιχείο που επιθυμούν να τροποποιήσουν.
Τα στοιχεία προσωποποίησης (personalization data) που αναφέραμε παραπάνω για το αρχείο συστήματος είναι τα 5 πρώτα και όπως βλέπουμε αφορούν κυρίως δικαιώματα πρόσβασης και αναφορές για στοιχεία ασφαλείας. Για να μπορούμε να μεταβάλλουμε όλα τα στοιχεία προσωποποίησης ενός αρχείου συγχρόνως (εξοικονομώντας έτσι χρόνο επεξεργασίας και μήκος κώδικα) έχει προβλεφθεί ένα συνολικό personalization tag.
Για παράδειγμα, στο αρχείο συστήματος, τα στοιχεία προσωποποίησης έχουν tag από 21h ως 26h και το personalization tag είναι 20h. Έτσι αντί να ζητήσει η εφαρμογή να κάνει ανανέωση στο στοιχείο με tag 21h, στο στοιχείο με tag 22h κ.ο.κ. (5 κλίσεις συνάρτησης ανανέωσης), θα ζητήσει να κάνει ανανέωση στα στοιχεία personalization με tag 20h (1 κλίση συνάρτησης ανανέωσης).
Αυτές οι ιδιότητες με τα στοιχεία προσωποποίησης ισχύουν για κάθε αρχείο της κάρτας και διευκολύνουν τη δουλειά του σχεδιαστή μίας εφαρμογής.
Πρέπει τέλος να αναφέρουμε ότι τα tags (με μήκος 1byte το κάθε ένα) για τα στοιχεία των αρχείων δεν αποθηκεύονται στην κάρτα.

4.2.3.2 Αρχείο Μετρητή (Counter File)

Το αρχείο αυτό αποτελείται από στοιχεία δεδομένων που χρησιμοποιούνται για να κάνουν απονομή (Award) ή αφαίρεση – αποζημίωση (Redeem) πόντων. Μέχρι 30 μετρητές μπορούν να υπάρχουν σε μία κάρτα. Οι μετρητές μπορούν να διαφέρουν ως προς τη δομή τους ανάλογα με τα χαρακτηριστικά που τους αποδίδουμε με το Structure byte (byte δομής).

Bit	Στοιχείο	Λεπτομέρειες στοιχείου	Μέγεθος
Bit 0	Cumulative Balance	Αριθμός πόντων που έχουν αποδοθεί στο μετρητή από τη δημιουργία του	3 bytes
Bit 1	Visit Counter	Αριθμός φορών που έχει γίνει απόδοση πόντων στο μετρητή από τη δημιουργία του	2 bytes
Bit 2	Validity Date – Award	Καθορίζει ημ/νίες εγκυρότητας για απόδοση πόντων	6 bytes
Bit 3	Validity Date –Redeem	Καθορίζει ημ/νίες εγκυρότητας για αφαίρεση πόντων	6 bytes
Bit 4	Rules Specified for Counter	Ορίζει τους κανόνες που μπορούν να εφαρμοστούν στον μετρητή	2 bytes
Bit 5	Label	Η αλφαριθμητική επιγραφή του μετρητή	8 bytes
Bit 6 -7	Reserved for Future Use	-	

Για να ενεργοποιηθεί κάθε ένα από αυτά τα στοιχεία για τον μετρητή, πρέπει το bit που του αντιστοιχεί στο Structure byte να έχει τη λογική τιμή 1. Έτσι αν επιθυμούμε ο μετρητής να έχει ενεργοποιημένη τη Cumulative Balance, τον Visit Counter και την Label, το Structure byte θα έχει τη τιμή 23h = 0010 $0011_{<2>}$.
Οι ημερομηνίες εγκυρότητας για απόδοση / αποζημίωση πόντων έχουν τη σημασία ότι ορίζονται ημ/νίες για τις οποίες επιτρέπεται να γίνουν αποδόσεις ή αποζημιώσεις πόντων μέσα στα πλαίσια συγκεκριμένης πολιτικής του προγράμματος εμπιστοσύνης. Αν για παράδειγμα μία αλυσίδα καταστημάτων προσφέρει έξυπνες κάρτες στους πελάτες της στα πλαίσια ενός προγράμματος εμπιστοσύνης το οποίο ξεκινάει μία εβδομάδα μετά την διανομή των καρτών και λήγει σε 12 μήνες από την ενεργοποίησή του, τότε αυτό το διάστημα θα οριστεί ως διάστημα εγκυρότητας.
Οι κανόνες που επιτρέπεται να επιδράσουν στον κάθε μετρητή "κωδικοποιούνται" μέσα σε 2 bytes (16 bits). Κάθε bit αντιστοιχεί στο SFI ενός rule (μπορούν να υπάρχουν μέχρι 16 rule files σε μία κάρτα) με αύξουσα σειρά από δεξιά προς τα αριστερά. Έτσι αν θέλουμε να επιτρέψουμε τους κανόνες 3 και 9 (με SFI 03h και 09h αντίστοιχα) για ένα μετρητή, τότε τα 2 bytes που χρησιμοποιούνται από τον μετρητή για να προσδιορίσουν τους επιτρεπόμενους κανόνες θα έχουν τη τιμή 01h 04h = 0000 0001 0000 $0100_{<2>}$.
Τα κύρια στοιχεία ενός αρχείου μετρητή που πρέπει να καθορίζονται είναι τα ακόλουθα.

Μέγεθος	Περιεχόμενο	Λεπτομέρειες
1 byte	Access condition for Update / Delete	Ορίζει το δικαίωμα να ανανεωθεί ή να διαγραφεί το αρχείο μετρητή
1 byte	Access condition for Read	Ορίζει το δικαίωμα να διαβαστεί το αρχείο μετρητή
1 byte	Key reference for Transaction Proof	SFI του secret key αρχείου που χρησιμοποιείται για τον υπολογισμό μίας απόδειξης συναλλαγής
1 byte	Access condition for Award	Ορίζει το δικαίωμα να γίνει απόδοση πόντων στον μετρητή
1 byte	Access condition for Redeem	Ορίζει το δικαίωμα να γίνει αφαίρεση πόντων από τον μετρητή
3 bytes	Balance	Δείχνει τον αριθμό των πόντων που είναι αποθηκευμένοι στον μετρητή

4.2.3.3 Αρχείο Κανόνα (Rule File)

Ένα αρχείο κανόνα είναι ένα σύνολο μακροεντολών που μπορούν να εκτελεστούν κατά τη χρήση της εντολής "Use Rule" (εφαρμογή κανόνα). Κάθε μακροεντολή αναφέρεται σε μία πράξη award ή redeem (απόδοση ή αφαίρεση πόντων) που θα γίνει σε ένα συγκεκριμένο μετρητή. Ένας κανόνας αποτελείται το πολύ από 4 μακροεντολές και συνεπακόλουθα ένας κανόνας μπορεί να συμπεριλάβει μέχρι 4 μετρητές στο "πεδίο δράσης" του.
Προφανώς ένας κανόνας δεν έχει νόημα ύπαρξης χωρίς καμία μακροεντολή και έτσι μία μακροεντολή είναι το ελάχιστο που μπορεί να υπάρχει σε ένα αρχείο κανόνα.
Τα στοιχεία που πρέπει να καθορίζονται για ένα αρχείο κανόνα φαίνονται ακολούθως:

Μέγεθος	Περιεχόμενο	Λεπτομέρειες
1 byte	Access condition for Update / Delete	Ορίζει το δικαίωμα να ανανεωθεί ή να διαγραφεί το αρχείο κανόνα
1 byte	Access condition for Read	Ορίζει το δικαίωμα να διαβαστεί το αρχείο κανόνα
1 byte	Key reference for Transaction Proof	SFI του secret key αρχείου που χρησιμοποιείται για τον υπολογισμό μίας απόδειξης συναλλαγής
1 byte	Access condition for Use Rule	Ορίζει το δικαίωμα να εκτελεστεί ο κανόνας με την εντολή Use Rule
1 byte	Version	Έκδοση του κανόνα
8 bytes	Macro Instruction 1	Ορίζει το είδος της πράξης που θα εκτελεστεί σε συγκεκριμένο μετρητή
. .	. .	. .
8 bytes	Macro Instruction 4	Ορίζει το είδος της πράξης που θα εκτελεστεί σε συγκεκριμένο μετρητή

Μία μακροεντολή δομείται ως εξής:

Λειτουργία (1 byte)	SFI μετρητή (1 byte)	Παράμετρος Α (3 bytes)	Παράμετρος Β (3 bytes)

Η λειτουργία αναφέρεται στην πράξη που θα γίνει σε ένα συγκεκριμένο counter κατά την εκτέλεση του κανόνα. Αν η πράξη είναι απόδοση πόντων (award) το byte αυτό παίρνει τη τιμή 01h, αν η πράξη είναι αποζημίωση πόντων (redeem) η τιμή του byte είναι 02h ενώ για να παρακαμφθεί η μακροεντολή αυτή (να μην εκτελεστεί καμία πράξη) το byte παίρνει τη τιμή 00h.
Το επόμενο στοιχείο της μακροεντολής είναι το SFI του μετρητή στον οποίο θα εκτελεστεί η πράξη της μακροεντολής.
Η παράμετρος Α είναι η πρώτη παράμετρος μετατροπής και αντιπροσωπεύει τον αριθμό των πόντων που θα απονεμηθούν ή θα αφαιρεθούν από τον μετρητή κάθε φορά που η τιμή που ορίζει η παράμετρος Β (δεύτερη παράμετρος μετατροπής) επιτευχθεί.
Όταν εκτελείται ο κανόνας, η κύρια παράμετρος της εντολής εκτέλεσής του είναι το ποσό συναλλαγής (amount) για το οποίο εκτελείται. Ο ακέραιος αριθμός των φορών που "χωράει" η παράμετρος Β στο ποσό συναλλαγής (amount) είναι και ο αριθμός με τον οποίο θα πολλαπλασιαστεί η παράμετρος Α για να υπολογιστούν οι πόντοι που θα αποδοθούν ή θα αφαιρεθούν από τον μετρητή. Η αντίστοιχη εξίσωση δίνεται από τον τύπο 4.2.3.3:

$$\text{Πόντοι} = (\text{ΠαράμετροςΑ}) \bullet Int\left(\frac{\text{Ποσό Συναλλαγής}}{\text{Παράμετρος Β}}\right) \qquad (4.2.3.3)$$

Η παράμετρος Β και το ποσό συναλλαγής έχουν την ίδια μονάδα μέτρησης.
Έτσι, αν για παράδειγμα μία εταιρία διανομής πετρελαίου θέρμανσης εφαρμόζει για τους τακτικούς πελάτες της ένα πρόγραμμα εμπιστοσύνης κατά το οποίο για κάθε 800 λίτρα παραγγελίας πετρελαίου ο πελάτης κερδίζει 16 πόντους, τότε αν ένας πελάτης παραγγείλει 2600 λίτρα θα κερδίσει: $16 \bullet Int\left(\frac{2600\text{λίτρα}}{800\text{λίτρα}}\right) = 16 \bullet 3 = 48\,\text{πόντους}$,

όπου προφανώς Παράμετρος Α = 16, Παράμετρος Β = 800, Ποσό συναλλαγής = 2600.

4.2.3.4 Αρχείο Μυστικού Κωδικού (Secret Code File)

Οι μυστικοί κωδικοί χρησιμοποιούνται για προστασία αρχείων. Ένα secret code αρχείο περιέχει ένα μόνο secret code, μεγέθους 8 byte. Για να προστατεύει ένας secret code ένα αρχείο, θα πρέπει να δηλώνεται ως μέσο προστασίας στο δικαίωμα πρόσβασης του αρχείου αυτού (access condition). Τα στοιχεία του αρχείου που πρέπει να περιλαμβάνονται είναι:

Μέγεθος	Περιεχόμενο	Λεπτομέρειες
1 byte	Access condition for Update / Delete	Ορίζει το δικαίωμα να ανανεωθεί ή να διαγραφεί το secret code αρχείο
1 byte	Maximum Attempts Allowed / Attempts Remaining	Ορίζει το μέγιστο επιτρεπτό αριθμό και τον εναπομείναντα αριθμό προσπαθειών παρουσίασης του κωδικού πριν η κάρτα απορριφθεί και ο κωδικός μπλοκαριστεί
8 bytes	Secret Code Value	Ορίζει τη τιμή του μυστικού κωδικού

Όπως βλέπουμε δεν υπάρχει το στοιχείο δικαιώματος ανάγνωσης του αρχείου αυτού γιατί δεν επιτρέπεται.
Ο μέγιστος επιτρεπτός αριθμός συνεχόμενων προσπαθειών παρουσίασης του κωδικού είναι πάντα μεγαλύτερος ή ίσος από τον εναπομείναντα αριθμό προσπαθειών παρουσίασης του κωδικού. Και οι δύο δεν μπορούν να πάρουν τιμή μεγαλύτερη του $15_{<10>}$. Η μείωση των εναπομεινάντων προσπαθειών κατά τη διάρκεια λανθασμένης παρουσίασης του κωδικού γίνεται με τη χρήση ενός ειδικού μετρητή επικύρωσης (Secret Code Ratification Counter) ο οποίος περιέχει και τα δύο παραπάνω στοιχεία.

4.2.3.5 Αρχείο Μυστικού Κλειδιού (Secret Key File)

Ένα secret key αρχείο περιέχει ένα μόνο secret key, μεγέθους 16 byte. Τα στοιχεία του αρχείου αυτού που πρέπει να προσδιορίζονται είναι τα εξής:

Μέγεθος	Περιεχόμενο	Λεπτομέρειες
1 byte	Access condition for Update / Delete	Ορίζει το δικαίωμα να ανανεωθεί ή να διαγραφεί το secret key αρχείο
1 byte	Maximum Attempts Allowed / Attempts Remaining	Ορίζει το μέγιστο επιτρεπτό αριθμό και τον εναπομείναντα αριθμό προσπαθειών παρουσίασης του κλειδιού πριν η κάρτα απορριφθεί και το κλειδί μπλοκαριστεί
8 bytes	Secret Key Value	Ορίζει το πρώτο μισό της τιμής που δίνεται για το 3DES κλειδί
8 bytes	Secret Key Value	Ορίζει το δεύτερο μισό της τιμής που δίνεται για το 3DES κλειδί

Τα κλειδιά αυτά (secret keys) χρησιμοποιούνται για την εκτέλεση του Secure Messaging (ασφαλούς μετάδοσης μηνυμάτων) και για τον υπολογισμό των αποδείξεων συναλλαγής (Transaction Proof) που θα εξετάσουμε σε επόμενη ενότητα. Όπως και με τα secret codes, για να χρησιμοποιηθεί ένα secret key ως μέσο προστασίας πρέπει να αναφέρεται στο δικαίωμα πρόσβασης του στοιχείου που θέλει να προστατεύσει.
Σε αντιστοιχία με τον Secret Code Ratification Counter των secret code αρχείων, υπάρχει ο Secret Key Ratification Counter. Οι ιδιότητες του μέγιστου επιτρεπτού αριθμού και του εναπομείναντος αριθμού προσπαθειών παρουσίασης του κλειδιού είναι ίδιες με των αντιστοίχων ενός secret code αρχείου.

4.2.3.6 Αρχείο Εγγραφής (Record File)

Ένα αρχείο εγγραφών χρησιμοποιείται για την αποθήκευση δεδομένων. Οι GemClub κάρτες υποστηρίζουν γραμμικά αρχεία με εγγραφές μεταβλητού μήκους. Κάθε αρχείο εγγραφών (record file) περιέχει εγγραφές (records). Οι εγγραφές ορίζονται με αύξοντα αριθμό από το 1 ως το 254, δηλαδή ένα record file μπορεί να περιέχει μέχρι 254 records και το πρώτο record παίρνει τον αριθμό 1.
Το μέγεθος κάθε record εξαρτάται από τις συνθήκες προστασίας του αρχείου και πρέπει να είναι πάντα πολλαπλάσιο των 4 bytes. Αν το αρχείο προστατεύεται με secure messaging το μέγεθος κάθε record δε μπορεί να ξεπερνάει τα 24 bytes. Σε οποιαδήποτε άλλη περίπτωση το μέγιστο μέγεθος ενός record είναι τα 255 bytes.
Τα στοιχεία που πρέπει να περιέχονται σε ένα αρχείο εγγραφών είναι:

Μέγεθος	Περιεχόμενο	Λεπτομέρειες
1 byte	Access condition for Update / Delete	Ορίζει το δικαίωμα να ανανεωθεί ή να διαγραφεί το record αρχείο
1 byte	Access condition for Read	Ορίζει το δικαίωμα να διαβαστεί το record αρχείο
L_1	Record 1	Δεδομένα
. .	. .	. .
L_n	Record N	Δεδομένα

Κάθε record περιέχει ένα byte συστήματος (system byte), δεδομένα και αν χρειάζεται rounding bytes για να είναι το μέγεθός του πολλαπλάσιο του 4 (bytes).
Έτσι αν έχουμε 25 bytes πληροφορίας και δε χρησιμοποιείται secure messaging, το μέγεθος του record θα είναι 28 bytes = 1(SB) + 25(data) + 2(rounding bytes).
Αντίστοιχα, κάθε αρχείο εγγραφών περιέχει 8 bytes για τον file descriptor και τα διάφορα records του. Έτσι αν ένα αρχείο εγγραφών περιέχει 3 records με μέγεθος 16 bytes, 32 bytes και 92 bytes αντίστοιχα, το μέγεθος του αρχείου θα είναι:
148 bytes = 8 (file descriptor) + 16 (record 1) + 32 (record 2) + 92 (record 3).

4.2.3.7 EMV-DIR File

Το αρχείο αυτό είναι στην ουσία ένα αρχείο εγγραφών (record file) το οποίο χρησιμοποιείται για την εξομοίωση με το σύστημα συναλλαγών EMV. Το EMV είναι ένα πρότυπο που έχει θεσμοθετηθεί για τις χρηματικές συναλλαγές με έξυπνες κάρτες σε POS (Point Of Sale) terminals και ATMs και υποστηρίζει βασικά την επιλογή εφαρμογής (π.χ. e-purse ή loyalty) μέσω ονόματος εφαρμογής (Select application file by name).
Το record file που χρησιμοποιείται από την κάρτα για το EMV-DIR δηλώνεται στο αρχείο συστήματος και έχει συγκεκριμένη δομή που καθορίζεται από τα πρότυπα EMV.
Τα χαρακτηριστικά EMV χρησιμοποιούνται μόνο στις κάρτες GemClub-EMV.

4.2.4 Συνθήκες πρόσβασης (Access Conditions)

Οι συνθήκες (δικαιώματα) πρόσβασης σε αρχεία και εντολές είναι το μέσο προστασίας που προσφέρουν οι GemClub κάρτες δίνοντας υψηλό επίπεδο ασφαλείας. Ρυθμίζονται μετά τη δημιουργία των αρχείων στην κάρτα και πριν την παράδοσή της στους τελικούς χρήστες.
Οι λειτουργίες που υποβάλλονται για έλεγχο πρόσβασης είναι οι ακόλουθες:

- Create: δημιουργία αρχείων (system, counter, rule, secret code/key, record files)
- Update/Delete: ανανέωση ή διαγραφή αρχείων (system, counter, rule, secret code, secret key, record files)
- Read: ανάγνωση αρχείων (system, counter, rule, record files)–δεν υποστηρίζεται SM
- Award: εκτέλεση εντολής απόδοσης πόντων (counter files)
- Redeem: εκτέλεση εντολής αποζημίωσης πόντων (counter files)
- Use Rule: εκτέλεση κανόνα (rule files)

Μία συνθήκη πρόσβασης καθορίζεται σε 1 byte με τον ακόλουθο τρόπο:

b7	b6 b5	b4 b3 b2 b1 b0
PIN	Secret Code / Key	SFI
1: χρήση PIN 0: μη χρήση PIN	00: μη χρήση Secret Code/SM προστασίας 01: χρήση Secret Code προστασίας 10: χρήση SM προστασίας 00: απαγόρευση πρόσβασης	SFI του Secret Code / Key αρχείου που θα χρησιμοποιηθεί για το τύπο προστασίας που ορίστηκε στα προηγούμενα bit

Η ασφαλής μετάδοση μηνυμάτων (Secure Messaging) γίνεται με τη χρήση μυστικών κλειδιών (Secret Keys). Δε μπορεί να υπάρχει ταυτόχρονα προστασία παρουσίασης Secret Code και Secure Messaging (SM).
Αν λοιπόν επιθυμούμε ένα δικαίωμα πρόσβασης να ορίζει προστασία με παρουσίαση PIN και Secure Messaging με τη χρήση του Secret Key 02 (με SFI=02h) τότε το byte του access condition θα έχει τη τιμή C2h = 1100 $0010_{<2>}$.
Αντίστοιχα, αν επιθυμούμε προστασία χωρίς PIN και με παρουσίαση του Secret Code 5 (με SFI=05h), τότε το byte του access condition θα είναι 25h = 0010 $0101_{<2>}$.
Πρέπει να τονιστεί ότι για να χρησιμοποιηθεί ένα Secret Code ως PIN (Personal Identification Number του κατόχου της κάρτας) πρέπει να έχει οριστεί το SFI του στο αρχείο συστήματος (PIN Code File Reference Information) αλλιώς όποια συνθήκη πρόσβασης περιλαμβάνει παρουσίαση του PIN θα την παρακάμψει.

4.2.5 Secure Messaging

Όπως έχει ειπωθεί, η ασφαλής μετάδοση μηνυμάτων (Secure Messaging) βασίζεται στα Secret Keys (μυστικά κλειδιά) και χρησιμοποιείται για να εξασφαλίσει την αυθεντικότητα και την ακεραιότητα των δεδομένων που ανταλλάσσονται μεταξύ του host συστήματος ή του τερματικού και της κάρτας. Χρησιμοποιεί τον 3DES αλγόριθμο. Χρησιμεύει επίσης για την πιστοποίηση της ταυτότητας μεταξύ τερματικού και κάρτας.
Όταν προστατεύεται η εκτέλεση μίας εντολής για παράδειγμα, το τερματικό στέλνει ένα κωδικό πιστοποίησης μηνύματος (Message Authentication Code = MAC_IN) στη κάρτα, η οποία εξετάζει το κωδικό αυτό και αν είναι σωστός, αν δηλαδή το τερματικό έχει τη δικαιοδοσία να εκτελέσει την εντολή που ζητάει και αν η μεταφορά των δεδομένων έχει γίνει σωστά, εκτελεί την εντολή και στέλνει στο τερματικό ένα αντίστοιχο κωδικό MAC_OUT. Το εκάστοτε MAC εμπεριέχει το Secret Key που έχει οριστεί ότι θα προστατεύει την εντολή.
Στην περίπτωση που η εντολή είναι η ανανέωση της τιμής ενός κλειδιού, η τιμή αυτή κρυπτογραφείται με τη μέθοδο 3DES πριν αποσταλεί από το τερματικό.

4.2.6 Application Protocol Data Unit (APDU)

Οι κάρτες GemClub δέχονται εντολές και απαντήσεις με μορφή συμβατή με την Application Data Protocol (Πρωτόκολλο Δεδομένων Εφαρμογής) διάταξη που ορίζεται από το πρότυπο ISO 7816-4. Αυτό επιτρέπει στις συγκεκριμένες εντολές που υποστηρίζουν οι κάρτες GemClub να είναι συμβατές με τη βιβλιοθήκη της διεπαφής του reader (π.χ. για την εντολή βιβλιοθήκης G4_ExchangeApdu που αναφέρθηκε στην ενότητα 4.1.2, σελ.31).
Το στρώμα μεταφοράς του πρωτοκόλλου της GemClub είναι συμβατό με το πρότυπο επικοινωνίας ISO 7816-3 T=0 που συζητήσαμε σε προηγούμενη ενότητα, σύμφωνα με το οποίο τα μηνύματα APDU μετατρέπονται σε μηνύματα TPDU (Transport Protocol Data Unit). Τα μηνύματα TPDU είναι συμβατά με τις εντολές ISO IN και ISO OUT της διεπαφής του reader, οι οποίες όπως έχει αναφερθεί χρησιμοποιούνται για την επικοινωνία της κάρτας με το reader και το host σύστημα. Έτσι ο reader στέλνει ISO IN / ISO OUT TPDU εντολές στη κάρτα και αυτή στέλνει TPDU απαντήσεις στον reader.
Η διάταξη μίας εντολής που λαμβάνει η κάρτα έχει την ακόλουθη μορφή:

Header				**Body**		
CLA	INS	P1	P2	Lc	Parameters/Data	Le
Τάξη–Τύπος εντολής	Κωδικός εντολής	Παράμετροι εντολής (διευθυνσοδότησης)		Μήκος πεδίου δεδομένων (Data)	Παράμετροι – Δεδομένα εντολής	Αναμενόμενο μήκος δεδομένων απάντησης

Το κομμάτι header έχει τη μορφή που ορίζει το πρωτόκολλο επικοινωνίας T=0 (ενότητα 3.7) και τα πεδία του είναι υποχρεωτικά. Το κομμάτι body (σώμα εντολής) είναι προαιρετικό.
Η διάταξη μίας απάντησης που στέλνει η κάρτα έχει την ακόλουθη μορφή:

Body	**Trailer**
Data	SW1, SW2

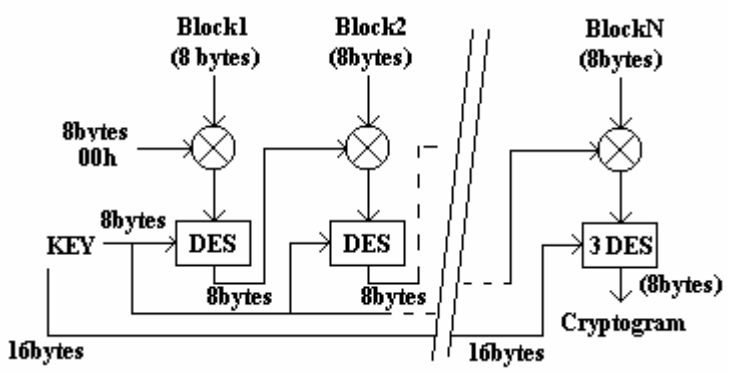
Το σώμα της απάντησης είναι προαιρετικό. Το κομμάτι trailer είναι υποχρεωτικό και περιέχει δύο bytes κατάστασης (status bytes) που δείχνουν την κατάσταση της κάρτας μετά την εκτέλεση της εντολής.
Παράδειγμα με τη μορφή συγκεκριμένης εντολής και απάντησης ακολουθεί στην ενότητα 4.2.10.

4.2.7 Message Authentication Code (MAC)

Ένας κωδικός πιστοποίησης μηνύματος (MAC) χρησιμοποιείται για την ασφαλή μετάδοση μηνυμάτων (Secure Messaging). Υπάρχουν δύο κύριες μορφές MAC, το MAC_IN και το MAC_OUT που μεταδίδονται προς και από την κάρτα αντίστοιχα.
Κάθε MAC χαρακτηρίζεται από μία ετικέτα tag (σαν τα tags των στοιχείων κάθε αρχείου) ανάλογα με την λειτουργία του. Έτσι αν δημιουργείται ένα MAC_IN για να προστατεύσει την εντολή απόδοσης πόντων (award) χαρακτηρίζεται από το TAG_MAC_IN_AWARD ενώ αν δημιουργείται ως απάντηση μίας εντολής εκτέλεσης κανόνα (use rule) η οποία προστατεύεται με secure messaging, τότε η ετικέτα του είναι το TAG_MAC_OUT_USE_RULE.
Η μορφή ενός MAC είναι:

MAC_IN / MAC_OUT		
Secret Key	TAG	MAC data and padding

Το πεδίο Secret Key (16 bytes) είναι το κλειδί που αναφέρεται στο δικαίωμα πρόσβασης της λειτουργίας για την οποία δημιουργείται το MAC. Το πεδίο data and padding αποτελείται από μπλοκ των 8 byte που περιέχουν τη τιμή του μετρητή CTC της κάρτας, το header της εντολής και το αντίστοιχο σώμα της. Όπως και στην περίπτωση των records, για να είναι τα μπλοκ πληροφορίας του πεδίου αυτού μεγέθους 8 byte, συμπληρώνονται όπου χρειάζεται με rounding-padding bytes. Κάθε μπλοκ περιέχει 1 byte που αντιστοιχεί στο tag του MAC και πρέπει οπωσδήποτε σε ένα MAC να υπάρχει μετά τα δεδομένα της εντολής ένα byte με τιμή 80h. Αν χρειάζονται padding bytes παίρνουν τη 00h. Το κομμάτι MAC data and padding είναι αυτό που κωδικοποιείται με τον αλγόριθμο 3DES. Αναλυτικό παράδειγμα υπολογισμού ενός MAC θα δοθεί στην ενότητα 4.2.10.



Σχήμα 4.2.7 – Διαδικασία Υπολογισμού MAC

Στο Σχήμα 4.2.7 φαίνεται η διαδικασία υπολογισμού ενός MAC. Τα 8 MSB (Most Significant Bytes) του Secret Key που περιλαμβάνεται στο MAC χρησιμοποιούνται για να κωδικοποιήσουν κατά τον κρυπτογραφικό αλγόριθμο DES όλα τα μπλοκ πληροφορίας του MAC (τμήμα data and padding) εκτός του τελευταίου μπλοκ, το οποίο κωδικοποιείται κατά τον αλγόριθμό 3DES με τη χρήση όλου του Secret Key (και των 16 byte). Το κάθε μπλοκ δεν μπαίνει κατευθείαν ως πηγή δεδομένων για την εκτέλεση του αλγορίθμου DES αλλά περνάει πρώτα από μία λογική πύλη XOR μαζί με το αποτέλεσμα-κρυπτόγραμμα της προηγούμενης εκτέλεσης του αλγορίθμου (με το προηγούμενο μπλοκ δεδομένων). Εξαιρείται το πρώτο μπλοκ που περνάει από την πύλη XOR με μία σειρά 8 byte τιμής 00h. Το τελικό κρυπτόγραμμα έχει μήκος 8 bytes.
Στην ασφαλή μετάδοση μηνυμάτων η διαδικασία που ακολουθείται ξεκινάει με τον υπολογισμό από το τερματικό-reader του MAC_IN, το οποίο στέλνεται στην κάρτα. Η κάρτα δημιουργεί μόνη της το ίδιο MAC χρησιμοποιώντας το κλειδί που έχει αποθηκευμένο στο σώμα της και το συγκρίνει με αυτό που παρέλαβε από το τερματικό. Αν τα δύο αυτά ταυτίζονται, εκτελεί την εντολή που της στάλθηκε και δημιουργεί ένα MAC_OUT με το ίδιο Secret Key, το στέλνει στο τερματικό και αυτό με τη σειρά του εξετάζει την ορθότητα του MAC_OUT.

4.2.8 Απόδειξη Συναλλαγής (Transaction Proof)

Η απόδειξη συναλλαγής είναι ένα πιστοποιητικό που υπολογίζεται στο τέλος συγκεκριμένων συναλλαγών με τη κάρτα (απόδοση / αφαίρεση πόντων ή εκτέλεση κανόνα) και χρησιμοποιείται για να αποδείξει την αυθεντικότητα της συναλλαγής στον διαχειριστή της εφαρμογής. Υπολογίζεται από την κάρτα μετά την ολοκλήρωση μίας συναλλαγής και αποστέλλεται στο τερματικό, το οποίο το προωθεί στον διαχειριστή της εφαρμογής μαζί με πληροφορίες για την συναλλαγή.
Έχει δομή ίδια με αυτή του MAC_IN, όπου σαν Secret Key χρησιμοποιεί ένα μοναδικό κλειδί που έχει οριστεί στο αρχείο συστήματος για το σκοπό αυτό και περιλαμβάνει μία

ετικέτα tag και μπλοκ των 8 byte με τις πληροφορίες της συναλλαγής μεταξύ άλλων. Το κλειδί που χρησιμοποιείται για τον υπολογισμό ενός transaction proof είναι γνωστό μόνο στην κάρτα και στον διαχειριστή της εφαρμογής, γι' αυτό και το τερματικό δεν μπορεί να ελέγξει την ορθότητα της απόδειξης συναλλαγής.

4.2.9 Εντολές

Οι εντολές που υποστηρίζονται από το λειτουργικό σύστημα της κάρτας GemClub χωρίζονται σε δύο κατηγορίες, στις εντολές Διαχείρισης που χρησιμοποιούνται για τη διαχείριση της κάρτας και στις εντολές Εφαρμογής που χρησιμοποιούνται για να εκτελέσουν λειτουργίες στη κάρτα. Σε κάθε εντολή αντιστοιχεί διαφορετικός συνδυασμός των παραμέτρων CLA, INS, P1, P2, Lc, και Le που αποτελούν μία εντολή.

4.2.9.1 Εντολές Διαχείρισης (Administrative Commands)

Υπάρχουν 3 εντολές διαχείρισης που υποστηρίζονται από τις κάρτες GemClub.

- Create Object: δημιουργία ενός νέου αρχείου στην μνήμη εφαρμογών της κάρτας. Στη διάρκεια αυτής της διαδικασίας, αφιερώνεται ένας χώρος στη μνήμη EEPROM και ένα νέο αρχείο δημιουργείται εκεί.
- Delete Object: διαγραφή ενός αρχείου από τη μνήμη εφαρμογών και απελευθέρωση χώρου στη μνήμη EEPROM.
- Select Communication Speed: αλλαγή της ταχύτητας επικοινωνίας.

Αν η δημιουργία ενός αρχείου προστατεύεται από κάποια συνθήκη πρόσβασης που δεν ικανοποιηθεί ή αν το αρχείο υπάρχει ήδη, τότε η εντολή δεν εκτελείται και επιστρέφει μήνυμα λάθους. Η εντολή διαγραφής ενός αρχείου-στοιχείου αποτυγχάνει αν το στοιχείο δεν υπάρχει ή αν δεν ικανοποιείται η συνθήκη πρόσβασης.

4.2.9.2 Εντολές Εφαρμογής (Application Commands)

Οι εντολές εφαρμογής που υποστηρίζονται από τις κάρτες GemClub παρατίθενται ακολούθως. Είναι αυτές κυρίως που σχετίζονται με τα χαρακτηριστικά σχεδιασμού του κάθε τύπου κάρτας. Δηλαδή μία κάρτα που έχει σχεδιασθεί κυρίως για προγράμματα εμπιστοσύνης χρησιμοποιεί συγκεκριμένες εντολές για να υποστηρίξει τον στόχο της, όπως οι εντολές απόδοσης / αφαίρεσης πόντων, οι οποίες ανήκουν στις εντολές εφαρμογής.

- Append Record: δημιουργεί μία νέα εγγραφή (record) σε ένα υπάρχον αρχείο εγγραφών (record file) και αρχικοποιεί τα περιεχόμενά του. Ανάλογα με το αν χρησιμοποιείται Secure Messaging ή όχι διαφέρει η παράμετρος CLA του header της εντολής και το μήκος της αναμενόμενης απάντησης από την κάρτα. Η εντολή απορρίπτεται αν δεν περιλαμβάνεται ή δεν είναι σωστό κάποιο από τα στοιχεία που είναι απαραίτητα (π.χ. το SFI του αρχείου εγγραφών στο οποίο θα προστεθεί η νέα εγγραφή) ή αν κάποια συνθήκη πρόσβασης δεν ικανοποιείται.
- Award: απόδοση πόντων σε ένα μετρητή της κάρτας. Με την εκτέλεση της εντολής μεταβάλλεται (αυξάνεται) η balance του μετρητή και αυξάνονται η Cumulative Balance και ο Visit Counter (αν έχουν οριστεί να περιλαμβάνονται). Επίσης αν έχει οριστεί στα στοιχεία του μετρητή, θα δημιουργηθεί μία απόδειξη συναλλαγής που θα σταλεί στο τερματικό. Η εντολή απορρίπτεται αν δεν ικανοποιείται η συνθήκη πρόσβασης ή αν λείπει κάποιο από τα υποχρεωτικά στοιχεία, όπως επίσης αν ο έλεγχος εγκυρότητας ημερομηνίας δεν είναι επιτυχής.
- Get Response: αυτή η εντολή χρησιμοποιείται για να μεταδώσει APDU responses από τη κάρτα στο τερματικό.

- Read Parameter: ένα στοιχείο πληροφορίας ή πληροφορίες για την κάρτα, όπως ο σειριακός αριθμός της κάρτας (Card Serial Number) που είναι μοναδικός και αποδίδεται σε μία κάρτα κατά την κατασκευή της. Ανάλογα με το αν θα διαβαστούν πληροφορίες για την ίδια τη κάρτα ή κάποιο στοιχείο πληροφορίας αποθηκευμένο σε αρχείο στην κάρτα, οι παράμετροι P1 και P2 του header της εντολής παίρνουν διαφορετική τιμή. Η εντολή επιστρέφει μήνυμα λάθους αν η συνθήκη πρόσβασης δεν ικανοποιείται ή αν το στοιχείο πληροφορίας που προσπαθούμε να αναγνώσουμε δεν υπάρχει.
- Read Record: διαβάζει μία εγγραφή (record) από ένα αρχείο εγγραφών. Η εγγραφή που επιθυμούμε να διαβάσουμε προσδιορίζεται μέσα στο αρχείο από τον αύξοντα αριθμό που της έχει αποδοθεί κατά τη δημιουργία της. Η εντολή επιστρέφει μήνυμα λάθους αν η συνθήκη πρόσβασης δεν ικανοποιείται, αν η εγγραφή δεν υπάρχει ή αν δεν προσδιορίζεται το αρχείο εγγραφών από το οποίο θα γίνει η ανάγνωση.
- Redeem: αφαίρεση πόντων-μονάδων από ένα μετρητή της κάρτας. Με την εκτέλεση της εντολής μεταβάλλεται (μειώνεται) η balance του μετρητή και αυξάνονται η Cumulative Balance και ο Visit Counter (αν έχουν οριστεί να περιλαμβάνονται). Όπως και στην περίπτωση της απόδοσης πόντων, αν έχει οριστεί στα στοιχεία του μετρητή, θα δημιουργηθεί μία απόδειξη συναλλαγής που θα σταλεί στο τερματικό. Η εντολή απορρίπτεται αν δεν ικανοποιείται η συνθήκη πρόσβασης, αν λείπει κάποιο από τα υποχρεωτικά στοιχεία, ή αν ο έλεγχος εγκυρότητας ημερομηνίας δεν είναι επιτυχής.
- Select AF by Name: εξομοιώνει την εντολή "Select File" του EMV προτύπου για να υπάρχει συμβατότητα με τις EMV ιδιότητες επιλογής εφαρμογής σε εφαρμογές πληρωμής. Μπορεί να εκτελεστεί μόνο σε GemClub-EMV κάρτες. Απορρίπτεται αν το Record File που χρησιμοποιείται ως EMV-DIR αρχείο δεν έχει την απαραίτητη διάταξη ή αν δεν υπάρχει η αναφορά του στο αρχείο συστήματος.
- Update Parameter: ανανεώνει-μεταβάλλει ένα στοιχείο πληροφορίας, αποθηκευμένο σε αρχείο της κάρτας. Ανάλογα με το αν χρησιμοποιείται Secure Messaging ή όχι διαφέρει η παράμετρος CLA του header της εντολής και το μήκος της αναμενόμενης απάντησης από την κάρτα. Απορρίπτεται αν κάποιο από τα στοιχεία της εντολής δεν περιλαμβάνεται ή δεν είναι σωστό ή αν η συνθήκη πρόσβασης δεν ικανοποιείται.
- Update Record: μεταβάλλει το περιεχόμενο μίας εγγραφής (record) από ένα αρχείο εγγραφών. Η εγγραφή που επιθυμούμε να τροποποιήσουμε προσδιορίζεται μέσα στο αρχείο από τον αύξοντα αριθμό που της έχει αποδοθεί κατά τη δημιουργία της. Η εντολή απορρίπτεται αν η συνθήκη πρόσβασης δεν ικανοποιείται, αν η εγγραφή δεν υπάρχει ή αν δεν προσδιορίζεται το αρχείο εγγραφών στο οποίο ανήκει η εν λόγω εγγραφή.
- Use Rule: εκτελεί ένα κανόνα, δηλαδή μία ακολουθία μακροεντολών που επιδρούν πάνω σε μετρητές της κάρτας. Αν έχει οριστεί στα στοιχεία του κανόνα, θα δημιουργηθεί μία απόδειξη συναλλαγής που θα σταλεί στο τερματικό. Απορρίπτεται αν η συνθήκη πρόσβασης δεν ικανοποιείται ή αν κάποιο από τα στοιχεία που χρειάζονται για να τρέξει η εντολή δεν υπάρχει ή δεν είναι έγκυρο.
- Verify Secret Code: συγκρίνει ένα μυστικό κωδικό αποθηκευμένο στη κάρτα με ένα κωδικό που αποστέλλεται με την εντολή από το τερματικό. Μπορεί να χρησιμοποιηθεί για να ελεγχθεί πόσες προσπάθειες παρουσίασης του κωδικού απομένουν πριν η κάρτα απορριφθεί και ο κωδικός της κάρτας κλειδωθεί. Απορρίπτεται η εκτέλεσή της αν κάποιο από τα στοιχεία της εντολής δεν είναι σωστό.

4.2.10 Παραδείγματα Σύνταξης - Χρήσης Εντολών

Θα δοθούν 2 παραδείγματα χρήσης των εντολών που αναφέρθηκαν στην προηγούμενη ενότητα τα οποία θα βοηθήσουν στην κατανόηση της σύνταξής τους και των παραμέτρων τους.

4.2.10.1 Παράδειγμα 1

Θα χρησιμοποιήσουμε την εντολή Update Parameter για να μεταβάλλουμε το δικαίωμα πρόσβασης για ανανέωση / διαγραφή (Update/Delete access condition) του αρχείου κανόνα με SFI=03h (Rule File 3). Η παρούσα συνθήκη πρόσβασης για ανανέωση / διαγραφή του αρχείου αυτού ορίζει ότι πρέπει να παρουσιασθεί ο μυστικός κωδικός 4 (Secret Code File με SFI=04h) και επιθυμούμε να το μεταβάλλουμε ούτως ώστε να απαιτείται πλέον η παρουσίαση του PIN του κατόχου και να εκτελείται Secure Messaging με το μυστικό κλειδί 2 (Secret Key με SFI=02h).
Η δομή της εντολής Update Parameter είναι η ακόλουθη:

Πεδίο	Περιγραφή
CLA	80h
INS	DEh
P1	Data Element Tag – το tag του στοιχείου που επιθυμούμε να μεταβάλλουμε
P2	SFI του αρχείου στο οποίο βρίσκεται το στοιχείο που επιθυμούμε να μεταβάλλουμε
Lc	10h + μέγεθος του στοιχείου που επιθυμούμε να μεταβάλλουμε
Data	Terminal Data – Στοιχεία Τερματικού (8 bytes) Data Element Value – νέα τιμή του στοιχείου που επιθυμούμε Incoming MAC or Secret Code – 8 bytes
Le	0Ah

Το tag του access condition για Update/Delete ενός Rule File έχει την τιμή 81h (προκύπτει από πίνακα με όλα τα tags που παρατίθεται στο Παράρτημα Β).
Το SFI του αρχείου κανόνα του οποίου θέλουμε να αλλάξουμε τη συνθήκη πρόσβασης για ανανέωση / διαγραφή έχει την τιμή 03h.
Όπως έχουμε πει, ένα δικαίωμα πρόσβασης έχει μήκος 1 byte, οπότε Lc = 10h + 01h bytes, δηλαδή Lc = 16 + 1 = 17 bytes.
Για κάθε τερματικό μπορεί να οριστεί μία ταυτότητα που αποτελείται από 8 bytes και ονομάζεται Terminal Data. Ας ορίσουμε στο παράδειγμα αυτό ότι:
Terminal Data = 01h 23h 45h 67h 89h ABh CDh EFh.
Αφού επιθυμούμε η συνθήκη πρόσβασης να απαιτεί την παρουσίαση του PIN και να περιλαμβάνει Secure Messaging με το Secret Key 2 (αρχείο με SFI=02h), το byte της συνθήκης πρόσβασης θα έχει τη τιμή C2h = 1100 $0010_{<2>}$, σύμφωνα με τους κανόνες που αναλύσαμε στην ενότητα 4.2.4.
Η υπάρχουσα συνθήκη πρόσβασης για ανανέωση / διαγραφή του αρχείου κανόνα προσδιορίζει ότι πρέπει για οποιαδήποτε αλλαγή στο αρχείο να παρουσιάσουμε τον μυστικό κωδικό 4. Έστω λοιπόν ότι αυτός ο κωδικός είναι ο 2Ah E4h 11h 8Ch 09h F5h 36h 7Bh.
Έτσι η εντολή που στέλνει το τερματικό στην κάρτα στο παράδειγμα αυτό έχει την ακόλουθη μορφή:

Header				Body		
CLA	INS	P1	P2	Lc	Parameters/Data	Le
80h	DEh	81h	03h	11h	01h 23h 45h 67h 89h ABh CDh EFh C2h 2Ah E4h 11h 8Ch 09h F5h 36h 7Bh	0Ah

Η δομή της απάντησης στην εντολή αυτή είναι:

Πεδίο	Περιγραφή
Data	Card Transaction Counter – 2 bytes Outgoing MAC – 8 bytes αν έχει χρησιμοποιηθεί Secure Messaging αλλιώς 0 bytes
SW1–SW2	Status bytes

Στην απάντηση της κάρτας προς το τερματικό, αποστέλλεται η τρέχουσα τιμή του CTC μετρητή της κάρτας μετά την επεξεργασία της εντολής, το MAC_OUT αν έχει χρησιμοποιηθεί Secure Messaging και τα byte κατάστασης που δηλώνουν την κατάσταση της κάρτας μετά την εκτέλεση της εντολής.
Στη περίπτωσή μας, δεν αποστέλλεται MAC-OUT και το πεδίο Data έχει μέγεθος 2 bytes. Αν η εκτέλεση της εντολής έχει γίνει με επιτυχία τα byte κατάστασης που επιστρέφονται είναι 61h 00h.

4.2.10.2 Παράδειγμα 2

Θα χρησιμοποιήσουμε την εντολή Redeem για να αφαιρέσουμε 34 πόντους από το μετρητή με SFI 12h (μετρητής 18). Η εντολή Redeem στο μετρητή αυτό προστατεύεται με Secure Messaging με τη χρήση του μυστικού κλειδιού 13 (Secret Key File με SFI=0Dh). Επίσης έχει οριστεί ότι θα υπολογίζεται απόδειξη συναλλαγής για τις λειτουργίες award / redeem σε αυτό το μετρητή με τη χρήση του μυστικού κλειδιού 5 (Secret Key File με SFI=05h).
Δεχόμαστε και σε αυτή την περίπτωση ότι τα χαρακτηριστικά του τερματικού προσδιορίζουν τιμή για την παράμετρο Terminal Data = 01h 23h 45h 67h 89h ABh CDh EFh.
Επίσης σε αυτή την εντολή χρειάζεται σαν είσοδος η τρέχουσα ημερομηνία για να μπορέσει να συγκριθεί με τις ημερομηνίες εγκυρότητας που μπορεί να υπάρχουν στο μετρητή. Οι ημερομηνίες κωδικοποιούνται σε 3 bytes σε μορφή BCD με διάταξη YY.MM.DD (2 ψηφία δηλαδή για το έτος, δύο ψηφία για το μήνα και τέλος 2 ψηφία για την ημέρα). Χρησιμοποιούνται δηλαδή για το έτος τα δύο τελευταία ψηφία.
Για να αποφευχθεί πρόβλημα με το έτος 2000 που μπορεί να μπερδευτεί με το 1900 κ.ό.κ. έχει γίνει μία σύμβαση ούτως ώστε το έτος 2000 να μεταφράζεται σε 03, το 2001 να μεταφράζεται σε 04 κ.ό.κ.
Έτσι αν η τρέχουσα ημερομηνία είναι 19-10-2004, θα μεταφρασθεί για να αποσταλεί στην κάρτα σε 3 bytes ως 07h 0Ah 13h.
Η δομή της εντολής Redeem είναι η ακόλουθη:

Πεδίο	Περιγραφή
CLA	80h
INS	4Eh
P1	02h
P2	SFI του μετρητή από τον οποίο θα αφαιρεθούν οι πόντοι
Lc	16h
Data	Terminal Data - Στοιχεία Τερματικού (8 bytes) Current Date – Τρέχουσα Ημερομηνία Amount – το ποσό (αριθμός πόντων) που θα αφαιρεθεί από το μετρητή Incoming MAC or Secret Code – 8 bytes
Le	0Ah

Αυτό που μένει να υπολογιστεί είναι το MAC_IN που θα σταλεί με την εντολή για να ικανοποιήσει την συνθήκη πρόσβασης για τη λειτουργία Redeem στο συγκεκριμένο μετρητή. Το tag για το συγκεκριμένο MAC_IN είναι το TAG_MAC_IN_REDEEM και έχει τιμή 21h που προκύπτει από πίνακα με όλα τα MAC tags ο οποίος παρατίθεται στο Παράρτημα Β.
Η δομή του πεδίου Mac data and padding του συγκεκριμένου MAC είναι σχηματικά η ακόλουθη, σε συμφωνία με όσα αναφέρθηκαν στην ενότητα 4.2.7.

MAC data and padding	Block1= [Tag $_{1\ byte}$ || CTC $_{2\ bytes}$ || CLA $_{1\ byte}$ || INS $_{1\ byte}$ || P1 $_{1\ byte}$ || P2 $_{1\ byte}$ || Lc $_{1\ byte}$] Block2= [Tag $_{1\ byte}$ || Terminal Data $_{7\ bytes}$] Block3= [Tag $_{1\ byte}$ || Terminal Data $_{1\ byte}$ || Current Date $_{3\ bytes}$ || Amount $_{3\ bytes}$] Block4= [Tag $_{1\ byte}$ || 80h 00h 00h 00h 00h 00h 00h]

Θέτοντας το ποσό που θα αφαιρεθεί από το μετρητή με SFI 12h στη τιμή 22h=$34_{<10>}$, θεωρώντας ότι ο CTC έχει τιμή 00h 05h = $5_{<10>}$ και αντικαθιστώντας τα μεγέθη προκύπτουν τα ακόλουθα μπλοκ δεδομένων στο MAC.

MAC data and padding	Block1= [21h 00h 05h 80h 4Eh 02h 12h 16h] Block2= [21h 01h 23h 45h 67h 89h ABh CDh] Block3= [21h EFh 07h 0Ah 13h 00h 00h 22h] Block4= [21h 80h 00h 00h 00h 00h 00h 00h]

Τα 4 block δεδομένων του MAC αυτού θα επεξεργαστούν σύμφωνα με το Σχήμα 4.2.7 για να προκύψει το κρυπτόγραμμα που επιθυμούμε. Συγκεκριμένα θα προκύψουν τα ακόλουθα αποτελέσματα:
Result 1 = DES (Block1, MSB Secret Key 13)
Result 2 = DES (Block2 ⊗ Result1, MSB Secret Key 13)
Result 3 = DES (Block3 ⊗ Result2, MSB Secret Key 13)
MAC_IN = 3DES (Block4 ⊗ Result3, Secret Key 13)

Το κρυπτόγραμμα αυτό έχει μέγεθος 8 bytes.
Η μορφή της απάντησης (response) στην εντολή Redeem που εκτελέσαμε είναι:

Πεδίο	Περιγραφή
Data	Card Transaction Counter – 2 bytes Transaction Proof – 8 bytes New Balance of Counter – 3 bytes Outgoing MAC – 8 bytes αν έχει χρησιμοποιηθεί Secure Messaging αλλιώς 0 bytes
SW1–SW2	Status bytes

Στο παράδειγμα αυτό, στην απάντηση της κάρτας προς το τερματικό, αποστέλλεται η τρέχουσα τιμή του CTC μετρητή της κάρτας μετά την επεξεργασία της εντολής, το MAC_OUT που υπολογίζεται από τη κάρτα με τη χρήση του μυστικού κλειδιού 13, η απόδειξη συναλλαγής (transaction proof) που υπολογίζεται από την κάρτα με τη χρήση του μυστικού κλειδιού 5 και τα byte κατάστασης που δηλώνουν την κατάσταση της κάρτας μετά την εκτέλεση της εντολής.
Όπως έχει αναφερθεί, η απόδειξη συναλλαγής δημιουργείται όπως και ένα MAC. Το tag της απόδειξης συναλλαγής για τη συγκεκριμένη λειτουργία είναι το TAG_PROOF_REDEEM και έχει τιμή 23h.
Η δομή του πεδίου data and padding του συγκεκριμένου transaction proof είναι σχηματικά η ακόλουθη, σε συμφωνία με τη δομή των MAC.

data and padding	Block1= [Tag $_{1\ byte}$ || CTC $_{2\ bytes}$ || Counter SFI $_{1\ byte}$ || Terminal Data $_{4\ bytes}$] Block2= [Tag $_{1\ byte}$ || Terminal Data $_{4\ bytes}$ || Current Date $_{3\ bytes}$] Block3= [Tag $_{1\ byte}$ || Transaction Amount $_{3\ bytes}$ || 80h 00h 00h 00h]

Τα υπόλοιπα βήματα του υπολογισμού του transaction proof είναι ίδια με αυτά του MAC_IN που υπολογίστηκε παραπάνω. Τέλος, ο υπολογισμός του MAC_OUT ακολουθεί την ίδια διαδικασία και στο τερματικό μεταδίδεται ένα κρυπτογραφημένο μήνυμα.
Με τα παραπάνω παραδείγματα, έγινε μία παρουσίαση της διαδικασίας επικοινωνίας μεταξύ τερματικού και κάρτας, των συγκεκριμένων εντολών που χαρακτηρίζουν τις κάρτες GemClub και κυρίως των διαδικασιών ασφαλείας που προσφέρουν υψηλή ποιότητα και αυθεντικότητα στην ανταλλαγή εντολών και δεδομένων. Είναι βεβαίως προφανές ότι για τις εφαρμογές αυτές που χρησιμοποιούν Secure Messaging, πρέπει να υπάρχει τρόπος υπολογισμού των MAC, δηλαδή πρέπει η εφαρμογή να περιλαμβάνει τον κώδικα του κρυπτογραφικού αλγορίθμου DES και 3DES.

4.2.11 GemClub Interface Library

Η βιβλιοθήκη διεπαφής της GemClub έχει αναπτυχθεί για να παρέχει τρόπο διεπαφής μεταξύ μίας εφαρμογής και των καρτών μέσω των readers της Gemplus. Οι συναρτήσεις που περιλαμβάνει μπορούν να κλιθούν από ένα πρόγραμμα γραμμένο είτε σε C είτε σε Visual Basic κάτω από ένα περιβάλλον Windows. Η βιβλιοθήκη έχει στόχο να προσφέρει στο σχεδιαστή εφαρμογής ευκολία χρήσης και πρόσβασης στις ιδιότητες των καρτών. Μπορεί να λειτουργήσει με όλους τους Gemplus readers και είναι σχεδιασμένη ειδικά για τις κάρτες GemClub.
Έτσι, ενώ η βιβλιοθήκη της διεπαφής των GemCore reader (ενότητα 4.1.2) έχει σχεδιαστεί ούτως ώστε να επικοινωνεί μία εφαρμογή μέσω των συγκεκριμένων GemCore reader με οποιαδήποτε κάρτα, η βιβλιοθήκης της διεπαφής των GemClub καρτών έχει σχεδιαστεί ούτως ώστε να επικοινωνεί μία εφαρμογή μέσω οποιουδήποτε reader της Gemplus με τις συγκεκριμένες GemClub κάρτες.
Προφανώς λοιπόν, όταν μία εφαρμογή βασίζεται σε λειτουργίες του reader (μπορεί για παράδειγμα να τρέχει εφαρμογή πάνω στον ίδιο τον reader, σε ειδικό χώρο της μνήμης του) και επιθυμεί να επικοινωνεί με διάφορους τύπους καρτών χωρίς να έχουν τόση σημασία τα ιδιαίτερα χαρακτηριστικά τους, θα χρησιμοποιήσει την βιβλιοθήκη διεπαφής του reader.
Στην αντίθετη περίπτωση όπου η εφαρμογή βασίζεται στις ξεχωριστές ιδιότητες της κάρτας και χρησιμοποιεί τον reader απλά ως μέσο διαβίβασης εντολών, θα χρησιμοποιηθεί κυρίως η βιβλιοθήκη της διεπαφής της κάρτας.
Στην περίπτωση των τριών εφαρμογών που εξετάζονται στη παρούσα διπλωματική εργασία και αναλύονται στα επόμενα κεφάλαια, χρησιμοποιήθηκε ως βάση η βιβλιοθήκη διεπαφής των καρτών GemClub.
Οι συναρτήσεις-εντολές που περιλαμβάνει η βιβλιοθήκη αυτή παρατίθενται ακολούθως.

- G_AwardEnh: απόδοση πόντων σε ένα μετρητή της κάρτας.
- G_CreateObjectEnh: δημιουργία ενός νέου αρχείου στην κάρτα.
- G_DeleteObjectEnh: διαγραφή ενός αρχείου από την κάρτα.
- G_RedeemEnh: αφαίρεση πόντων από ένα μετρητή της κάρτας.
- G_SMAppendRecordEnh: δημιουργία νέας εγγραφής (record) σε ένα υπάρχον αρχείο εγγραφών (record file) και αρχικοποίηση των περιεχομένων του χρησιμοποιώντας Secure Messaging.
- G_SMUpdateRecordEnh: μεταβολή του περιεχομένου μίας εγγραφής (record) σε ένα αρχείο εγγραφών με χρήση Secure Messaging.
- G_SMUpdateKeyEnh: μεταβολή της τιμής ενός μυστικού κλειδιού με χρήση Secure Messaging.

- G_UpdateParamEnh: μεταβολή ενός στοιχείου πληροφορίας, αποθηκευμένου σε αρχείο της κάρτας.
- G_UseRuleEnh: εκτέλεση ενός κανόνα πάνω σε μετρητές της κάρτας.

Οι παραπάνω συναρτήσεις που επιδρούν στις κάρτες, δεν διαφέρουν στο αντικείμενό τους από τις αντίστοιχες εντολές που συντάσσονται με την κλασσική μορφή των TPDU μηνυμάτων προς την κάρτα (χρήση εντολών ISO_IN και ISO_OUT). Διαφέρουν όμως στο ότι λειτουργούν ως προς τους κανόνες ασφαλείας με dual mode, δηλαδή η δομή τους είναι ίδια είτε υπάρχουν συνθήκες πρόσβασης είτε όχι και κυρίως στο ότι δεν χρειάζεται ο υπολογισμός των MAC και Transaction Proof γιατί γίνεται εσωτερικά και κατευθείαν από την συνάρτηση. Το μόνο που χρειάζεται να αναφερθεί είναι το Secret Key που χρησιμοποιείται για Secure Messaging και τα υπόλοιπα βήματα γίνονται κατευθείαν από την συνάρτηση. Επίσης δεν χρειάζεται να εκτελείται η εντολή Get Response για να παραληφθεί η απάντηση της κάρτας και να χωριστεί στις αντίστοιχες παραμέτρους, αφού αυτό γίνεται αυτόματα μέσω της συνάρτησης. Περισσότερες λεπτομέρειες θα δούμε σε παράδειγμα που ακολουθεί στην ενότητα 4.2.11.1.
Οι ακόλουθες συναρτήσεις είναι αντίστοιχες αυτών που αναφέρθηκαν ανωτέρω, με τη διαφορά ότι το πεδίο data του τμήματος body μίας ISO_IN TPDU εντολής και το πεδίο data του τμήματος body μίας ISO_OUT TPDU εντολής (απάντηση της κάρτας) δεν αναλύονται στα στοιχεία τους στην ίδια τη συνάρτηση αλλά χρειάζονται συμπληρωματικές συναρτήσεις BuildDataIn και BuildDataOut για να δέσουν τα επιμέρους στοιχεία στο στοιχείο data ή να εξάγουν από το στοιχείο data τα επιμέρους στοιχεία αντίστοιχα. Σε αυτή την περίπτωση συναρτήσεων, χρειάζεται ο υπολογισμός MAC και Transaction Proof εκτός των συναρτήσεων.

- G_Award: απόδοση πόντων σε ένα μετρητή της κάρτας.
- G_BuildAwardDataIn: δημιουργία του πεδίου data του σώματος της εντολής Award.
- G_BuildAwardDataOut: εξαγωγή των στοιχείων δεδομένων του πεδίου data της απάντησης της κάρτας στην εντολή Award.
- G_CreateObject: δημιουργία ενός νέου αρχείου στην κάρτα.
- G_BuildCreateObjectDataIn: δημιουργία του πεδίου data του σώματος της εντολής Create Object.
- G_BuildCreateObjectDataOut: εξαγωγή των στοιχείων δεδομένων του πεδίου data της απάντησης της κάρτας στην εντολή Create Object.
- G_DeleteObject: διαγραφή ενός αρχείου από την κάρτα.
- G_BuildDeleteObjectDataIn: δημιουργία του πεδίου data του σώματος της εντολής Delete Object.
- G_Redeem: αφαίρεση πόντων από ένα μετρητή της κάρτας.
- G_BuildRedeemDataIn: δημιουργία του πεδίου data του σώματος της εντολής Redeem.
- G_BuildRedeemDataOut: εξαγωγή των στοιχείων δεδομένων του πεδίου data της απάντησης της κάρτας στην εντολή Redeem.
- G_UpdateParam: μεταβολή ενός στοιχείου πληροφορίας της κάρτας.
- G_BuildUpdateParamDataIn: δημιουργία του πεδίου data του σώματος της εντολής Update Parameter.
- G_BuildUpdateParamDataOut: εξαγωγή των στοιχείων δεδομένων του πεδίου data της απάντησης της κάρτας στην εντολή Update Parameter.
- G_UseRule: εκτέλεση ενός κανόνα πάνω σε μετρητές της κάρτας.
- G_BuildUseRuleDataIn: δημιουργία του πεδίου data του σώματος της εντολής Use Rule.
- G_BuildUseRuleDataOut: εξαγωγή των στοιχείων δεδομένων του πεδίου data της απάντησης της κάρτας στην εντολή Use Rule.

- G_SMAppendRecord: δημιουργία νέας εγγραφής (record) σε ένα υπάρχον αρχείο εγγραφών και αρχικοποίηση των περιεχομένων του με χρήση Secure Messaging.
- G_BuildSMAppendRecordDataIn: δημιουργία του πεδίου data του σώματος της εντολής Append Record με χρήση Secure Messaging.
- G_BuildSMAppendRecordDataOut: εξαγωγή των στοιχείων δεδομένων του πεδίου data της απάντησης της κάρτας στην εντολή Append Record με χρήση Secure Messaging.
- G_SMUpdateRecord: μεταβολή του περιεχομένου μίας εγγραφής (record) σε ένα αρχείο εγγραφών με χρήση Secure Messaging.
- G_BuildSMUpdateRecordDataIn: δημιουργία του πεδίου data του σώματος της εντολής Update Record με χρήση Secure Messaging.
- G_BuildSMUpdateRecordDataOut: εξαγωγή των στοιχείων δεδομένων του πεδίου data της απάντησης της κάρτας στην εντολή Update Record με χρήση Secure Messaging.

Τέλος στην βιβλιοθήκη περιέχονται και οι συναρτήσεις που ακολουθούν, οι οποίες είτε δεν υποστηρίζουν Secure Messaging, είτε ανήκουν στην κατηγορία των εντολών διαχείρισης, είτε τέλος είναι εντολές βοηθητικές (οι τελευταίες) και έχουν γενικά λιγότερες παραμέτρους και πιο απλή μορφή.

- G_ReadParam: διαβάζει ένα στοιχείο πληροφορίας στην κάρτα.
- G_GetResponse: η εντολή αυτή χρησιμοποιείται για να παραλάβει ο reader responses μετά από διάφορες εντολές.
- G_AppendRecord: δημιουργία νέου record σε ένα αρχείο εγγραφών και αρχικοποίηση των περιεχομένων του.
- G_ReadRecord: διαβάζει μία εγγραφή από ένα αρχείο εγγραφών.
- G_SelectByName: εξομοιώνει την EMV εντολή “Select File”.
- G_UpdateRecord: μεταβολή του περιεχομένου μίας εγγραφής σε ένα αρχείο εγγραφών.
- G_VerifySecretCode: σύγκριση ενός μυστικού κωδικού αποθηκευμένου στην κάρτα με ένα κωδικό που αποστέλλεται με την εντολή από το τερματικό.
- G_SetParam: αλλαγή της ταχύτητας επικοινωνίας.
- G_Bld_AC: δημιουργία συνθήκης πρόσβασης.
- G_Bld_Macro: δημιουργία μακροεντολής.
- G_ComputeMAC: υπολογισμός MAC κρυπτογράμματος.
- G_OpenSession: ανοίγει σύνοδο της κάρτας με το host σύστημα.
- G_UpdateKey: μεταβολή της τιμής ενός μυστικού κλειδιού.

4.2.11.1 Παράδειγμα 1

Θα δώσουμε ένα παράδειγμα χρήσης των εντολών βιβλιοθήκης της διεπαφής των καρτών GemClub. Συγκεκριμένα θα θεωρήσουμε την ίδια περίπτωση με αυτή του Παραδείγματος 2 της ενότητας 4.2.10.2.
Έτσι θα χρησιμοποιήσουμε τη συνάρτηση G_RedeemEnh για να αφαιρέσουμε 34 πόντους από το μετρητή με SFI 12h (μετρητής 18). Η εντολή Redeem στο μετρητή αυτό προστατεύεται με Secure Messaging με τη χρήση του μυστικού κλειδιού 13 (Secret Key File με SFI=0Dh). Επίσης έχει οριστεί ότι θα υπολογίζεται απόδειξη συναλλαγής (Transaction Proof) για τις λειτουργίες award / redeem σε αυτό το μετρητή με τη χρήση του μυστικού κλειδιού 5 (Secret Key File με SFI=05h).
Η συνάρτηση G_RedeemEnh δέχεται 10 παραμέτρους ως είσοδο, δηλαδή 10 παράμετροι δίνουν τη τιμή τους στη συνάρτηση, ενώ 8 παράμετροι ενημερώνονται ως έξοδος της συνάρτησης, παίρνουν ή μεταβάλλουν τη τιμή τους δηλαδή μετά την εκτέλεση της συνάρτησης – εντολής.
Η δήλωση της συνάρτησης δίνεται ακολούθως:

```
INT16 G_DECL G_RedeemEnh
(
const WORD16 ChannelNb,
const WORD16 Class,
const WORD16 CounterId,
const BYTE * const TerminalData,
const WORD32 TerminalDataLen,
const WORD32 CurrentDate,
const WORD32 Amount,
const BYTE * const SecretCodeOrKey,
const WORD32 SecretCodeOrKeyLen,
WORD32 * const CTCValue,
BYTE * const Proof,
WORD32 * const ProofLen,
WORD32 * const Balance,
BYTE * const MacIn,
WORD32 * const MacInLen,
BYTE * const MacOut,
WORD32 * const MacOutLen
);
```

Η παράμετρος ChannelNb δείχνει το λογικό νούμερο που αντιστοιχεί στο κανάλι επικοινωνίας που έχει ανοίξει μεταξύ host συστήματος και reader. Η τιμή της παραμέτρου είναι γνωστή σε μία εφαρμογή μετά το άνοιγμα του καναλιού. Η παράμετρος Class (τάξη της συνάρτησης) είναι για τη συγκεκριμένη συνάρτηση 80h. Η παράμετρος CounterId δείχνει τη ταυτότητα του μετρητή από τον οποίο θα γίνει αφαίρεση πόντων και στο παράδειγμα αυτό είναι το SFI 12h.
Δεχόμαστε και σε αυτή την περίπτωση ότι τα χαρακτηριστικά του τερματικού ορίζουν για την παράμετρο TerminalData = 01h 23h 45h 67h 89h ABh CDh EFh. Το μήκος της παραμέτρου είναι TerminalDataLen = 08h. Έστω ότι η τρέχουσα ημερομηνία είναι Current Date=07h 0Ah 13h (19-10-2004). Ο αριθμός των πόντων που θα αφαιρεθούν από τον μετρητή είναι Amount = $34_{<10>}$ = 22h.
Αφού στο παράδειγμα χρειάζεται το Secret Key 5 για Secure Messaging για την προστασία της εντολής Redeem θα θέσουμε στην παράμετρο SecretCodeOrKey τη τιμή του Secret Key 5 και η παράμετρος SecretCodeOrKeyLen θα είναι ίση με $16_{<10>}$=10h.
Η παράμετρος CTCValue περιέχει τη τιμή του μετρητή CTC της κάρτας η οποία μπορεί να διαβαστεί νωρίτερα με την εκτέλεση μίας εντολής Read Parameter από την εφαρμογή. Με αυτά τα δεδομένα καλούμε την συνάρτηση ως εξής:

```
G_RedeemEnh(ChannelNb,Class,CounterId,TerminalData,TerminalDataLen,
            CurrentDate,Amount,SecretCodeOrKey,SecretCodeOrKeyLen,
            &CTCValue,Proof,&ProofLen,&Balance,MacIn,&MacInLen,
            MacOut,&MacOutLen);
```

Με την εκτέλεση της συνάρτησης ενημερώνεται η τιμή της παραμέτρου CTCValue και των μεταβλητών Proof, ProofLen, Balance, MacIn, MacInLen, MacOut, MacOutLen.

5

Σχεδίαση Τριών Εφαρμογών

5.1 Εφαρμογή Μισθοδοσίας

Η πρώτη εφαρμογή που σχεδιάστηκε πάνω στη τεχνολογία των έξυπνων καρτών GemClub αφορά ένα πρόγραμμα μισθοδοσίας, ένα πρόγραμμα δηλαδή υπολογισμού του μισθού που αντιστοιχεί σε ένα εργαζόμενο για την εργασία του μέσα σε συγκεκριμένο χρονικό διάστημα. Η εφαρμογή αυτή περιέχει διάφορες παραμέτρους όπως η υπερωριακή εργασία και έχει δομηθεί με τέτοιο τρόπο ούτως ώστε να αντιστοιχεί σε πραγματικά δεδομένα.
Το πρόγραμμα θεωρήθηκε ότι εφαρμόζεται σε ένα πανεπιστημιακό ίδρυμα και απευθύνεται σε όλους τους εργαζομένους του. Σε αυτούς περιλαμβάνονται καθηγητές, εργαζόμενοι σε γραμματείες, κτίρια διοίκησης ή στη βιβλιοθήκη, καθώς επίσης και διδακτορικοί φοιτητές ή εξωτερικοί συνεργάτες σε μελέτες-προγράμματα που εκτελούνται εντός του πανεπιστημιακού ιδρύματος, σε χώρους όπως τα εργαστήρια.
Λόγω του σχεδιασμού του και του τρόπου υλοποίησής του όμως, το πρόγραμμα αυτό μπορεί να εφαρμοστεί κάλλιστα σε διάφορες εταιρίες, δημόσιες υπηρεσίες, τραπεζικά ιδρύματα κ.ό.κ. αφού η βασική δομή του ως προς τον υπολογισμό των ωρών εργασίας κάθε μέρα δε χρειάζεται αλλαγές. Οι μόνες αλλαγές έχουν να κάνουν με παραμέτρους του τελικού υπολογισμού του μισθού των εργαζομένων, όπως συντελεστές προσαύξησης για την υπερωριακή εργασία ή οι προβλεπόμενες ώρες κανονικής εργασίας.
Για παράδειγμα, ενώ σε ένα χώρο εργαστηρίου του πανεπιστημίου, οι υπεύθυνοι-βοηθοί δεν χρειάζεται να προσέλθουν συγκεκριμένη ώρα στο εργαστήριο αλλά πρέπει να συμπληρώσουν 4 ώρες εργασίας, σε μία τράπεζα οι εργαζόμενοι πρέπει να εισέρχονται μέχρι τις 8 το πρωί και να συμπληρώνουν 8 ώρες εργασίας. Ο τρόπος όμως που η εφαρμογή υπολογίζει το χρονικό διάστημα μεταξύ εισόδου και εξόδου του χρήστη και στις δύο περιπτώσεις είναι ο ίδιος.
Ακολούθως θα εξετάσουμε τις προδιαγραφές της εφαρμογής αυτής όσον αφορά τα τεχνικά χαρακτηριστικά αλλά και το περιεχόμενό της.

5.1.1 Τεχνικές Προδιαγραφές

Στο χώρο εργασίας στον οποίο έχει πρόσβαση ο εργαζόμενος, υπάρχει υπολογιστής ο οποίος έχει το ρόλο του host συστήματος της εφαρμογής. Στον υπολογιστή αυτό συνδέεται ο reader GCR410, μέσω του οποίου γίνεται η επικοινωνία με τις κάρτες των εργαζομένων. Την εφαρμογή χειρίζεται εξουσιοδοτημένος διαχειριστής ο οποίος κατέχει δική του κάρτα.
Σε κάθε εργαζόμενο αντιστοιχεί μία κάρτα με προσωπικά του στοιχεία και άλλες πληροφορίες της εφαρμογής αποθηκευμένα σε αυτή. Ο εργαζόμενος κατά την είσοδό του στο χώρο εργασίας ή την έξοδό του από αυτή, εισάγει την κάρτα του στον reader, ο οποίος "ειδοποιεί" την εφαρμογή για το γεγονός αυτό. Η εφαρμογή τότε εκτελεί συγκεκριμένες διαδικασίες, ανάλογα με τα δεδομένα που διαβάζει από την κάρτα και το σύστημα και όταν τελειώσει την επεξεργασία, ειδοποιεί το χρήστη να αφαιρέσει την κάρτα από τον reader.
Για τον συνολικό υπολογισμό του μισθού του ένας εργαζόμενος πρέπει να προσέλθει στο διαχειριστή της εφαρμογής, να εισάγει την κάρτα του στο reader και όταν η εφαρμογή τελειώσει με την ανάγνωση και επεξεργασία των δεδομένων της, να την αφαιρέσει από αυτόν. Σε περίπτωση οποιουδήποτε προβλήματος απευθύνεται πάντα στον διαχειριστή της εφαρμογής ο οποίος έχει πρόσβαση στα δεδομένα της κάρτας.

5.1.2 Προγραμματιστικές Προδιαγραφές

Οι λειτουργίες τις οποίες πρέπει η εφαρμογή μισθοδοσίας να υποστηρίζει, ορίζουν συγκεκριμένες προδιαγραφές για το περιεχόμενο των καρτών, τα δεδομένα της εφαρμογής, και την ασφάλεια επικοινωνίας.
Οι κυριότερες από αυτές, με βάση τις οποίες έγινε η σχεδίαση της εφαρμογής, παρατίθενται ακολούθως. Έτσι το πρόγραμμα μισθοδοσίας πρέπει να προσφέρει:

- Πιστοποίηση ταυτότητας διαχειριστή προγράμματος: το πρόγραμμα θα πρέπει να ελέγχει ότι ο χρήστης που θα διαχειριστεί την εφαρμογή είναι εξουσιοδοτημένος. Το πρόγραμμα θα ξεκινάει μόνο μέσω του διαχειριστή.
- Πιστοποίηση ταυτότητας κατόχου κάρτας: το πρόγραμμα θα ζητάει απόδειξη ότι ο χρήστης της κάρτας είναι ο νόμιμος κάτοχός της.
- Δυνατότητα ασφαλούς παρέμβασης του διαχειριστή στην εφαρμογή, αν αυτό χρειαστεί: θα πρέπει σε διάφορα σημεία του προγράμματος που χρειάζεται παρεμβολή του διαχειριστή να πιστοποιείται εκ νέου η ταυτότητά του.
- Ασφαλής επικοινωνία και ανταλλαγή δεδομένων μεταξύ τερματικού – host system (στο οποίο τρέχει η εφαρμογή) και καρτών: θα χρησιμοποιηθεί κάποια τεχνική ασφαλείας για να είναι εξασφαλισμένη η εγκυρότητα της επικοινωνίας μεταξύ των δύο πλευρών.
- Συνεχής ροή προγράμματος: το πρόγραμμα εισόδου - εξόδου χρηστών στο χώρο εργασίας, θα πρέπει να είναι συνέχεια έτοιμο για εισαγωγή καρτών στο reader, επικοινωνία με αυτές και εκτέλεση λειτουργιών, εκτός απρόοπτου.
- Υπολογισμός χρονικής διαφοράς μεταξύ εισόδου και εξόδου ενός χρήστη: η εφαρμογή πρέπει να μετράει με ακρίβεια το χρονικό διάστημα μεταξύ μίας εισόδου χρήστη και της αμέσως επόμενης εξόδου του από το χώρο εργασίας.
- Εμφάνιση χρόνου εισόδου, χρόνου εξόδου και διάρκειας παραμονής στο χώρο εργασίας για ένα χρήστη: το πρόγραμμα θα εμφανίζει την ώρα εισόδου κάθε χρήστη όταν αυτός εισέρχεται στο χώρο εργασίας, ενώ όταν εξέρχεται θα εμφανίζει την ώρα εξόδου και το χρόνο παραμονής του στο χώρο εργασίας από την προηγούμενη είσοδό του.
- Δυνατότητα κατανόησης αλλαγής ημερομηνίας του συστήματος: το πρόγραμμα θα πρέπει να έχει τη "δυνατότητα" να διαβάζει από το σύστημα την εκάστοτε ημερομηνία.
- Δυνατότητα πολλαπλών εισόδων και εξόδων μέσα σε μία ημέρα χωρίς απώλεια ή σύγχυση πληροφορίας: ένας χρήστης μπορεί να εισέρχεται και να εξέρχεται από το χώρο εργασίας του πάνω από μία φορά στη διάρκεια μίας ημέρας, χτυπώντας κάθε φορά την κάρτα του, και το πρόγραμμα θα πρέπει να υπολογίζει κάθε φορά το νέο σύνολο ωρών εργασίας του για την ημέρα αυτή.
- Υποστήριξη πολλαπλών χρηστών: η εφαρμογή θα πρέπει να αντιλαμβάνεται την ταυτότητα του κατόχου της κάρτας και να μπορεί να λειτουργήσει με πολλαπλές εισόδους και εξόδους διαφορετικών χρηστών.
- Έλεγχος ημερομηνίας εξόδου σε σχέση με ημερομηνία εισόδου του χρήστη: θα πρέπει στο πρόγραμμα να γίνεται έλεγχος αν η ημερομηνία εξόδου ενός εργαζομένου ταυτίζεται με την ημερομηνία εισόδου του στο χώρο εργασίας. Αυτό χρειάζεται για να μπορούν να υπολογιστούν σωστά οι ώρες εργασίας του χρήστη, αν αυτός έχει τύχει να ξεχάσει κάποια μέρα να "χτυπήσει" την κάρτα του κατά την έξοδό του.
- Υπολογισμός υπερωριών: ανάλογα με την πολιτική μισθοδοσίας και υπερωριών, θα πρέπει το πρόγραμμα να υπολογίζει πόσες ώρες εργασίας ενός χρήστη στη διάρκεια μίας ημέρας ανήκουν στις κανονικές ώρες εργασίας και πόσες είναι υπερωρίες.
- Υπολογισμός και εμφάνιση στο τέλος συγκεκριμένης χρονικής περιόδου (συνήθως ενός μήνα) των συνολικών ωρών εργασίας, κανονικής και υπερωριακής, και του αντίστοιχου μισθού για ένα χρήστη: στο τέλος του μήνα που ο εργαζόμενος πηγαίνει

με την κάρτα του στο διαχειριστή του προγράμματος για να υπολογιστεί ο μισθός του, θα πρέπει να εμφανίζονται οι συνολικές ώρες εργασίας του και το σύνολο του μισθού που θα παραλάβει.

- Συνυπολογισμός στο τελικό μισθό ενός χρήστη των ημερών αδείας που έχει χρησιμοποιήσει στο διάστημα υπολογισμού του μισθού: θα πρέπει το πρόγραμμα να λαμβάνει υπόψη του τις ημέρες που ένας εργαζόμενος δεν έχει προσέλθει στο χώρο εργασίας του λόγω αδείας και να προσμετρά τις ημέρες αυτές ως μέρες κανονικής εργασίας στους υπολογισμούς του τελικού μισθού.
- Δυνατότητα προκαταβολής μισθού: ο εργαζόμενος θα πρέπει να έχει τη δυνατότητα να πάρει μία προκαταβολή του μισθού του, ανάλογα με τις ώρες που έχει δουλέψει μέχρι τότε μέσα στο μήνα.
- Ύπαρξη ορίων προκαταβολής μισθού και ωρών υπερωρίας ανά μήνα: θα υπάρχουν όρια για το ποσό της προκαταβολής που μπορεί ένας εργαζόμενος να λάβει και για το ύψος των υπερωριών που μπορεί να κάνει σε ένα μήνα.
- Έλεγχος εγκυρότητας ως προς τα όρια ωρών υπερωρίας, ημερών αδείας, ποσού προκαταβολής: το πρόγραμμα θα πρέπει να ελέγχει κατά τον υπολογισμό του τελικού μισθού του εργαζομένου να μην ξεπεραστούν τα όρια που αναφέρθηκαν πριν.
- Προσωποποίηση κάρτας ανά χρήστη: κάθε κάρτα θα περιέχει στοιχεία ταυτότητας του κατόχου τα οποία το πρόγραμμα θα πρέπει να αντιλαμβάνεται.
- Αναφορά λαθών που μπορεί να προκύψουν: το πρόγραμμα θα πρέπει να αναφέρει τυχόν λάθη κατά την εκτέλεση οποιασδήποτε λειτουργίας.
- Δυνατότητα εντοπισμού και επιδιόρθωσης σφαλμάτων που τυχόν παρουσιάζονται: πέρα από την αναφορά των λαθών, η εφαρμογή θα πρέπει να έχει τη δυνατότητα αντιμετώπισής τους και επιδιόρθωσης τους.
- Ανάκτηση εφαρμογής μετά από μη αναμενόμενο τερματισμό της χωρίς απώλεια σημαντικών δεδομένων και στοιχείων των χρηστών: αν τυχόν το πρόγραμμα τερματιστεί, όπως για παράδειγμα από διακοπή ρεύματος, η εφαρμογή θα πρέπει να επαναξεκινήσει χωρίς πρόβλημα και να μην έχουν χαθεί σημαντικά στοιχεία. Γι' αυτό άλλωστε τα περισσότερα στοιχεία που αφορούν το χρήστη αποθηκεύονται στην ίδια την κάρτα του.

5.1.3 Περιγραφή Σχεδίασης

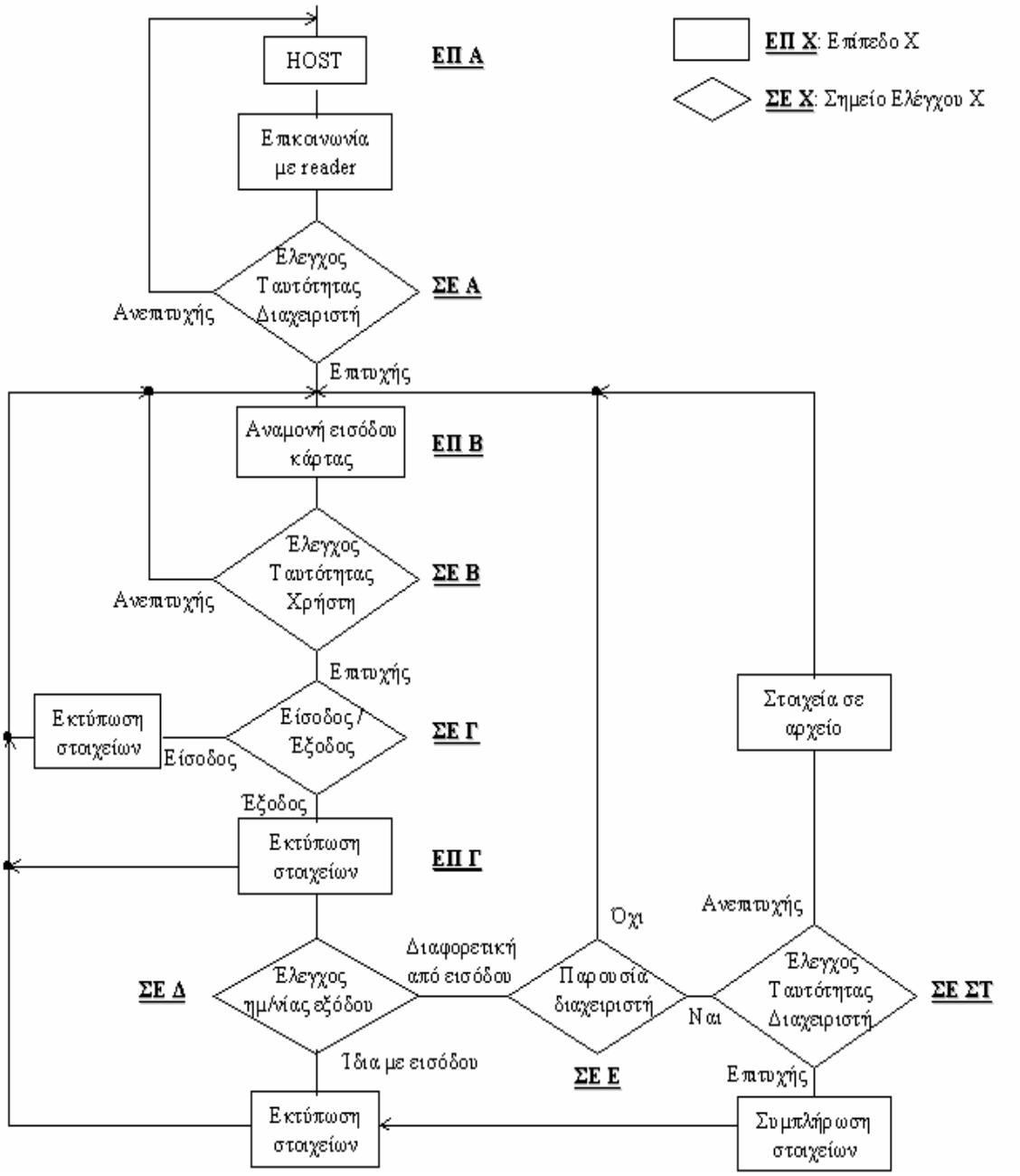
Όπως είδαμε, η εφαρμογή της μισθοδοσίας περιέχει πολλές παραμέτρους και πρέπει να υποστηρίζει ποικίλες λειτουργίες. Με σκοπό λοιπόν να γίνει εύκολη η κατανόηση της δομής της και των διαφόρων διεργασιών που εκτελούνται, καθώς και για να είναι πιο εύχρηστη για τον σχεδιαστή και τον τελικό διαχειριστή της, χωρίστηκε σε δύο κύρια μέρη.

Το πρώτο μέρος είναι το πρόγραμμα που χειρίζεται την είσοδο και την έξοδο των εργαζομένων στο χώρο εργασίας, μετράει το διάστημα που παραμένουν σε αυτόν και εκτυπώνει πληροφορίες που αφορούν τη διαδικασία εισόδου, εξόδου και τα χαρακτηριστικά της εργασίας του εργαζόμενου. Λόγω του ότι η προγραμματιστική δυσκολία αυτού του κομματιού παρουσιάζεται στον υπολογισμό των ωρών εργασίας και το διαχωρισμό τους σε κανονικές και υπερωριακές, το πρόγραμμα αυτό ονομάστηκε "Υπερωρίες".

Το δεύτερο μέρος είναι ένα πρόγραμμα που εκτελεί τους υπολογισμούς του συνολικού μισθού του εργαζόμενου στο τέλος του μήνα όταν αυτός πάει να εισπράξει το μισθό που αντιστοιχεί στο σύνολο της εργασίας τους για αυτό το χρονικό διάστημα. Το κομμάτι αυτό προσφέρει επίσης διαχειριστικές λειτουργίες όπως η χειροκίνητη εισαγωγή στοιχείων εργασίας του εργαζόμενου από τον διαχειριστή (κυρίως για αντιμετώπιση σφαλμάτων που έχουν παρουσιασθεί στο πρόγραμμα "Υπερωρίες"), βοηθητικές λειτουργίες όπως η περίπτωση της προκαταβολής του μισθού και τέλος λειτουργίες σύνδεσης με την εφαρμογή του ηλεκτρονικού πορτοφολιού που θα αναλύσουμε αργότερα. Το μέρος αυτό της εφαρμογής ονομάστηκε "Εξαργύρωση" λόγω του ότι το πρόγραμμα αυτό αφορά κυρίως την περίπτωση που ο εργαζόμενος επιθυμεί την εξαργύρωση των ωρών εργασίας του κατά τη διάρκεια του προηγούμενου μήνα. Τα δύο μέρη που αναφέρουμε αναλύονται στις επόμενες ενότητες.

5.1.3.1 Πρόγραμμα "Υπερωρίες"

Όπως ειπώθηκε σε προηγούμενη ενότητα, το πρόγραμμα αυτό τρέχει σε ένα υπολογιστή που έχει το ρόλο του host συστήματος και στον οποίο είναι συνδεδεμένος ο reader GCR410. Το πρόγραμμα ξεκινάει μόνο με πρωτοβουλία του διαχειριστή και αφού ο ίδιος εισάγει την κάρτα του στο reader.
Η δομή του προγράμματος φαίνεται σχηματικά στο ακόλουθο διάγραμμα, το οποίο θα εξηγήσουμε αναλυτικότερα ακολούθως.



Σχήμα 5.1.3.1 - Δομή Προγράμματος «Υπερωρίες»

- Επίπεδο Α:
  Όταν ο διαχειριστής ανοίγει το πρόγραμμα, αυτό ανοίγει κανάλι επικοινωνίας με τον reader και περιμένει την είσοδο μίας κάρτας, για να μπορέσει έπειτα να εξακριβώσει την ταυτότητά του. Με την είσοδο μίας κάρτας, ανοίγει και σύνοδος μεταξύ συστήματος και κάρτας (session). Επίσης όταν ανοίγει το πρόγραμμα, ανοίγει για γράψιμο και ένα αρχείο text το οποίο χρησιμοποιείται ως log αρχείο για την καταγραφή σημαντικών στοιχείων και για την αναφορά λαθών. Η χρησιμότητα αυτού του αρχείου είναι μεγάλη, ειδικά στις περιπτώσεις σφαλμάτων, όπως θα δούμε στο επόμενο κεφάλαιο.
- Σημείο Ελέγχου Α:
  Η εξακρίβωση της ταυτότητας του διαχειριστή γίνεται με τη σύγκριση ενός μυστικού κλειδιού που είναι αποθηκευμένο στην κάρτα, με τρόπο που θα εξηγηθεί παρακάτω. Το κλειδί είναι κρυπτογραφημένο και υπάρχει και σε κάθε κάρτα εργαζόμενου για να μπορεί να είναι ασφαλής οποιαδήποτε μετατροπή δεδομένων της κάρτας που σχετίζονται με αυτό το πρόγραμμα. Το Secret Key αυτό είναι αφιερωμένο στον διαχειριστή του προγράμματος "Υπερωρίες" και χρησιμοποιείται για να πιστοποιήσει την ταυτότητα του διαχειριστή και για όλες τις συναλλαγές μεταξύ reader και καρτών (για αυτό το πρόγραμμα) οι οποίες χρειάζονται Secure Messaging.
  Δεν είναι απαραίτητο να εισάγει ο διαχειριστής τη δική του κάρτα για να ξεκινήσει η εφαρμογή αλλά μπορεί να γίνει και με την είσοδο της κάρτας ενός εργαζόμενου. Όταν δηλαδή ξεκινάει το πρόγραμμα και ανοίγει ο δίαυλος επικοινωνίας με τον reader και την κάρτα, ζητάει ένα κωδικό διαχειριστή. Ο διαχειριστής πρέπει να εισάγει τον κωδικό του ο οποίος με κατάλληλη διαδικασία θα συγκριθεί με ένα κλειδί αποθηκευμένο σε όλες τις κάρτες για την πιστοποίηση της ταυτότητάς του.
  Για λόγους λειτουργικότητας βέβαια, είναι προτιμότερο το ξεκίνημα της εφαρμογής να γίνεται από τον διαχειριστή με τη δική του κάρτα όταν αυτός εισέρχεται στο χώρο εργασίας, για να αφήνει έτσι το πρόγραμμα να τρέχει και να είναι έτοιμο όταν ο πρώτος εργαζόμενος μπει στο χώρο εργασίας.
  Αν ο διαχειριστής δεν παρουσιάσει τον σωστό κωδικό το πρόγραμμα κλείνει το δίαυλο επικοινωνίας με τον reader και την κάρτα αφού ενημερώσει το log αρχείο και τερματίζει, ενώ στην αντίθετη περίπτωση συνεχίζει και περνάει στο επίπεδο Β (όπως φαίνεται στο Σχήμα 5.1.3.1), που ελέγχει αν στην υποδοχή του reader υπάρχει κάρτα.
- Επίπεδο Β:
  Αυτό το επίπεδο αναμονής κάρτας είναι το σημείο στο οποίο επιστρέφει το πρόγραμμα μετά από κάθε είσοδο ή έξοδο εργαζόμενου για να περιμένει επόμενη εισαγωγή κάρτας στον reader, είναι δηλαδή το σημείο αρχής του βρόχου που αποτελεί το κύριο σώμα του προγράμματος. Αν ο διαχειριστής δεν έχει αφαιρέσει την κάρτα του από τον reader το πρόγραμμα περνάει στο επόμενο επίπεδο, όπως και στην περίπτωση οποιουσδήποτε εργαζόμενου που εισέρχεται στο χώρο εργασίας του. Το κομμάτι που ακολουθεί μετά το επίπεδο Β είναι και η βάση του προγράμματος και θα αναλυθεί αμέσως παρακάτω. Θεωρούμε λοιπόν ότι ο διαχειριστής έχει παρουσιάσει σωστά τον κωδικό του, η εφαρμογή έχει ξεκινήσει και βρίσκεται στο στάδιο όπου αναμένει μία κάρτα στην είσοδο του reader.
- Σημείο Ελέγχου Β:
  Μόλις το πρόγραμμα εντοπίσει μία κάρτα στην υποδοχή του reader ανοίγει μία σύνοδο μαζί της (session), καλωσορίζει τον κάτοχό της και ζητάει την παρουσίαση του κωδικού του. Ο κωδικός που εισάγει ο κάτοχος συγκρίνεται με τον μοναδικό κωδικό που του αντιστοιχεί και που είναι αποθηκευμένος στην κάρτα του.
  Αν η προσπάθεια παρουσίασης του κωδικού αυτού είναι ανεπιτυχής και έχει ξεπεραστεί το όριο προσπαθειών για τον κάτοχο, η κάρτα κλειδώνει και απορρίπτεται και το πρόγραμμα κλείνει τη σύνοδο με την κάρτα, γράφει στο log αρχείο ότι ο κάρτα αυτή κλειδώθηκε και επιστρέφει στο επίπεδο Α, περιμένοντας εκ νέου την εισαγωγή μίας κάρτας. Αν από την άλλη η επαλήθευση του κωδικού είναι επιτυχής, το

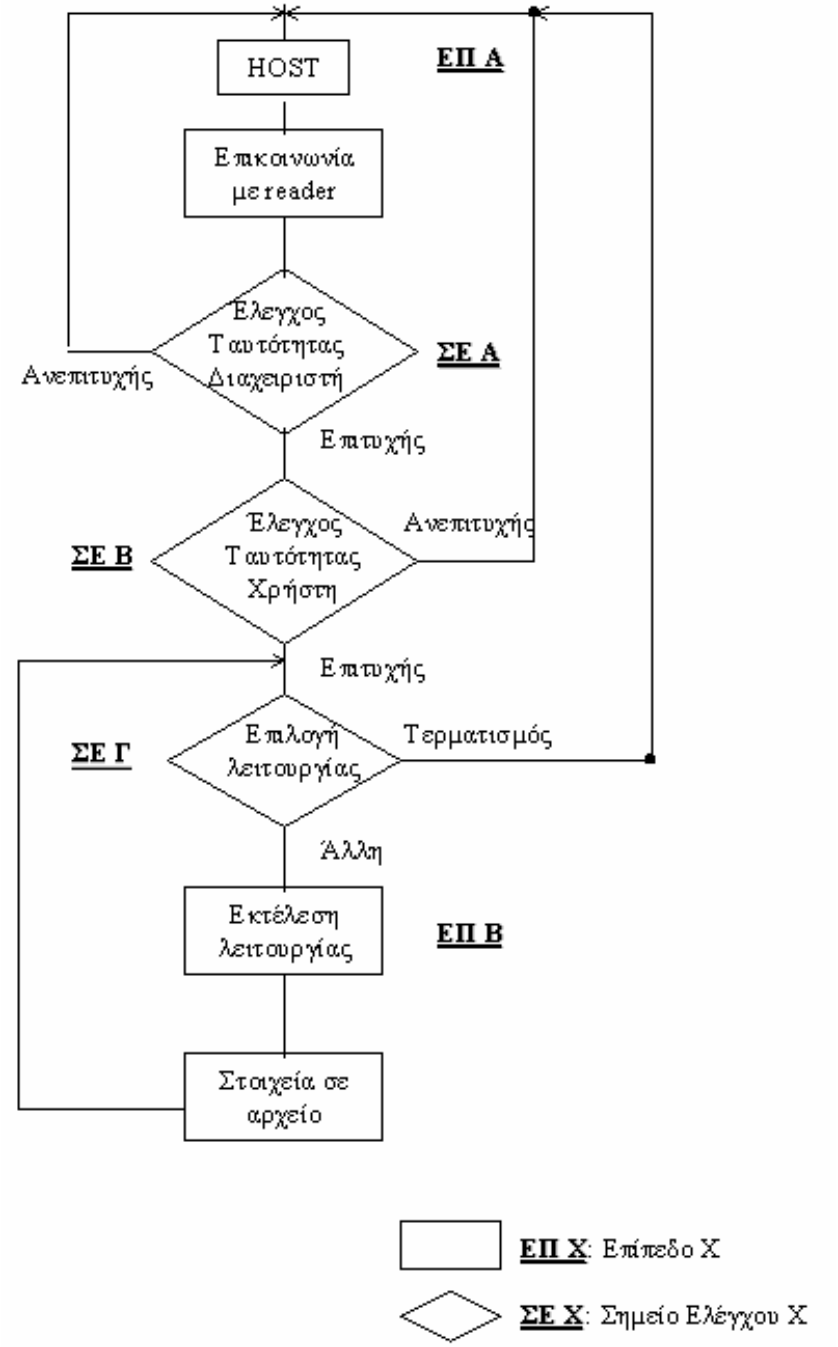
πρόγραμμα προχωράει στο σημείο ελέγχου Γ, όπου ελέγχει αν ο κάτοχός της εισέρχεται ή εξέρχεται από τον χώρο εργασίας του.

- Σημείο Ελέγχου Γ:
  Στην περίπτωση που ο εργαζόμενος εισέρχεται στον χώρο εργασίας, το πρόγραμμα ενημερώνει την κάρτα του για την ημερομηνία και την ώρα εισόδου του καθώς και για το γεγονός ότι κάνει είσοδο στο χώρο, τυπώνει αυτά τα στοιχεία στην οθόνη και το log αρχείο και προτρέπει τον εργαζόμενο να αφαιρέσει την κάρτα από τον reader. Μόλις η κάρτα αφαιρεθεί από την υποδοχή του reader το πρόγραμμα επιστρέφει στο επίπεδο Β.
- Επίπεδο Γ:
  Αν στο σημείο ελέγχου Γ διαπιστωθεί πως πρόκειται για έξοδο του χρήστη από τον χώρο εργασίας του τότε το πρόγραμμα περνάει στο επίπεδο Γ, όπου τυπώνει στην οθόνη και στο log αρχείο την ημερομηνία και την ώρα εξόδου του χρήστη και έπειτα περνάει στο σημείο ελέγχου Δ.
- Σημείο Ελέγχου Δ:
  Σε αυτό το σημείο ελέγχεται αν η ημερομηνία εξόδου ταυτίζεται με την ημερομηνία της αμέσως προηγούμενης εισόδου του εργαζόμενου. Αυτό γίνεται με σύγκριση της ημερομηνίας του συστήματος εκείνη την στιγμή με αυτή που είναι αποθηκευμένη στην κάρτα για την τελευταία είσοδο του χρήστη.
- Επίπεδο Γ:
  Όταν οι ημερομηνίες αυτές ταυτίζονται σημαίνει πως ο εργαζόμενος μπήκε και βγήκε την ίδια μέρα από το χώρο εργασίας του. Στην περίπτωση αυτή υπολογίζεται από το πρόγραμμα το χρονικό διάστημα που παρέμεινε ο εργαζόμενος στο χώρο εργασίας από την τελευταία είσοδό του σε αυτόν, υπολογίζει μέσα στην ημέρα πόσες ώρες κανονικής εργασίας και πόσες ώρες υπερωριακής εργασίας έχει κάνει ο εργαζόμενος, αποθηκεύει αυτά τα στοιχεία με κατάλληλο τρόπο στην κάρτα και τα τυπώνει στο log αρχείο και στην οθόνη. Τέλος το πρόγραμμα προτρέπει τον κάτοχο να αφαιρέσει την κάρτα του από το reader και μετά από αυτό επιστρέφει στο επίπεδο Β.
- Σημείο Ελέγχου Ε:
  Αν στο σημείο ελέγχου Δ οι ημερομηνίες τελευταίας εισόδου και εξόδου του εργαζόμενου διαπιστωθεί πως είναι διαφορετικές σημαίνει ότι ο εργαζόμενος δε χτύπησε την κάρτα του στην τελευταία έξοδό του από το χώρο εργασίας ή ότι άλλαξε η μέρα χωρίς να έχει φύγει ακόμα από το χώρο εργασίας. Τότε το πρόγραμμα ζητάει την παρουσία του διαχειριστή για να συμπληρωθούν με ασφάλεια αυτές οι πληροφορίες. Αν ο διαχειριστής δεν είναι παρών τη στιγμή εκείνη, το πρόγραμμα εγγράφει στην κάρτα την πληροφορία ότι έγινε έξοδος από τον χρήστη χωρίς όμως να συμπληρωθούν οι ώρες εργασίας του στην κάρτα και επιστρέφει στο επίπεδο Β.
- Σημείο Ελέγχου ΣΤ:
  Στην περίπτωση που ο διαχειριστής είναι παρών, το πρόγραμμα προχωράει στο σημείο ελέγχου ΣΤ, όπου ζητείται η παρουσίαση ενός δεύτερου κωδικού από τον διαχειριστή για να συγκριθεί με ένα αντίστοιχο κωδικό στην κάρτα.
  Αν η παρουσίαση του κωδικού αυτού είναι επιτυχής, το πρόγραμμα προτρέπει τον διαχειριστή να συμπληρώσει ο ίδιος τις ώρες εργασίας του εργαζόμενου για την προηγούμενη μέρα και υπολογίζει πόσες ώρες κανονικής εργασίας και πόσες ώρες υπερωριακής εργασίας του αντιστοιχούν. Έπειτα αποθηκεύει αυτά τα στοιχεία με κατάλληλο τρόπο στην κάρτα και τα τυπώνει στο log αρχείο και στην οθόνη. Τέλος το πρόγραμμα προτρέπει τον κάτοχο να αφαιρέσει την κάρτα του από το reader και επιστρέφει στο επίπεδο Β.
  Αν πάλι ο διαχειριστής δεν παρουσιάσει μετά από τις επιτρεπόμενες προσπάθειες τον σωστό κωδικό, η κάρτα κλειδώνει, όσα στοιχεία αφορούν αυτήν την έξοδο του χρήστη (πόσες ώρες έχουν περάσει από την τελευταία του είσοδο και πώς χωρίζονται σε κανονική εργασία και υπερωρίες) αποθηκεύονται στο αρχείο log, το γεγονός της εξόδου του χρήστη εγγράφεται στην κάρτα και το πρόγραμμα επιστρέφει στο επίπεδο Β αφού η κάρτα αφαιρεθεί από τον reader.

Από ότι έχουμε δει μέχρι τώρα, κάθε κάρτα στην εφαρμογή αυτή περιέχει ένα μυστικό κωδικό του κατόχου, ένα μυστικό κλειδί του διαχειριστή (για το ξεκίνημα της εφαρμογής και το Secure Messaging) και ένα μυστικό κωδικό του διαχειριστή για να μπορεί να συμπληρώσει στο σημείο ελέγχου ΣΤ χειροκίνητα τις ώρες εργασίας ενός εργαζόμενου. Ο δεύτερος αυτός κωδικός του διαχειριστή υφίσταται για να παρέχει αυξημένη ασφάλεια. Τα στοιχεία αυτά της ασφάλειας θα συζητηθούν αναλυτικότερα στο κεφάλαιο που αφορά την υλοποίηση των εφαρμογών.

5.1.3.2 Πρόγραμμα "Εξαργύρωση"

Η δομή του προγράμματος "Εξαργύρωση" φαίνεται στο ακόλουθο σχήμα, το οποίο θα αναλύσουμε ακολούθως.



Σχήμα 5.1.3.2 - Δομή Προγράμματος «Εξαργύρωση»

- Επίπεδο Α:
  Όπως και στο πρόγραμμα "Υπερωρίες", ο διαχειριστής ανοίγει το πρόγραμμα το οποίο με τη σειρά του ανοίγει κανάλι επικοινωνίας με τον reader και περιμένει την είσοδο μίας κάρτας, για να μπορέσει στη συνέχεια να εξακριβώσει την ταυτότητα του διαχειριστή. Με την είσοδο μίας κάρτας, ανοίγει μία σύνοδος (session) μεταξύ συστήματος και κάρτας.
  Με το που ανοίγει το πρόγραμμα ανοίγουν αυτή τη φορά δύο αρχεία text για γράψιμο. Το ένα χρησιμοποιείται ως log αρχείο για την καταγραφή σημαντικών στοιχείων και για την αναφορά λαθών, ενώ το δεύτερο χρησιμοποιείται ως μία απόδειξη για τον εργαζόμενο των συναλλαγών που πραγματοποιούνται. Όταν δηλαδή για παράδειγμα υπολογίζεται ο τελικός του μισθός για κάποιο μήνα, η ανάλυσή του εγγράφεται σε αυτό το αρχείο και εκτυπώνεται, για να μπορεί ο εργαζόμενος στο μέλλον να έχει αυτά τα στοιχεία αν τα χρειαστεί.
- Σημείο Ελέγχου Α:
  Μόλις το πρόγραμμα ανοίξει δίαυλο επικοινωνίας με τον reader και την κάρτα, ζητάει να ελέγξει την ταυτότητα του διαχειριστή. Επειδή το πρόγραμμα αυτό δεν χρειάζεται να τρέχει καθημερινά αλλά μόνο μετά από αίτηση του εργαζόμενου για να εκτελεστούν κάποιες λειτουργίες, έχει διαμορφωθεί έτσι ώστε να τρέχει για κάθε χρήστη και μετά να τερματίζει. Έτσι όταν ο εργαζόμενος προσέρχεται και επιθυμεί τη χρήση του προγράμματος, ο διαχειριστής ανοίγει το πρόγραμμα και πιστοποιεί την ταυτότητά του χρησιμοποιώντας την κάρτα του εργαζόμενου. Έπειτα εκτελεί τις λειτουργίες που επιθυμεί ο εργαζόμενος και τερματίζει το πρόγραμμα.
  Αν τώρα ο διαχειριστής δεν παρουσιάσει τον σωστό κωδικό το πρόγραμμα κλείνει το δίαυλο επικοινωνίας με τον reader και την κάρτα αφού ενημερώσει το log αρχείο και τερματίζει, ενώ στην αντίθετη περίπτωση συνεχίζει και περνάει στο σημείο ελέγχου Β.
- Σημείο Ελέγχου Β:
  Στο σημείο αυτό το πρόγραμμα καλωσορίζει τον κάτοχο της κάρτας και ζητάει την παρουσίαση του κωδικού του. Αν η προσπάθεια παρουσίασης του κωδικού αυτού είναι ανεπιτυχής και έχει ξεπεραστεί το όριο προσπαθειών για τον κάτοχο, η κάρτα κλειδώνει, το πρόγραμμα γράφει τα απαραίτητα στοιχεία στο log αρχείο, κλείνει τη σύνοδο με την κάρτα και το κανάλι με τον reader και τερματίζει. Αν πάλι η επαλήθευση του κωδικού του κατόχου είναι επιτυχής, το πρόγραμμα προχωράει στο σημείο ελέγχου Γ
- Σημείο Ελέγχου Γ:
  Σε αυτό το σημείο ο διαχειριστής τίθεται να επιλέξει ανάμεσα σε 11 λειτουργίες ποια θέλει να εκτελέσει. Αν η λειτουργία που διαλέξει είναι ο τερματισμός του προγράμματος, αυτό κλείνει τη σύνοδο με την κάρτα και το κανάλι με τον reader και τερματίζει. Αν επιλέξει οποιαδήποτε άλλη από τις 10 λειτουργίες που υπολείπονται, το πρόγραμμα περνάει στο επίπεδο Β, της εκτέλεσης της λειτουργίας.
- Επίπεδο Β:
  Στο επίπεδο Β, το πρόγραμμα εκτελεί τη λειτουργία που επέλεξε ο διαχειριστής σύμφωνα με παραμέτρους που τυχόν χρειάζονται και εισάγει ο διαχειριστής. Οι λειτουργίες αυτές σχετίζονται όχι μόνο με την εφαρμογή της μισθοδοσίας αλλά και με τη διαχείριση του ηλεκτρονικού πορτοφολιού και θα τις αναλύσουμε στην ενότητα 5.1.3.4. Τα στοιχεία – αποτελέσματα της λειτουργίας που εκτελεί το πρόγραμμα τα εγγράφει στο αρχείο log και όπου χρειάζεται και στο αρχείο απόδειξης προς χρήση του κατόχου της κάρτας. Αν στην εκτέλεση κάποιας λειτουργίας προκύψει σφάλμα αυτό αναφέρεται στο αρχείο log. Σε όλες τις περιπτώσεις, μετά την εκτέλεση της λειτουργίας το πρόγραμμα επιστρέφει στο σημείο ελέγχου Γ, όπου ο διαχειριστής μπορεί να διαλέξει την επόμενη λειτουργία που επιθυμεί να εκτελέσει.

Στο πρόγραμμα αυτό χρησιμοποιούνται δύο από τους κωδικούς της κάρτας που αναφέραμε παραπάνω, ο κωδικός του κατόχου και το μυστικό κλειδί του διαχειριστή.

5.1.3.3 Πολιτική Υπερωριών

Στην εφαρμογή της μισθοδοσίας ακολουθήθηκε όπως έχει ειπωθεί μία συγκεκριμένη πολιτική για τις υπερωρίες που υπολογίζονται στον μισθό ενός εργαζόμενου. Το στοιχείο που χρησιμοποιεί η εφαρμογή για τους υπολογισμούς της μισθοδοσίας είναι το ωρομίσθιο του εργαζόμενου, το οποίο και αποθηκεύεται ως πληροφορία στην κάρτα του.
Θεωρείται ότι η κανονική εργασία ενός εργαζόμενου συνίσταται σε 8 ώρες ημερησίως για τις εργάσιμες ημέρες. Αν σε μία καθημερινή ο εργαζόμενος ξεπεράσει τις ώρες αυτές, ο παραπάνω χρόνος εργασίας αντιστοιχεί σε υπερωριακή εργασία.
Τα πρώτα 15 λεπτά υπερωριακής εργασίας μέσα σε καθημερινή αμείβονται με προσαύξηση 30% επί του ωρομισθίου του εργαζόμενου. Έτσι αν το ωρομίσθιό του είναι 10 €/ώρα, για τα 15 επιπλέον λεπτά εργασίας του θα πληρωθεί:

$$15_{\text{ΛΕΠΤΑ}} \bullet 1{,}3 \bullet \left(\frac{10}{60}\right)_{€/\text{ΛΕΠΤΟ}} = 3{,}25€ .$$

Ο υπόλοιπος χρόνος υπερωριακής εργασίας του εργαζόμενου, πλέον των 15 λεπτών, αμείβεται με προσαύξηση 50% επί του ωρομισθίου του εργαζόμενου. Σύμφωνα με αυτό, αν ένας εργαζόμενος κάνει υπερωριακή εργασία 1 ώρας και 30 λεπτών, στα πρώτα 15 λεπτά θα αντιστοιχεί το ποσοστό 30% ενώ στην υπόλοιπη 1 ώρα και 15 λεπτά θα αντιστοιχεί το ποσοστό 50%. Για αυτή την εργασία λοιπόν, ένας εργαζόμενος με ωρομίσθιο 10 €/ώρα θα πληρωθεί:

$$15_{\text{ΛΕΠΤΑ}} \bullet 1{,}3 \bullet \left(\frac{10}{60}\right)_{€/\text{ΛΕΠΤΟ}} + 1_{\text{ΩΡΑ}} \bullet 1{,}5 \bullet 10_{€/\text{ΩΡΑ}} + 15_{\text{ΛΕΠΤΑ}} \bullet 1{,}5 \bullet \left(\frac{10}{60}\right)_{€/\text{ΛΕΠΤΟ}} = 22€.$$

Αν ένας εργαζόμενος εργαστεί Σάββατο, η εργασία θεωρείται υπερωριακή και αμείβεται με προσαύξηση 50% επί του ωρομισθίου. Έτσι ένας εργαζόμενος με ωρομίσθιο 10 €/ώρα, για εργασία 1 ώρας και 30 λεπτών το Σάββατο, θα αμειφθεί με:

$$1_{\text{ΩΡΑ}} \bullet 1{,}5 \bullet 10_{€/\text{ΩΡΑ}} + 30_{\text{ΛΕΠΤΑ}} \bullet 1{,}5 \bullet \left(\frac{10}{60}\right)_{€/\text{ΛΕΠΤΟ}} = 22{,}5€.$$

Αν πάλι εργαστεί Κυριακή ή αργία, η εργασία θεωρείται υπερωριακή με προσαύξηση 50% επί του ωρομισθίου του εργαζόμενου και οι ώρες εργασίας του προσμετρώνται ως διπλάσιες. Δηλαδή αν ένας εργαζόμενος με ωρομίσθιο 10 €/ώρα, δουλέψει μία Κυριακή για 2 ώρες και 23 λεπτά, θα μετρηθεί ότι έχει δουλέψει 4 ώρες και 46 λεπτά και θα αμειφθεί για αυτές με:

$$4_{\text{ΩΡΕΣ}} \bullet 1{,}5 \bullet 10_{€/\text{ΩΡΑ}} + 46_{\text{ΛΕΠΤΑ}} \bullet 1{,}5 \bullet \left(\frac{10}{60}\right)_{€/\text{ΛΕΠΤΟ}} = 71{,}5€.$$

Οι ημέρες αδείας υπολογίζονται ως ημέρες καθημερινής κανονικής εργασίας 8 ωρών. Έτσι αν ένας εργαζόμενος έχει ωρομίσθιο 10 €/ώρα, για μία ημέρα αδείας θα πληρωθεί: $8_{\text{ΩΡΕΣ}} \bullet 10_{€/\text{ΩΡΑ}} = 80€.$

Στην εφαρμογή αυτή της μισθοδοσίας, ο εργαζόμενος έχει ένα ανώτατο όριο υπερωριών που μπορεί να κάνει μέσα σε ένα μήνα. Το όριο αυτό μπορεί να διαφέρει από εργαζόμενο σε εργαζόμενο και είναι μία πληροφορία που αποθηκεύεται στην κάρτα του και χρησιμοποιείται κατά τον υπολογισμό του τελικού μισθού του στο πρόγραμμα της "Εξαργύρωσης".
Αν ξεπεράσει το όριο των υπερωριών του για ένα μήνα είναι στη κρίση του διαχειριστή αν θα προσμετρήσει αυτές τις ώρες ως υπερωρίες ή όχι και αν θα τις "αφαιρέσει" από τον επόμενο μήνα. Αυτό γίνεται γιατί υπάρχει η πιθανότητα ένας εργαζόμενος να μην προσέλθει στο τέλος του μήνα για να πληρωθεί αλλά κάποια μέρα μέσα στον επόμενο μήνα, όπου θα έχουν συμπληρωθεί και ώρες εργασίας του νέου μήνα.

Συγκεκριμένα, στην περίπτωση που ένας εργαζόμενος ξεπεράσει το όριο αυτό των υπερωριών, ο διαχειριστής έχει 3 επιλογές.
Η πρώτη είναι να μην υπολογίσει τις υπερωρίες καθόλου και ο εργαζόμενος να μην πληρωθεί ποτέ για αυτές, με τη λογική ότι το όριο το γνωρίζει ο εργαζόμενος και πρέπει να μην το υπερβαίνει.
Η δεύτερη είναι να τις μετρήσει ως πληρωτέες μέσα στον ίδιο μήνα, δεχόμενος κάποιες ειδικές ανάγκες που προέκυψαν στο χώρο εργασίας.
Η τελευταία και πιο ενδιαφέρουσα επιλογή είναι να πληρωθούν οι υπερωρίες αυτές στον εργαζόμενο, όχι όμως ως υπερωρίες αυτού του μήνα αλλά ως προκαταβολή κάποιου επόμενου μήνα. Δηλαδή, ο εργαζόμενος θα εισπράξει μεν τον ίδιο μήνα το ποσό που τους αντιστοιχεί, θα το χρωστάει όμως τον επόμενο μήνα και θα αφαιρεθεί από το ποσό που θα αντιστοιχεί στις υπερωρίες του μήνα εκείνου. Αν τον επόμενο μήνα, το ποσό που αντιστοιχεί στις υπερωρίες που θα έχει κάνει δεν αρκεί για να καλύψει την προκαταβολή υπερωριακής εργασίας που έχει χρησιμοποιήσει, η διαφορά θα μεταφερθεί στον επόμενο μήνα κ.ο.κ.
Η προκαταβολή υπερωριακής εργασίας διαφέρει από την προκαταβολή κανονικής εργασίας την οποία θα δούμε στην επόμενη ενότητα.

5.1.3.4 Λειτουργίες Προγράμματος "Εξαργύρωση"

Υπάρχουν 10 λειτουργίες πέραν της λειτουργίας του τερματισμού τις οποίες παρέχει το πρόγραμμα αυτό. Θα τις εξετάσουμε ακολούθως αναλυτικά:

- Υπολογισμός μισθού εργαζόμενου:
  Με αυτή την επιλογή, το πρόγραμμα υπολογίζει το μισθό που θα πληρωθεί ο κάτοχος της κάρτας για τις ώρες εργασίας που είναι καταγεγραμμένες στην κάρτα του. Συγκεκριμένα το πρόγραμμα διαβάζει από την κάρτα με κατάλληλο τρόπο πόσες ώρες κανονικής και υπερωριακής εργασίας έχει δουλέψει από την προηγούμενη φορά που πληρώθηκε, το όριο των υπερωριών του για ένα μήνα, το όριο των ημερών αδείας του για ένα χρόνο καθώς και πληροφορίες που αφορούν τυχούσα προκαταβολή που μπορεί να έχει λάβει και στοιχεία για τη χρήση παραπάνω ωρών υπερωριακής εργασίας του προηγούμενου μήνα. Αν δηλαδή ο εργαζόμενος την προηγούμενη φορά που πληρώθηκε, είχε υπερβεί το όριο των υπερωριών του και είχε πάρει προκαταβολή υπερωριακής εργασίας, το ποσό αυτό θα φαίνεται στον υπολογισμό του μισθού του.
  Αρχικά εκτυπώνει στην οθόνη και το αρχείο log τις ώρες κανονικής και υπερωριακής εργασίας που έχει συμπληρώσει ο εργαζόμενος από την τελευταία φορά που πληρώθηκε, τον πληροφορεί για το αν έχει υπερβεί το όριο των υπερωριών και εμφανίζει τις παραπάνω ώρες, εμφανίζει τα ποσά κανονικής και υπερωριακής προκαταβολής που έχει τυχόν πάρει ο εργαζόμενος, τον προτρέπει να εισάγει τις ημέρες αδείας που έχει χρησιμοποιήσει για το διάστημα υπολογισμού, υπολογίζει και εμφανίζει τα χρήματα που θα πληρωθεί για αυτές τις μέρες και τέλος εμφανίζει το συνολικό μισθό που αντιστοιχεί στον εργαζόμενο με βάση όλα τα δεδομένα που αναφέραμε.
  Όλα αυτά τα στοιχεία είναι πληροφοριακά και η συγκεκριμένη λειτουργία χρησιμεύει για να πληροφορήσει τον εργαζόμενο για το μισθό που θα έπαιρνε αν εξαργύρωνε την εργασία του εκείνη τη στιγμή. Η λειτουργία αυτή εκτελείται συνήθως νωρίτερα από τη μέρα πληρωμής του μισθού, όταν ο εργαζόμενος θέλει για παράδειγμα να πληροφορηθεί πόσες υπερωρίες έχει κάνει.
- Υπολογισμός και εξαργύρωση μισθού εργαζόμενου:
  Η λειτουργία αυτή εκτελείται όταν ο εργαζόμενος επιθυμεί να πληρωθεί για την εργασία που έχει εκτελέσει μέσα σε συγκεκριμένο χρονικό διάστημα (συνήθως μέσα σε ένα μήνα). Όπως και στην προηγούμενη λειτουργία το πρόγραμμα διαβάζει από την κάρτα με κατάλληλο τρόπο πόσες ώρες κανονικής και υπερωριακής εργασίας έχει δουλέψει από την προηγούμενη φορά που πληρώθηκε, το όριο των υπερωριών

του για ένα μήνα, τον αριθμό των ημερών αδείας που μπορεί να χρησιμοποιήσει, καθώς και τα ποσά κανονικής και υπερωριακής προκαταβολής που έχει πάρει.
Όπως ειπώθηκε νωρίτερα, υπερωριακή προκαταβολή θεωρείται το ποσό που έχει πληρωθεί ένας εργαζόμενος για ώρες υπερωριακής εργασίας που υπερέβαιναν το επιτρεπόμενο όριο κάποιο προηγούμενο μήνα. Η προκαταβολή αυτή έχει σίγουρα πληρωθεί στον εργαζόμενο σε κάποια από τις προηγούμενες εξαργυρώσεις του.
Κανονική προκαταβολή θεωρείται η προκαταβολή χρημάτων από το ποσό που αντιστοιχεί στις ώρες κανονικής εργασίας του εργαζόμενου μέσα στο χρονικό διάστημα υπολογισμού του μισθού, σίγουρα δηλαδή μετά από την τελευταία φορά εξαργύρωσης. Αν όμως ένας εργαζόμενος έχει πάρει υπερωριακή προκαταβολή κάποιο προηγούμενο χρονικό διάστημα, δεν δικαιούται να πάρει κανονική προκαταβολή μέχρι να "ξεπληρώσει" με την υπερωριακή εργασία του το ποσό που "χρωστάει".
Αρχικά η λειτουργία αυτή εκτυπώνει στην οθόνη, το αρχείο log και το αρχείο receipt (απόδειξης του κατόχου) τις ώρες κανονικής και υπερωριακής εργασίας που έχει συμπληρώσει ο εργαζόμενος από την τελευταία φορά που πληρώθηκε. Εμφανίζει στη συνέχεια στην οθόνη και εγγράφει στα αρχεία τα ποσά κανονικής και υπερωριακής προκαταβολής του εργαζόμενου (αν έχει πάρει τέτοιες προκαταβολές).
Αν ο εργαζόμενος έχει υπερβεί το όριο των υπερωριών του, εμφανίζει τις ώρες που είναι πάνω από το όριο και προτρέπει τον διαχειριστή να λάβει μία απόφαση για το αν ο εργαζόμενος θα πληρωθεί τα χρήματα που αντιστοιχούν στις παραπάνω ώρες και αν η πληρωμή τους θα θεωρηθεί ως υπερωριακή προκαταβολή.
Μετά και από αυτή την πληροφορία, το πρόγραμμα ζητάει να συμπληρωθούν οι ημέρες αδείας που έχει χρησιμοποιήσει ο εργαζόμενος στο χρονικό διάστημα υπολογισμού του μισθού του και ελέγχει αν ο αριθμός των ημερών που δηλώνει ο χρήστης υπερβαίνει τον αριθμό των ημερών που δικαιούται ο εργαζόμενος. Αν ναι, τότε τον πληροφορεί για αυτή την υπέρβαση και τον καλεί να συμπληρώσει ένα αποδεκτό αριθμό ημερών. Όταν αυτό γίνει, το πρόγραμμα ενημερώνει την κάρτα για το νέο αριθμό ημερών αδείας που δικαιούται ο εργαζόμενος.
Τέλος, το πρόγραμμα βασισμένο σε όλες τις πληροφορίες που αναφέραμε κάνει τους τελικούς υπολογισμούς για το συνολικό μισθό που θα πληρωθεί ο εργαζόμενος, τυπώνει το τελικό ποσό στην οθόνη και το εγγράφει και στα δύο αρχεία.
Μετά από αυτή τη λειτουργία ο εργαζόμενος πληρώνεται το ποσό που έχει υπολογιστεί και έπειτα πρέπει να εκτελεστεί οπωσδήποτε η λειτουργία αρχικοποίησης – μηδενισμού συγκεκριμένων αρχείων της κάρτας, για να είναι η κάρτα έτοιμη ξανά για χρήση στην εφαρμογή.

- <u>Αρχικοποίηση – Μηδενισμός αρχείων κάρτας</u>:
  Η λειτουργία αυτή εκτελείται πάντα μετά από τη λειτουργία υπολογισμού – εξαργύρωσης του μισθού του εργαζόμενου. Ο διαχειριστής απλά τρέχει τη διαδικασία η οποία αρχικοποιεί - μηδενίζει το περιεχόμενο κάποιων αρχείων στην κάρτα που αφορούν την εφαρμογή αυτή για να μπορεί να χρησιμοποιηθεί για το νέο μήνα. Κατά τη διάρκεια αυτής της λειτουργίας, αρχικοποιούνται κάποια αρχεία εγγραφών και ορισμένοι μετρητές της κάρτας που χρησιμοποιούνται για τον υπολογισμό των ωρών κανονικής και υπερωριακής εργασίας του εργαζόμενου. Λεπτομέρειες για τα αρχεία αυτά και τα περιεχόμενά τους θα δοθούν στο επόμενο κεφάλαιο που αναλύεται η υλοποίηση της εφαρμογής και η τελική σύσταση της κάρτας ενός χρήστη.
- <u>Ενημέρωση επιτρεπόμενων ημερών αδείας εργαζόμενου</u>:
  Όπως αναφέρθηκε παραπάνω, στην κάρτα υπάρχει ανά πάσα στιγμή η πληροφορία του αριθμού των ημερών αδείας που δικαιούται ο εργαζόμενος μέχρι το τέλος του χρόνου. Στο τέλος του χρόνου, στην τελευταία πληρωμή του εργαζόμενου, ο αριθμός αυτός πρέπει να ανανεωθεί για το νέο έτος. Αν υπάρχουν ημέρες αδείας οι οποίες δεν έχουν χρησιμοποιηθεί από τον εργαζόμενο, αυτές μπορούν να μεταφερθούν στο επόμενο έτος, ανάλογα με την κρίση του διαχειριστή και τις συμβάσεις εργασίας.
  Η λειτουργία αυτή ενημερώνει την κάρτα για το νέο συνολικό αριθμό ημερών αδείας.

- Κανονική προκαταβολή:
  Σύμφωνα με τις προδιαγραφές της εφαρμογής, ένας εργαζόμενος μπορεί να πάρει μία προκαταβολή από το μισθό του πριν την πλήρη εξαργύρωσή του. Η προκαταβολή που θα πάρει αφαιρείται από το ποσό που αντιστοιχεί στις ώρες κανονικής εργασίας που έχει συμπληρώσει από τη τελευταία του πληρωμή μέχρι εκείνη τη στιγμή. Έτσι ο εργαζόμενος μπορεί να πάρει προκαταβολή μικρότερη ή ίση με το ποσό αυτό. Έχει οριστεί το ποσό που μπορεί να πάρει να είναι μόνο ακέραιο.
  Στην περίπτωση που ο εργαζόμενος έχει πάρει υπερωριακή προκαταβολή σε προηγούμενο χρονικό διάστημα την οποία δεν έχει "ξεπληρώσει" ακόμα με τις υπερωρίες του, δεν επιτρέπεται να πάρει κανονική προκαταβολή.
  Έτσι όταν η λειτουργία καλείται να εκτελεστεί, ελέγχει αν υπάρχει διαθέσιμο ποσό κανονικής εργασίας ή αν υπάρχει οφειλή υπερωριακής προκαταβολής. Και στις δύο αυτές περιπτώσεις το πρόγραμμα απορρίπτει το αίτημα για κανονική προκαταβολή. Τα στοιχεία αυτά εγγράφονται και στο αρχείο log.
  Αν καμία από τις δύο παραπάνω συνθήκες δεν ικανοποιηθεί, το πρόγραμμα συνεχίζει και προτρέπει το διαχειριστή να εισάγει το ποσό της προκαταβολής που επιθυμεί ο εργαζόμενος. Το ποσό που εισάγει ελέγχεται πάλι με το ποσό που αντιστοιχεί στις ώρες κανονικής εργασίας του εργαζόμενου και μόνο όταν το πρώτο είναι μικρότερο ή ίσο με το δεύτερο προχωράει στην παραχώρηση της προκαταβολής, στην ενημέρωση των αρχείων της κάρτας για αυτή την πράξη και στην εγγραφή στο αρχείο receipt του ποσού προκαταβολής.
- Χειροκίνητη ενημέρωση μετρητών κάρτας:
  Η λειτουργία αυτή είναι καθαρά διαχειριστική και χρησιμεύει κυρίως στην αντιμετώπιση τυχόν σφαλμάτων που έχουν παρουσιαστεί στην εκτέλεση του προγράμματος "Υπερωρίες". Συγκεκριμένα, ο διαχειριστής συμπληρώνει χειροκίνητα το περιεχόμενο των μετρητών της κάρτας οι οποίοι χρησιμεύουν για την αποθήκευση των ωρών εργασίας, κανονικής και υπερωριακής. Τα ποσά που συμπληρώνουν (οι πόντοι κατά βάση που προσθέτουν σε κάθε μετρητή) προκύπτουν από τα στοιχεία που έχουν εγγραφεί στο αρχείο log. Έτσι αν σε κάποιο σημείο του προγράμματος "Υπερωρίες" δεν έχουν ολοκληρωθεί επιτυχώς κάποιες συναλλαγές, τα στοιχεία των συναλλαγών αυτών βρίσκονται στο αρχείο log το οποίο και χρησιμοποιεί ο διαχειριστής για να υπολογίσει τις αλλαγές που πρέπει να κάνει στα περιεχόμενα της κάρτας για να αποκαταστήσει τη ζημιά. Παραδείγματα της χρήσης αυτής της λειτουργίας θα δούμε σε επόμενο κεφάλαιο.
- Ανάγνωση περιεχομένου ηλεκτρονικού πορτοφολιού εργαζόμενου:
  Ο διαχειριστής έχει τη δυνατότητα να ενημερώσει τον εργαζόμενο για το ποσό που περιέχεται στο ηλεκτρονικό του πορτοφόλι. Η λειτουργία αυτή, όπως και όλες οι λειτουργίες που ακολουθούν, ανήκει στην επόμενη εφαρμογή και θα τη συζητήσουμε αναλυτικότερα στην επόμενη ενότητα.
- Προσθήκη χρημάτων στο ηλεκτρονικό πορτοφόλι του εργαζόμενου:
  Ο διαχειριστής μπορεί με την εκτέλεση αυτής της λειτουργίας να προσθέσει κάποιο ποσό στο e-purse του κατόχου της κάρτας. Ένας εργαζόμενος μπορεί κατά την πληρωμή του να επιθυμεί να προσθέσει μέρος ή ολόκληρο το μισθό του στο e–purse για παράδειγμα.
- Εξαργύρωση ολόκληρου του ηλεκτρονικού πορτοφολιού του εργαζόμενου:
  Η λειτουργία αυτή χρησιμεύει για την αφαίρεση από το ηλεκτρονικό πορτοφόλι όλου του ποσού του και την απόδοσή του στον κάτοχο της κάρτας. Χρησιμοποιείται μόνο στην περίπτωση που ο κάτοχος επιθυμεί την λήξη της κάρτας του ή κάτω από ειδικές συνθήκες.
- Ανάγνωση πόντων από πρόγραμμα εμπιστοσύνης:
  Η τελευταία αυτή λειτουργία δίνει στον διαχειριστή και τον εργαζόμενο τη δυνατότητα να πληροφορηθεί για τον αριθμό των πόντων που έχει κερδίσει κατά τη διάρκεια ενός προγράμματος εμπιστοσύνης.

5.2 Εφαρμογή Ηλεκ/νικού Πορτοφολιού – Προγ/ματος Εμπιστοσύνης

Η δεύτερη εφαρμογή που σχεδιάστηκε με βάση τη τεχνολογία των GemClub καρτών είναι ένας συνδυασμός ηλεκτρονικού πορτοφολιού και προγράμματος εμπιστοσύνης. Η ίδια κάρτα που χρησιμοποιήθηκε για την προηγούμενη εφαρμογή χρησιμοποιείται και σε αυτήν, με τις κατάλληλες βέβαια προσθήκες για τις λειτουργίες αυτής της εφαρμογής.
Το ηλεκτρονικό πορτοφόλι όπως έχει αναφερθεί, είναι ένας τρόπος αποθήκευσης χρημάτων σε ηλεκτρονική μορφή (σαν κάρτα προπληρωμένης αξίας). Μπορεί να γίνει και πίστωση και χρέωση στο πορτοφόλι, δηλαδή μπορούν και να προστεθούν ποσά σε αυτό και να αφαιρεθούν.
Το πρόγραμμα εμπιστοσύνης αφορά την ανταμοιβή με πόντους ενός κατόχου κάρτας για τις αγορές του σε συγκεκριμένα σημεία πώλησης. Στην εφαρμογή αυτή ο κάτοχος της κάρτας θα χρησιμοποιεί την κάρτα του ως ηλεκτρονικό πορτοφόλι για την αγορά κάποιων προϊόντων και ανάλογα με το ύψος των αγορών θα κερδίζει πόντους τους οποίους έπειτα θα εξαργυρώνει κερδίζοντας κάποιο προϊόν ή κάποια υπηρεσία, ανάλογα με τι προβλέπει ο πάροχος του προγράμματος εμπιστοσύνης.
Η εφαρμογή αυτή υπάγεται στα ίδια πλαίσια με την προηγούμενη, και έτσι θεωρήθηκε ότι λαμβάνει θέση σε ένα πανεπιστημιακό ίδρυμα και απευθύνεται στους εργαζομένους του. Με την ίδια κάρτα που χρησιμοποιεί για την μισθοδοσία του, ο εργαζόμενος μπορεί να κάνει αγορές μεγάλου ή μικρού ύψους σε διάφορα σημεία του πανεπιστημίου, όπως τα κυλικεία, τα φωτοτυπεία, τα βιβλιοπωλεία ή και συνεργαζόμενα καταστήματα με χρήσιμο εξοπλισμό.
Όλα αυτά τα σημεία θα αποτελούν τα λεγόμενα σημεία πώλησης ή εξυπηρέτησης.
Το πρόγραμμα εμπιστοσύνης θα θεωρήσουμε ότι εφαρμόζεται σε όλα αυτά τα σημεία πώλησης που βρίσκονται στο χώρο του πανεπιστημίου και τα οποία ανάλογα με τους πόντους που έχει μαζέψει ο πελάτης και το αντικείμενο εργασίας τους προσφέρουν αντίστοιχα δώρα. Για παράδειγμα ένα βιβλιοπωλείο μπορεί να δίνει ως ανταμοιβή γραφική ύλη, βιβλία, αναλώσιμα κ.ο.κ.
Ακολούθως θα εξετάσουμε τις τεχνικές και προγραμματιστικές προδιαγραφές της εφαρμογής αυτής.

5.2.1 Τεχνικές Προδιαγραφές

Στα διάφορα σημεία πώλησης – εξυπηρέτησης, υπάρχει ένα τερματικό το οποίο χρησιμοποιείται για την εκτέλεση ενός τμήματος της εφαρμογής, του τμήματος δηλαδή που αφορά τις πωλήσεις προϊόντων με τρόπο πληρωμής το ηλεκτρονικό πορτοφόλι και των αποδόσεων / εξαργυρώσεων πόντων στα πλαίσια προγράμματος εμπιστοσύνης. Το τερματικό αυτό είναι συνδεδεμένο με τον reader GCR410, μέσω του οποίου γίνεται η επικοινωνία με τις κάρτες των πελατών. Το τμήμα αυτό της εφαρμογής χειρίζεται ο εκάστοτε πωλητής ο οποίος είναι εξουσιοδοτημένος διαχειριστής για το κομμάτι αυτό. Ο διαχειριστής πωλήσεων, όπως μπορούμε να τον καλέσουμε, δεν χρειάζεται να κατέχει δική του κάρτα αφού όλες οι λειτουργίες θα εκτελούνται μέσω των καρτών των πελατών.
Ο διαχειριστής πωλήσεων δεν ταυτίζεται με τον διαχειριστή της εφαρμογής μισθοδοσίας ο οποίος έχει αυξημένες ευθύνες. Συγκεκριμένα ο διαχειριστής πωλήσεων έχει εξουσιοδότηση να πραγματοποιεί πωλήσεις χρεώνοντας το ηλεκτρονικό πορτοφόλι του πελάτη, καθώς και να αποδίδει ή να εξαργυρώνει πόντους στα πλαίσια του προγράμματος εμπιστοσύνης. Δεν μπορεί για παράδειγμα να έχει πρόσβαση σε δεδομένα της κάρτας όπως το συνολικό περιεχόμενο του ηλεκτρονικού πορτοφολιού του πελάτη ή των πληροφοριών που αφορούν τη μισθοδοσία του πελάτη. Ο διαχειριστής του προγράμματος μισθοδοσίας έχει την αρμοδιότητα μεν να δει το περιεχόμενο του ηλεκτρονικού πορτοφολιού του εργαζόμενου και να προσθέσει ποσά σε αυτό αλλά δεν έχει εξουσιοδότηση να προσθαφαιρέσει πόντους. Τις λεπτομέρειες αυτές θα εξετάσουμε σε επόμενη ενότητα.

Η παρουσία τερματικού στα σημεία πώλησης αποτελεί απλά μία εξομοίωση της πραγματικής δομής ενός τέτοιου συστήματος αφού το πιο πιθανό σε πραγματικές συνθήκες είναι να υπάρχει μόνο ένας κατάλληλος reader στα σημεία πώλησης ο οποίος θα περιέχει και θα εκτελεί το κομμάτι της εφαρμογής μόνος του. Ένας τέτοιος reader θα περιείχε φυσικά μνήμη εφαρμογών, οθόνη LCD και πληκτρολόγιο.
Κάθε εργαζόμενος κατέχει, όπως ειπώθηκε και στην προηγούμενη εφαρμογή, δική του κάρτα με προσωπικά του στοιχεία και διάφορες άλλες πληροφορίες των εφαρμογών. Για να πραγματοποιήσει αγορές με τη χρήση του ηλεκτρονικού πορτοφολιού της κάρτας του και για να συμμετάσχει στο πρόγραμμα εμπιστοσύνης, πρέπει στο εκάστοτε σημείο πώλησης να εισάγει την κάρτα του στον reader όπως απαιτεί η εφαρμογή. Ανάλογα με το είδος της συναλλαγής που επιθυμεί ο πελάτης εκτελούνται κάποιες διαδικασίες και όταν αυτές ολοκληρωθούν ο πελάτης την αφαιρεί από τον reader.
Για την προσθήκη αξίας στο ηλεκτρονικό πορτοφόλι και για ανάγνωση του ποσού το οποίο περιέχει, ο εργαζόμενος πρέπει να προσέλθει στο διαχειριστή της εφαρμογής μισθοδοσίας ο οποίος χειρίζεται και τμήμα αυτής της εφαρμογής, να εισάγει την κάρτα του στο reader που είναι συνδεδεμένος στον υπολογιστή του διαχειριστή και όταν η εφαρμογή τελειώσει με επεξεργασία των δεδομένων της, να την αφαιρέσει από αυτόν.

5.2.2 Προγραμματιστικές Προδιαγραφές

Οι λειτουργίες τις οποίες πρέπει η συγκεκριμένη εφαρμογή του ηλεκτρονικού πορτοφολιού και του προγράμματος εμπιστοσύνης να υποστηρίζει, είναι:

- Πιστοποίηση ταυτότητας διαχειριστών εφαρμογής: η εφαρμογή θα πρέπει να προσφέρει δυνατότητα ελέγχου της ταυτότητας του διαχειριστή κάθε τμήματός της και να ξεκινάει μόνο μετά από επιτυχή πιστοποίησή τους.
- Δυνατότητα διαχωρισμού δικαιωμάτων για τους διαχειριστές των διαφορετικών τμημάτων της εφαρμογής: στην περίπτωση που η εφαρμογή χειρίζεται ανά τμήμα από διαφορετικούς διαχειριστές πρέπει να διαχωριστούν τα δικαιώματα που θα έχει ο καθένας στη κάρτα και στη χρήση της εφαρμογής.
- Πιστοποίηση ταυτότητας κατόχου κάρτας: οποιαδήποτε χρήση της κάρτας στα πλαίσια της εφαρμογής θα πρέπει να προϋποθέτει επιτυχή ταυτοποίηση του κατόχου της.
- Ασφαλής επικοινωνία και ανταλλαγή δεδομένων μεταξύ τερματικού και καρτών: η εφαρμογή θα πρέπει να εξασφαλίζει με κάποια μέθοδο την ασφαλή ανταλλαγή πληροφοριών μεταξύ τερματικού και καρτών.
- Υποστήριξη πολλαπλών χρηστών: η εφαρμογή θα πρέπει να αντιλαμβάνεται την ταυτότητα του κατόχου της κάρτας και να εκτελείται κανονικά για διαφορετικούς χρήστες και για πολλαπλές συναλλαγές.
- Δυνατότητα ανάγνωσης της ημερομηνίας του συστήματος: το πρόγραμμα που θα τρέχει στα σημεία πώλησης θα πρέπει να έχει τη δυνατότητα να διαβάζει από το σύστημα την εκάστοτε ημερομηνία και ώρα για να κρατάει σωστό αρχείο των συναλλαγών κάθε κάρτας.
- Ύπαρξη ορίων για την ανώτατη αξία ενός e-purse και για το ανώτερο ποσό που μπορεί να προστεθεί στο e-purse τη φορά: θα υπάρχει όριο για το ανώτατο ποσό που μπορεί να περιέχει ένα ηλεκτρονικό πορτοφόλι, όπως και για το ανώτατο ποσό που μπορεί να προστεθεί σε μία συναλλαγή στο ηλεκτρονικό πορτοφόλι.
- Έλεγχος εγκυρότητας ως προς τα παραπάνω όρια: το πρόγραμμα θα πρέπει σε κάθε συναλλαγή να ελέγχει να μη ξεπεραστούν τα όρια για τα ανώτατα ποσά που αφορούν το ηλεκτρονικό πορτοφόλι.
- Αναφορά λαθών που μπορεί να προκύψουν: το πρόγραμμα θα πρέπει να αναφέρει τυχόν λάθη κατά την εκτέλεση μίας λειτουργίας παρέχοντας πληροφορίες όπως ποια είναι η συγκεκριμένη λειτουργία που παρουσίασε σφάλμα.

- Δυνατότητα εντοπισμού και επιδιόρθωσης σφαλμάτων και ανάκτηση εφαρμογής μετά από τερματισμό: πέρα από την αναφορά των λαθών, η εφαρμογή θα πρέπει να μπορεί να αντιμετωπίσει τυχόν σφάλματα όπως και να ξεκινήσει σωστά μετά από αναπάντεχο τερματισμό.

5.2.3 Περιγραφή Σχεδίασης

Η εφαρμογή ηλεκτρονικού πορτοφολιού – προγράμματος εμπιστοσύνης αποτελείται από δύο κύρια τμήματα, όπως είδαμε και στις τεχνικές προδιαγραφές. Από το πρόγραμμα που εκτελείται στα σημεία πώλησης - εξυπηρέτησης και από το αντίστοιχο διαχειριστικό για το ηλεκτρονικό πορτοφόλι. Ο διαχωρισμός αυτός γίνεται κυρίως για λόγους ασφάλειας και προστασίας των προσωπικών δεδομένων και των συναλλαγών. Σύμφωνα και με πραγματικές εφαρμογές τέτοιου περιεχομένου, δεν επιτρέπεται στα σημεία πώλησης να μπορεί ο πωλητής να διαβάσει το περιεχόμενο του ηλεκ/νικού πορτοφολιού του πελάτη, να προσθέσει αξία σε αυτό ή να έχει πρόσβαση σε πληροφορίες της κάρτας που δεν τον αφορούν. Η προσθήκη αξίας στο ηλεκτρονικό πορτοφόλι πραγματοποιείται από άλλη αξιόπιστη πηγή που στην περίπτωσή μας ταυτίζεται με τον διαχειριστή του προγράμματος "Εξαργύρωση".
Έτσι το πρώτο τμήμα της εφαρμογής αφορά τα σημεία πώλησης και χειρίζεται τις αγορές με χρήση του ηλεκτρονικού πορτοφολιού, τις προσθαφαιρέσεις πόντων στα πλαίσια του προγράμματος εμπιστοσύνης ενώ επίσης εκτυπώνει στοιχεία για την κάθε συναλλαγή. Διαχειριστής αυτού του προγράμματος είναι ο διαχειριστής πωλήσεων που αναφέραμε παραπάνω. Το πρόγραμμα αυτό, λόγω του ότι εκτελείται στα σημεία πώλησης – εξυπηρέτησης καλείται "Σημείο Εξυπηρέτησης".
Το δεύτερο μέρος είναι αυτό που πραγματοποιεί τις προσθήκες αξίας στο ηλεκτρονικό πορτοφόλι, την ανάγνωση του περιεχομένου και την πλήρη εξαργύρωσή του σε ειδικές περιπτώσεις. Όσον αφορά το πρόγραμμα εμπιστοσύνης, το τμήμα αυτό της εφαρμογής δεν έχει καμία αρμοδιότητα χειρισμού πέρα από την απλή ανάγνωση του αριθμού των πόντων που έχει κερδίσει ο εργαζόμενος. Το μέρος αυτό το χειρίζεται ο διαχειριστής των προγραμμάτων "Υπερωρίες" και "Εξαργύρωση" και υπάγεται στο πρόγραμμα "Εξαργύρωση". Τις λειτουργίες που το αφορούν τις εξετάσαμε περιληπτικά στην ενότητα 5.1.3.4 αλλά θα τις συζητήσουμε λεπτομερέστερα σε επόμενη ενότητα.
Τα δύο τμήματα της εφαρμογής στα οποία μόλις αναφερθήκαμε θα αναλυθούν στις ενότητες που ακολουθούν.

5.2.3.1 Πρόγραμμα "Εξαργύρωση" - 2η εφαρμογή

Τη δομή του προγράμματος "Εξαργύρωση" έχουμε δει στο Σχήμα 5.1.3.2 της αντίστοιχης ενότητας. Η ροή του προγράμματος για το τμήμα που αφορά την εφαρμογή ηλεκ/νικού πορτοφολιού και προγράμματος εμπιστοσύνης ακολουθεί τα ίδια βήματα και είναι περιληπτικά:

- Επίπεδο Α:
  Ο διαχειριστής ανοίγει το πρόγραμμα, το οποίο ανοίγει κανάλι επικοινωνίας με τον reader και περιμένει την είσοδο μίας κάρτας. Με την είσοδο μίας κάρτας, ανοίγει μία σύνοδος (session) μεταξύ συστήματος και κάρτας. Επίσης ανοίγουν δύο αρχεία text για γράψιμο (αρχείο log και αρχείο receipt για τον εργαζόμενο).
- Σημείο Ελέγχου Α:
  Μετά την είσοδο της κάρτας το πρόγραμμα ζητάει να ελέγξει την ταυτότητα του διαχειριστή το οποίο γίνεται με τη σύγκριση του κωδικού που θα εισάγει με ένα κλειδί αποθηκευμένο σε κάθε κάρτα. Αν ο διαχειριστής δεν παρουσιάσει τον σωστό κωδικό το πρόγραμμα κλείνει το δίαυλο επικοινωνίας με τον reader και την κάρτα αφού ενημερώσει το log αρχείο και τερματίζει. Αν η ταυτοποίηση είναι επιτυχής περνάει στο σημείο ελέγχου Β.

- Σημείο Ελέγχου Β:
  Το πρόγραμμα καλωσορίζει τον κάτοχο της κάρτας και ζητάει την παρουσίαση του κωδικού του. Αν η παρουσίαση είναι ανεπιτυχής και έχει ξεπεραστεί το όριο προσπαθειών για τον κάτοχο, η κάρτα κλειδώνει και το πρόγραμμα τερματίζει με τις κατάλληλες ενέργειες. Αλλιώς το πρόγραμμα προχωράει στο σημείο ελέγχου Γ.
- Σημείο Ελέγχου Γ:
  Ο διαχειριστής τίθεται να επιλέξει ανάμεσα σε 5 λειτουργίες (3 αφορούν το ηλεκτρονικό πορτοφόλι και 1 το πρόγραμμα εμπιστοσύνης). Αν η λειτουργία είναι ο τερματισμός του προγράμματος, αυτό κλείνει το δίαυλο επικοινωνίας με την κάρτα και τον reader και τερματίζει. Αν επιλέξει κάποια από τις άλλες λειτουργίες, το πρόγραμμα περνάει στο επίπεδο Β.
- Επίπεδο Β:
  Το πρόγραμμα εκτελεί τη λειτουργία που επέλεξε ο διαχειριστής σύμφωνα με τα δεδομένα που του ορίζει. Τα αποτελέσματα της εκτέλεσης της λειτουργίας που εκτελεί το πρόγραμμα εγγράφονται στο αρχείο log και στο αρχείο receipt για την εξυπηρέτηση του κατόχου της κάρτας. Μετά την εκτέλεση της λειτουργίας το πρόγραμμα επιστρέφει στο σημείο ελέγχου Γ, όπου ο διαχειριστής μπορεί να διαλέξει την επόμενη λειτουργία που επιθυμεί να εκτελέσει.

Όπως έχει ήδη αναφερθεί, στο πρόγραμμα αυτό χρησιμοποιούνται δύο από τους κωδικούς της κάρτας, ο κωδικός του κατόχου και το μυστικό κλειδί του διαχειριστή του προγράμματος.

5.2.3.2 Λειτουργίες Προγράμματος "Εξαργύρωση" - 2η εφαρμογή

Οι 4 λειτουργίες που περιέχονται στο πρόγραμμα "Εξαργύρωση" και αφορούν την εφαρμογή ηλεκ/νικού πορτοφολιού και προγράμματος εμπιστοσύνης είναι αναλυτικά οι ακόλουθες:

- Ανάγνωση περιεχομένου ηλεκτρονικού πορτοφολιού εργαζόμενου:
  Ο διαχειριστής με τη λειτουργία αυτή διαβάζει το ποσό που περιέχεται στο ηλεκτρονικό πορτοφόλι του εργαζόμενου και τον ενημερώνει για αυτό. Το ποσό, το οποίο μπορεί να είναι και μηδενικό, εκτυπώνεται στην οθόνη και εγγράφεται και στο αρχείο log.
- Προσθήκη χρημάτων στο ηλεκτρονικό πορτοφόλι του εργαζόμενου:
  Με τη λειτουργία αυτή ο διαχειριστής μπορεί να προσθέσει κάποιο ποσό στο e-purse του κατόχου της κάρτας. Η λειτουργία αυτή έχει δύο περιορισμούς που αφορούν τη μέγιστη αξία που μπορεί να προστεθεί στο ηλεκτρονικό πορτοφόλι σε μία συναλλαγή και την ελάχιστη αξία του ηλεκτρονικού πορτοφολιού πάνω από την οποία δεν επιτρέπονται καθόλου προσθήκες.
  Συγκεκριμένα, έχει οριστεί στην πολιτική αυτής της εφαρμογής ότι αν η αξία του πορτοφολιού είναι μεγαλύτερη των 3000€, δεν μπορεί να προστεθεί οποιαδήποτε περαιτέρω αξία σε αυτό. Στην αντίθετη περίπτωση κατά την οποία το περιεχόμενο του ηλεκτρονικού πορτοφολιού αντιστοιχεί σε ποσό μικρότερο ή ίσο με τα 3000€, μπορεί να γίνει οποιαδήποτε προσθήκη αξίας μέχρι του ποσού των 1500€. Είναι έτσι προφανές ότι η μέγιστη αξία που δύναται να περιέχει το ηλεκτρονικό πορτοφόλι ενός εργαζόμενου είναι τα 4500€.
  Κατά την εκτέλεση λοιπόν αυτής της λειτουργίας, το πρόγραμμα ελέγχει την παρούσα αξία του e–purse. Αν η αξία αυτή υπερβαίνει τα 3000€, ενημερώνει τον διαχειριστή και τον κάτοχο της κάρτας ότι δεν μπορεί να γίνει καμία προσθήκη αξίας, εγγράφει την αξία στην οθόνη και το αρχείο log και επιστρέφει στο σημείο επιλογής επόμενης λειτουργίας. Αν η παρούσα αξία του e–purse δεν υπερβαίνει τα 3000€, προτρέπει τον διαχειριστή να εισάγει την αξία που επιθυμεί να προσθέσει στο ηλεκτρονικό πορτοφόλι, μέχρι του ποσού των 1500€. Μόλις το ποσό που θα εισάγει ο διαχειριστής κριθεί από το πρόγραμμα έγκυρο, πραγματοποιείται η προσθήκη της

αξίας, τυπώνεται στην οθόνη η επιβεβαίωση της προσθήκης αυτής και τα στοιχεία της συναλλαγής εγγράφονται και στα δύο αρχεία log και receipt.

- Εξαργύρωση ολόκληρου του ηλεκτρονικού πορτοφολιού του εργαζόμενου:
  Η λειτουργία αυτή πραγματοποιεί αφαίρεση από το ηλεκτρονικό πορτοφόλι όλης της αξίας του για να αποδοθεί στον κάτοχο της κάρτας. Χρησιμοποιείται μόνο σε ειδικές περιπτώσεις, όπως όταν ο κάτοχος επιθυμεί την λήξη της κάρτας του. Όταν ο διαχειριστής επιλέγει αυτή τη λειτουργία, το πρόγραμμα διαβάζει την παρούσα αξία του πορτοφολιού. Αν είναι μηδενική ενημερώνει τον διαχειριστή και το κάτοχο της κάρτας γι' αυτό και δεν εκτελείται. Αν υπάρχει αξία στο πορτοφόλι, το πρόγραμμα την αφαιρεί, ενημερώνει τον διαχειριστή για το ποσό που αφαιρέθηκε και πρέπει να αποδοθεί στον κάτοχο και εγγράφει αυτές τις πληροφορίες στα δύο αρχεία log και receipt. Ο διαχειριστής από τη στιγμή που επιλέγει την λειτουργία, δεν έχει άλλη ανάμειξη σε αυτή, δεν ορίζει αυτός δηλαδή το ποσό που θα αφαιρεθεί από το πορτοφόλι, απλά το πληροφορείται στο τέλος της λειτουργίας για να το αποδώσει στον εργαζόμενο. Γενικότερα, στο πρόγραμμα "Εξαργύρωση" δεν υπάρχει λειτουργία μερικής αφαίρεσης αξίας από το πορτοφόλι γιατί θεωρείται ότι το ηλεκτρονικό πορτοφόλι χρησιμοποιείται ως μέρος αποθήκευσης προπληρωμένης αξίας η οποία απλά αναλώνεται.
- Ανάγνωση πόντων του προγράμματος εμπιστοσύνης:
  Η λειτουργία αυτή χρησιμεύει μόνο για την πληροφόρηση του εργαζόμενου για τον αριθμό των πόντων που του έχουν αποδοθεί μέχρι τότε στο πρόγραμμα εμπιστοσύνης που έχει συμμετάσχει. Το πρόγραμμα τυπώνει στην οθόνη και εγγράφει στο αρχείο log τον αριθμό των πόντων.

5.2.3.3 Πρόγραμμα "Σημείο Εξυπηρέτησης"

Το πρόγραμμα αυτό έχει πανομοιότυπη δομή με αυτό του προγράμματος "Εξαργύρωση" που φαίνεται στο Σχήμα 5.1.3.2 και αναλύεται στην αντίστοιχη ενότητα. Οι μόνες διαφορές είναι στην ταυτότητα του διαχειριστή, στα αρχεία που ανοίγει το πρόγραμμα και στις λειτουργίες που καλείται να επιλέξει ο διαχειριστής.
Συγκεκριμένα, στο επίπεδο Α που το πρόγραμμα ξεκινάει, ανοίγει για γράψιμο ένα μόνο αρχείο, το αρχείο log του προγράμματος "Σημείο Εξυπηρέτησης". Έπειτα, στο σημείο ελέγχου Α, το πρόγραμμα ζητάει την επαλήθευση της ταυτότητας του διαχειριστή πωλήσεων ο οποίος είναι ο εξουσιοδοτημένος χρήστης. Τέλος οι λειτουργίες ανάμεσα στις οποίες πρέπει να επιλέξει ο διαχειριστής πωλήσεων στο σημείο ελέγχου Γ είναι 4 και εκτός αυτής του τερματισμού, είναι διαφορετικές από αυτές που έχουν αναφερθεί στα άλλα προγράμματα.
Για την εκτέλεση αυτού του προγράμματος χρησιμοποιούνται δύο από τους κωδικούς της κάρτας, ο κωδικός του κατόχου και το μυστικό κλειδί του διαχειριστή πληρωμών του προγράμματος.

5.2.3.4 Λειτουργίες Προγράμματος "Σημείο Εξυπηρέτησης"

Οι 3 λειτουργίες του προγράμματος (πέραν αυτής του τερματισμού) αφορούν την εφαρμογή ηλεκ/νικού πορτοφολιού και προγράμματος εμπιστοσύνης και αναλύονται ακολούθως:

- Πραγματοποίηση πώλησης με χρήση ηλεκτρονικού πορτοφολιού πελάτη:
  Ο διαχειριστής πωλήσεων με τη λειτουργία αυτή πραγματοποιεί μία πώληση η οποία πληρώνεται από τον πελάτη με τη χρήση του ηλεκτρονικού του πορτοφολιού. Ταυτόχρονα με την πώληση αυτή και ανάλογα με το ποσό που καλείται ο πελάτης να πληρώσει, αποδίδονται στον πελάτη και πόντοι από το πρόγραμμα εμπιστοσύνης που εφαρμόζεται στο σημείο πώλησης.
  Συγκεκριμένα, μόλις ο διαχειριστής επιλέξει τη λειτουργία αυτή, το πρόγραμμα διαβάζει εσωτερικά το ποσό που περιέχεται στο ηλεκτρονικό πορτοφόλι του πελάτη. Αν το ποσό είναι μηδενικό, δηλαδή το e-purse του πελάτη είναι άδειο, πληροφορεί

γι' αυτό τον πωλητή (διαχειριστή πωλήσεων) και τον κάτοχο και επιστρέφει στο σημείο επιλογής επόμενης λειτουργίας.
Αν υπάρχει αξία στο ηλεκτρονικό πορτοφόλι του πελάτη, το πρόγραμμα ζητάει από τον πωλητή να εισάγει το ποσό της αγοράς που επιθυμεί να πραγματοποιήσει ο πελάτης. Με την εισαγωγή του ποσού, ελέγχει πάλι αν στο ηλεκτρονικό πορτοφόλι του πελάτη υπάρχει τέτοια διαθέσιμη αξία. Αν όχι, προτρέπει τον πωλητή να εισάγει ένα έγκυρο ποσό. Αν ο πελάτης αποφασίσει να μην πραγματοποιήσει καμία αγορά λόγω μη διαθέσιμου υπολοίπου, ο πωλητής εισάγει το ποσό των 0€ ως ποσό αγοράς.
Μετά την εισαγωγή έγκυρου ποσού, το πρόγραμμα χρεώνει το ηλεκτρονικό πορτοφόλι του πελάτη (αφαιρεί αξία) και υπολογίζει σύμφωνα με τις παραμέτρους του προγράμματος εμπιστοσύνης, τον αριθμό των πόντων που θα αποδοθούν στον πελάτη (θα αποθηκευτούν δηλαδή σε κατάλληλο μετρητή στην κάρτα του). Το πρόγραμμα έπειτα ενημερώνει για το ποσό της αγοράς που πραγματοποίησε ο πελάτης και τους πόντους που του αντιστοιχούν και αποθηκεύει τα στοιχεία της συναλλαγής στο αρχείο log.
Είναι δυνατόν το ποσό της συναλλαγής να μην ικανοποιεί τις συνθήκες του προγράμματος εμπιστοσύνης και να μην αποδοθούν καθόλου πόντοι στον πελάτη. Το πρόγραμμα εμπιστοσύνης που παραμετροποιήθηκε σε αυτή την εφαρμογή, ρυθμίστηκε έτσι ούτως ώστε για κάθε 20€ που δίνει ο πελάτης να ανταμείβεται με 5 πόντους. Έτσι αν κάνει μία αγορά των 46€, θα του αποδοθούν σύμφωνα και με το τύπο 4.2.3.3 (σελ.38): $5 \bullet \text{int}\left(\frac{46}{20}\right) = 5 \bullet 2 = 10$ πόντοι.

- Ανάγνωση πόντων του προγράμματος εμπιστοσύνης:
  Η λειτουργία αυτή είναι η ίδια που υπάρχει και στο πρόγραμμα "Εξαργύρωση" και χρησιμεύει για την πληροφόρηση του εργαζόμενου για τον αριθμό των πόντων που έχει στην κάρτα του. Το πρόγραμμα τυπώνει στην οθόνη και εγγράφει στο αρχείο log τον αριθμό των πόντων.
- Εξαργύρωση πόντων του προγράμματος εμπιστοσύνης:
  Όταν ο πελάτης επιθυμεί την εξαργύρωση κάποιων πόντων, την ανταλλαγή τους δηλαδή με κάποιο από τα δώρα που τους αντιστοιχούν, ο διαχειριστής πωλήσεων εκτελεί αυτή τη λειτουργία. Το πρόγραμμα διαβάζει αρχικά τον αριθμό των πόντων που έχει ο πελάτης στην κάρτα του. Αν ο αριθμός αυτός είναι μηδενικός, ενημερώνει γι' αυτό το διαχειριστή και τον πελάτη και τερματίζει τη διαδικασία αυτή.
  Αν από την άλλη υπάρχουν πόντοι αποθηκευμένοι στη κάρτα του πελάτη, πληροφορεί το διαχειριστή πωλήσεων για το μέγιστο αριθμό πόντων που μπορεί ο πελάτης να εξαργυρώσει (ο αριθμός αυτός ταυτίζεται με τον αριθμό τον πόντων που είναι αποθηκευμένος στην κάρτα) και προτρέπει τον διαχειριστή να εισάγει τον αριθμό των πόντων που θα εξαργυρωθούν τελικά.
  Το πρόγραμμα στη συνέχεια ελέγχει αν οι πόντοι που ο διαχειριστής εισήγαγε είναι διαθέσιμοι στην κάρτα του πελάτη. Στην περίπτωση που δεν είναι ζητάει από τον διαχειριστή να εισάγει ένα επιτρεπτό αριθμό πόντων. Όταν πλέον ο αριθμός πόντων που εισάγει ο διαχειριστής είναι διαθέσιμος στην κάρτα, το πρόγραμμα πραγματοποιεί την αφαίρεσή τους, τυπώνει στην οθόνη τον αριθμό των πόντων που εξαργυρώθηκαν και εγγράφει τις αντίστοιχες πληροφορίες στο αρχείο log της εφαρμογής.
- Τερματισμός προγράμματος:
  Η τελευταία λειτουργία του προγράμματος αυτού, η οποία υπάρχει και σε όλα τα άλλα προγράμματα που έχουμε συζητήσει μέχρι τώρα είναι αυτή του τερματισμού. Αποτελεί τον σωστό τρόπο τερματισμού ενός προγράμματος και πρέπει να χρησιμοποιείται πάντα αντί για οποιαδήποτε άλλη μέθοδο. Με τη λειτουργία αυτή, το πρόγραμμα κλείνει αρχικά τα όποια αρχεία έχει ανοίξει για εγγραφή, έπειτα κλείνει τη σύνοδο (session) με την κάρτα και τέλος κλείνει το κανάλι επικοινωνίας με τον reader.

5.3 Εφαρμογή Εισιτηρίων

Η τρίτη και τελευταία εφαρμογή της διπλωματικής αυτής, η οποία στηρίχτηκε επίσης στη τεχνολογία των έξυπνων καρτών GemClub αφορά τα μέσα συγκοινωνίας και είναι μία εξομοίωση ενός απλού συστήματος έκδοσης εισιτηρίων με τη χρήση έξυπνων καρτών. Η εφαρμογή αυτή υλοποιήθηκε στην ίδια κάρτα που χρησιμοποιήθηκε και για τις προηγούμενες εφαρμογές με τις κατάλληλες προσθήκες και παραμετροποιήσεις. Συνήθως όμως για μία τέτοια εφαρμογή χρησιμοποιείται κάρτα αφιερωμένη στην εφαρμογή.
Η ουσία αυτής της εφαρμογής είναι ότι ο κάτοχος της κάρτας τη χρησιμοποιεί ως κάρτα προπληρωμένης αξίας όσον αφορά τα εισιτήρια στα μέσα συγκοινωνίας. Έτσι αντί να αγοράζει τα κοινά εισιτήρια, πληρώνει και αποθηκεύει την αξία τους στην κάρτα την οποία επικυρώνει κάθε φορά που χρησιμοποιεί ένα μέσο μαζικής μεταφοράς. Η κάρτα χρησιμοποιείται για όλα τα μέσα συγκοινωνίας και όταν εξαντληθεί μπορεί να επαναφορτιστεί (να ανανεωθεί με νέα αξία εισιτηρίων).
Κατά την χρήση της σε κάποιο μέσο μαζικής μεταφοράς, το αποδεικτικό της επικύρωσης της κάρτας, της χρήσης δηλαδή εισιτηρίου μπορεί να είναι είτε μία "απόδειξη" που παράγεται από τη συσκευή υποδοχής της κάρτας και πρέπει ο κάτοχος να κρατάει μέχρι την έξοδό του από το μέσο συγκοινωνίας, είτε η εγγραφή των πληροφοριών της επικύρωσης της κάρτας σε ένα αρχείο της κάρτας. Σε πιθανό έλεγχο εισιτηρίων, ο ελεγκτής διαβάζει την εκάστοτε κάρτα με τη δική του συσκευή ανάγνωσης.
Σε πραγματικές εφαρμογές, χρησιμοποιούνται ασύρματες κάρτες τις οποίες οι κάτοχοι πρέπει να "περάσουν" από ειδικούς reader. Οι reader είναι συνδεδεμένοι με μηχανισμούς που δεν επιτρέπουν την είσοδο στα μέσα συγκοινωνίας αν δεν έχει γίνει επικύρωση της κάρτας.
Στα πλαίσια της δικής μας εφαρμογής που απλά στοχεύει να παρουσιάσει το μηχανισμό που οι κάρτες χρησιμοποιούν για μία τέτοια εφαρμογή, χρησιμοποιούμε ως αποδεικτικό στοιχείο της επικύρωσης της κάρτας την εγγραφή της συναλλαγής σε ένα αρχείο της εφαρμογής. Ο reader GCR410 εξομοιώνει την συσκευή υποδοχής των καρτών σε ένα μέσο μαζικής μεταφοράς. Τέλος ως μέσα μαζικής μεταφοράς έχουμε θεωρήσει τα λεωφορεία και τρόλεϊ, τον ηλεκτρικό σιδηρόδρομο και το μετρό.
Στις επόμενες ενότητες θα εξετάσουμε τις προδιαγραφές της εφαρμογής στις οποίες έχει βασιστεί η σχεδίασή της.

5.3.1 Τεχνικές Προδιαγραφές

Στο σημείο όπου γίνεται η φόρτωση της κάρτας με την αξία των εισιτηρίων που επιθυμεί ο κάτοχός της, υπάρχει ένας υπολογιστής (host σύστημα) με τον οποίο είναι συνδεδεμένος ο reader GCR410. Ο reader είναι αυτός που πραγματοποιεί τεχνικά την επικοινωνία με τις κάρτες. Το τμήμα αυτό της εφαρμογής χειρίζεται ένας εξουσιοδοτημένος διαχειριστής που θα ονομάζεται από εδώ και πέρα για ευκολία διαχειριστής εισιτηρίων. Ο διαχειριστής αυτός δεν χρειάζεται να κατέχει δική του κάρτα αφού οι λειτουργίες ασφαλείας που χρειάζονται εκτελούνται μέσω των καρτών των πελατών όταν αυτοί προσέλθουν για την εξαγορά εισιτηρίων.
Για να γίνει η φόρτωση αξίας εισιτηρίων στην κάρτα, ο κάτοχος πρέπει να προσέλθει στο διαχειριστή εισιτηρίων, να εισάγει την κάρτα του στο reader και όταν η εφαρμογή τελειώσει να την αφαιρέσει από αυτόν.
Η εφαρμογή αυτή δεν έχει καμία σχέση με τις δύο προηγούμενες εφαρμογές που αναλύσαμε (μισθοδοσίας και ηλεκτρονικού πορτοφολιού - προγράμματος εμπιστοσύνης) πέραν του γεγονότος ότι υλοποιείται στην ίδια κάρτα. Έτσι ο διαχειριστής εισιτηρίων δεν έχει καμία πρόσβαση σε λειτουργίες που ανήκουν στις άλλες εφαρμογές ούτε μπορεί να επεξεργαστεί δεδομένα τους. Αντίστοιχα οι άλλοι διαχειριστές δεν έχουν δικαίωμα προσθαφαίρεσης αξίας στο μετρητή που χρησιμοποιείται για τα εισιτήρια. Αυτό θα φανεί κυρίως σε ανάλυση των λειτουργιών της εφαρμογής που θα ακολουθήσει.
Στα μέσα μαζικής μεταφοράς θεωρείται ότι υπάρχουν ειδικές συσκευές υποδοχής των καρτών που τρέχουν ένα μικρό πρόγραμμα αποθηκευμένο στη μνήμη τους, το οποίο απλά

αφαιρεί αξία από τις κάρτες. Τέτοιες συσκευές προφανώς δεν χρειάζεται να είναι συνδεδεμένες με τερματικό. Τις συσκευές αυτές εξομοιώνει ο reader GCR410, ο οποίος βέβαια λειτουργεί μόνο μέσω σύνδεσης με υπολογιστή. Για την επικύρωση της κάρτας (του ηλεκτρονικού του "εισιτηρίου") ο κάτοχος εισάγει την κάρτα στη συσκευή και μόλις η συσκευή τον ειδοποιήσει την αφαιρεί.

5.3.2 Προγραμματιστικές Προδιαγραφές

Τα κύρια στοιχεία που πρέπει να υπάρχουν σε αυτή την εφαρμογή είναι:

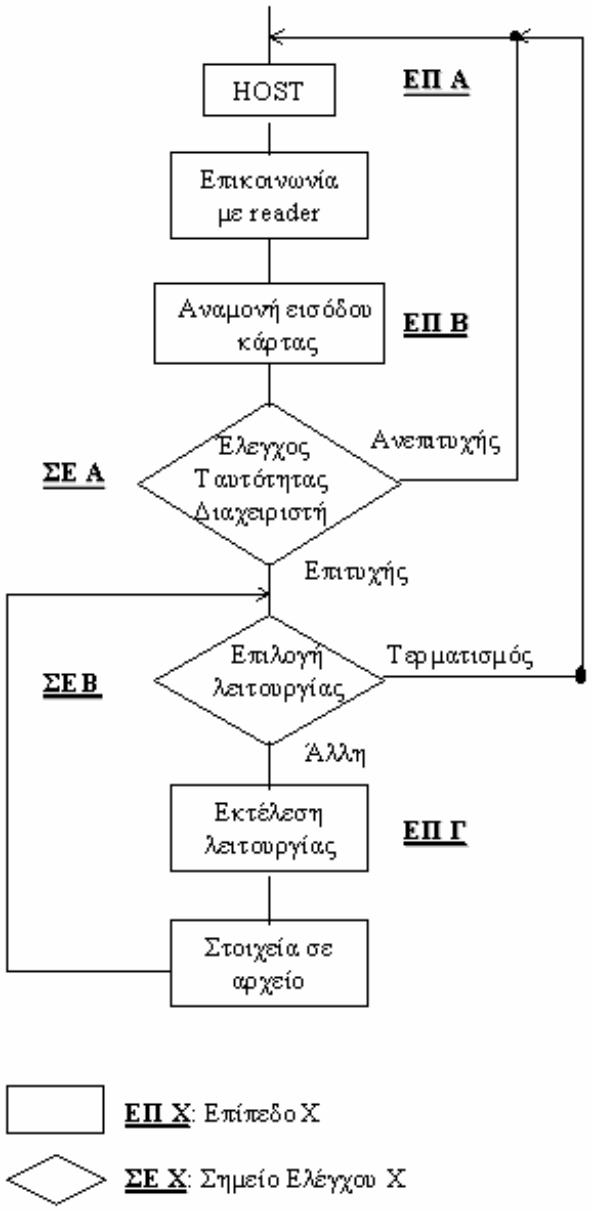
- Πιστοποίηση ταυτότητας διαχειριστή εφαρμογής: η εφαρμογή θα πρέπει να προσφέρει δυνατότητα ελέγχου της ταυτότητας του διαχειριστή εισιτηρίων.
- Ασφαλής επικοινωνία μεταξύ τερματικού φόρτωσης και καρτών: η εφαρμογή θα πρέπει να παρέχει ασφαλή επικοινωνία μεταξύ του τερματικού στο οποίο γίνεται η φόρτωση της αξίας των εισιτηρίων και των καρτών.
- Συνεχής ροή εφαρμογής στα μέσα μαζικής μεταφοράς: το πρόγραμμα που θεωρείται ότι θα τρέχει στις συσκευές υποδοχής καρτών στα μέσα συγκοινωνίας θα πρέπει να είναι συνέχεια έτοιμο για εισαγωγή καρτών στις συσκευές.
- Διαχωρισμός μεταξύ κανονικών / μειωμένων εισιτηρίων: το πρόγραμμα φόρτωσης αξίας εισιτηρίων θα πρέπει να προβλέπει την περίπτωση κατά την οποία τα εισιτήρια που επιθυμεί ο κάτοχός της είναι μειωμένα.
- Διαχωρισμός μεταξύ μέσων συγκοινωνίας: το πρόγραμμα φόρτωσης αξίας εισιτηρίων θα πρέπει επίσης να δίνει στον διαχειριστή εισιτηρίων τη δυνατότητα να "φορτώνει" εισιτήρια για όλα τα μέσα μεταφοράς τα οποία έχουν διαφορετική αξία για τα εισιτήριά τους.
- Αναφορά λαθών που μπορεί να προκύψουν: το πρόγραμμα θα πρέπει να αναφέρει τυχόν λάθη κατά την εκτέλεση μίας λειτουργίας.
- Δυνατότητα επιδιόρθωσης σφαλμάτων: μετά την αναφορά κάποιου σφάλματος η εφαρμογή θα πρέπει να μπορεί να το αντιμετωπίσει.

5.3.3 Περιγραφή Σχεδίασης

Όπως ειπώθηκε και νωρίτερα η εφαρμογή των εισιτηρίων χωρίζεται σε δύο κομμάτια. Στο τμήμα αυτό που αφορά την φόρτωση της κάρτας με αξία εισιτηρίων και σε εκείνο που εκτελείται στα μέσα συγκοινωνίας όταν κάποιος επικυρώνει την κάρτα του ως εισιτήριο.
Το πρώτο τμήμα της εφαρμογής καλείται "Κέντρο Εισιτηρίων" και το διαχειρίζεται ο διαχειριστής εισιτηρίων που έχει την ευθύνη να αποθηκεύει σε κατάλληλο χώρο στην κάρτα την αξία εισιτηρίων που επιθυμεί ο κάτοχός της. Όπως θα δούμε και παρακάτω, στην ανάλυση της δομής του προγράμματος, δεν ζητείται πουθενά η ταυτότητα του κατόχου της κάρτας. Αυτό συμβαίνει γιατί μία τυχούσα χρήση μίας κάρτας εισιτηρίων από μη νόμιμο χρήστη για τη φόρτωση αξίας εισιτηρίων δεν θα ήταν επιζήμια, ούτε για τον πραγματικό κάτοχο ούτε για τον πάροχο των καρτών. Ο μη νόμιμος χρήστης θα έπρεπε ούτως ή άλλως να πληρώσει την αξία των εισιτηρίων που επιθυμεί να φορτώσει στην κάρτα. Επίσης, δεν θα υπήρχε κίνδυνος ούτε για τις άλλες εφαρμογές που τρέχουν στις συγκεκριμένες κάρτες που έχουμε σχεδιάσει, αφού εκείνες προστατεύονται και από κωδικούς άλλων διαχειριστών και από τους κωδικούς των πραγματικών κατόχων.
Το δεύτερο μέρος καλείται "Μετακίνηση" και δεν έχει διαχειριστή αφού θεωρείται ότι εξομοιώνει το πρόγραμμα που τρέχει μόνο του στη μνήμη εφαρμογών κάθε reader που είναι τοποθετημένος σε μέσα μαζικής μεταφοράς. Επειδή το πρόγραμμα "Μετακίνηση" πρέπει να λειτουργεί με διαφορετική αξία σε κάθε μέσο μεταφοράς, υπάρχει μία παράμετρος μέσα στο πρόγραμμα που ορίζει σε ποιο μέσο μεταφοράς θα χρησιμοποιηθεί, βάση της οποίας ορίζεται και η αξία του εισιτηρίου (μειωμένου ή κανονικού).
Τα δύο αυτά τμήματα της εφαρμογής εξηγούνται ακολούθως.

5.3.3.1 Πρόγραμμα "Κέντρο Εισιτηρίων"

Τη δομή του προγράμματος "Κέντρο Εισιτηρίων" παρατηρούμε στο Σχήμα 5.3.3.1 που παρατίθεται ακολούθως.



Σχήμα 5.3.3.1 - Δομή Προγράμματος «Κέντρο Εισιτηρίων»

Η ροή του προγράμματος αυτού μοιάζει με τη ροή των προγραμμάτων "Εξαργύρωση" και "Κέντρο Εξυπηρέτησης" με κύρια όμως διαφορά την έλλειψη ελέγχου για την ταυτότητα του κατόχου. Συγκεκριμένα ακολουθούνται τα εξής βήματα:

- Επίπεδο Α:
  Ο διαχειριστής εισιτηρίων ανοίγει το πρόγραμμα, το οποίο ανοίγει κανάλι επικοινωνίας με τον reader, ανοίγει για γράψιμο ένα αρχείο log της εφαρμογής και περνάει στο επίπεδο Β όπου αναμένει είσοδο κάρτας στον reader.

- Επίπεδο Β:
  Με την είσοδο μίας κάρτας, το πρόγραμμα ανοίγει μία σύνοδο (session) με την κάρτα και περνάει στο σημείο ελέγχου Α.
- Σημείο Ελέγχου Α:
  Το πρόγραμμα ζητάει να ελέγξει την ταυτότητα του διαχειριστή εισιτηρίων. Η ταυτοποίηση γίνεται με τη σύγκριση του κωδικού που θα εισάγει με ένα κλειδί αποθηκευμένο σε κάθε κάρτα που αντιστοιχεί στον διαχειριστή αυτόν και μόνο. Αν ο διαχειριστής δεν παρουσιάσει τον σωστό κωδικό το πρόγραμμα κλείνει το δίαυλο επικοινωνίας με τον reader και την κάρτα, ενημερώνει το αρχείο log και τερματίζει. Αν η ταυτοποίηση είναι επιτυχής περνάει στο σημείο ελέγχου Β.
- Σημείο Ελέγχου Β:
  Ο διαχειριστής τίθεται να επιλέξει ανάμεσα σε 3 λειτουργίες (συμπεριλαμβανόμενου και του κανονικού τερματισμού του προγράμματος). Αν επιλέξει τερματισμό, το πρόγραμμα κλείνει τη σύνοδο με την κάρτα και το κανάλι με τον reader και τερματίζει, αλλιώς περνάει στο επίπεδο Β.
- Επίπεδο Γ:
  Το πρόγραμμα εκτελεί τη λειτουργία που επέλεξε ο διαχειριστής σύμφωνα με τα δεδομένα που ο ίδιος εισάγει μετά από προτροπή του προγράμματος. Τα αποτελέσματα της εκτέλεσης της λειτουργίας εγγράφονται στο αρχείο log. Μετά την εκτέλεση της λειτουργίας το πρόγραμμα επιστρέφει στο σημείο ελέγχου Β, όπου αναμένει νέα επιλογή λειτουργίας.

Στο πρόγραμμα αυτό χρησιμοποιείται για ασφάλεια μόνο το μυστικό κλειδί του διαχειριστή εισιτηρίων.

5.3.3.2 Λειτουργίες Προγράμματος "Κέντρο Εισιτηρίων"

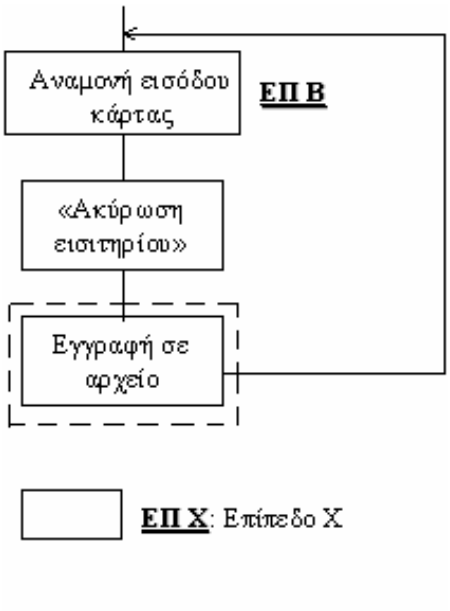
Οι 2 λειτουργίες που μπορεί ο διαχειριστής εισιτηρίων να εκτελέσει σε αυτό το πρόγραμμα, εκτός από τη λειτουργία του κανονικού τερματισμού του, είναι οι εξής:

- Αρχικοποίηση πληροφοριών εισιτηρίων:
  Με τη λειτουργία αυτή ο διαχειριστής ενημερώνει την κάρτα για το είδος των εισιτηρίων που χρησιμοποιεί ο κάτοχός της, αν δηλαδή είναι μειωμένης ή κανονικής αξίας. Το πρόγραμμα ζητάει από τον διαχειριστή να ορίσει αν τα εισιτήρια που χρησιμοποιεί ο κάτοχος της κάρτας είναι μειωμένα ή κανονικά και αφού ο διαχειριστής εισάγει την πληροφορία αυτή, ενημερώνει το κατάλληλο αρχείο στην κάρτα και επιστρέφει στο επίπεδο επιλογής λειτουργίας.
  Είναι πιθανό να υπάρχει αποθηκευμένη αξία εισιτηρίων ενός είδους στην κάρτα και με την αρχικοποίηση το είδος αυτό να μετατραπεί. Για παράδειγμα μπορεί η κάρτα να έχει αποθηκευμένη με κατάλληλο τρόπο αξία 10 κανονικών εισιτηρίων λεωφορείου που το καθένα κοστίζει 45 λεπτά. Έχει δηλαδή αποθηκευμένη αξία 4,5€ (450 λεπτών). Αν δεν εξαντληθούν και με την αρχικοποίηση αλλάξει το είδος των εισιτηρίων από κανονικά σε μειωμένα, η αποθηκευμένη αξία δεν χάνεται, απλά μετατρέπεται και έτσι αντιστοιχεί σε 22 μειωμένα εισιτήρια λεωφορείου (αξίας 20 λεπτών το καθένα). Βέβαια η αξία αυτή θα μπορεί να χρησιμοποιηθεί πλέον και για 11 μειωμένα εισιτήρια μετρό (αξίας 40 λεπτών το καθένα) ή για 15 μειωμένα εισιτήρια ηλεκτρικού σιδηρόδρομου (αξίας 30 λεπτών το καθένα).
  Η πληροφορία για το είδος των εισιτηρίων που χρησιμοποιεί ο κάτοχος της κάρτας είναι απαραίτητος για την εκτέλεση του προγράμματος "Μετακίνηση".
- Προσθήκη αξίας εισιτηρίων:
  Η λειτουργία αυτή πραγματοποιεί την προσθήκη της αξίας των εισιτηρίων που επιθυμεί ο κάτοχος της κάρτας να αγοράσει. Αρχικά το πρόγραμμα διαβάζει από την κάρτα το είδος των εισιτηρίων που χρησιμοποιεί ο κάτοχος της κάρτας. Στη συνέχεια ζητάει από τον διαχειριστή να επιλέξει το μέσο μεταφοράς για το οποίο προορίζονται

τα εισιτήρια. Μπορεί να επιλέξει μεταξύ λεωφορείων, μετρό και ηλεκτρικού σιδηρόδρομου. Μόλις ο διαχειριστής επιλέξει το μέσο μεταφοράς, το πρόγραμμα τυπώνει στην οθόνη την αξία ενός εισιτηρίου για αυτό το μέσο, μειωμένου ή κανονικού ανάλογα με το είδος που έχει διαβάσει νωρίτερα από την κάρτα. Στη συνέχεια ρωτάει τον διαχειριστή τον αριθμό των εισιτηρίων που ο κάτοχος της κάρτας επιθυμεί να αγοράσει και όταν ο διαχειριστής εισάγει τον αριθμό, τον πληροφορεί (τυπώνοντας στην οθόνη) ποια είναι η συνολική αξία των εισιτηρίων που μόλις δήλωσε ότι επιθυμεί ο κάτοχος της κάρτας. Προτρέπει τον διαχειριστή να επιβεβαιώσει την επιθυμία του και αν αυτό γίνει, κάνει την προσθήκη της αξίας στην κάρτα και ενημερώνει τον διαχειριστή ότι η συναλλαγή ολοκληρώθηκε. Επίσης εγγράφει στο αρχείο log την αξία που προστέθηκε και τη νέα συνολική αξία εισιτηρίων που είναι αποθηκευμένη στην κάρτα. Τέλος επιστρέφει στο σημείο επιλογής λειτουργίας.

5.3.3.3 Πρόγραμμα "Μετακίνηση"

Το πρόγραμμα αυτό έχει τη δομή που φαίνεται στο ακόλουθο σχήμα:



Σχήμα 5.3.3.3 - Δομή Προγράμματος «Μετακίνηση»

Στη σχεδίαση αυτή έχει παραλειφθεί το επίπεδο Α στο οποίο το πρόγραμμα ξεκινάει, ανοίγει κανάλι επικοινωνίας με τον reader και συνεχίζει στο επίπεδο Β. Αυτό έγινε γιατί θεωρείται ότι το πρόγραμμα θα περάσει μία φορά από εκεί και δε θα ξαναγυρίσει (εκτός αν προκύψει κάποιο πρόβλημα) αφού χρειάζεται να τρέχει συνέχεια στα μέσα μαζικής μεταφοράς σε κατάσταση αναμονής εισόδου κάρτας.

- Επίπεδο Β:
  Στο σημείο αυτό ο reader αναμένει την είσοδο μίας κάρτας. Όταν αυτό γίνει, το πρόγραμμα ανοίγει μία σύνοδο (session) με την κάρτα και στη συνέχεια διαβάζει από την κάρτα το είδος του εισιτηρίου που θα αφαιρέσει (μειωμένο / κανονικό). Τέλος υπολογίζει την αντίστοιχη αξία που θα αφαιρέσει από την κάρτα και επιχειρεί την αφαίρεση αυτή. Αν η κάρτα δεν έχει το απαραίτητο ποσό του εισιτηρίου, πληροφορεί τον κάτοχό της, τον προτρέπει να την αφαιρέσει από τον reader και περνάει ξανά στο επίπεδο αναμονής. Αλλιώς, μετά την αφαίρεση αξίας, γράφει σε ένα αρχείο τη συναλλαγή (προαιρετικό βήμα) και περνάει πάλι στο επίπεδο αναμονής κάρτας.

5.4 Γενικότερο Πλαίσιο Σχεδίασης

Μέχρι τώρα έχουμε δει τις προδιαγραφές και ανάγκες σχεδίασης για κάθε μία από τις 3 εφαρμογές της διπλωματικής αυτής. Όπως έχει ειπωθεί και οι 3 υλοποιούνται πάνω στην ίδια κάρτα, η κάθε μία με δικές της παραμέτρους. Επίσης έχουμε εξετάσει τα προγράμματα που αντιστοιχούν σε κάθε εφαρμογή και τα οποία χειρίζονται και επεξεργάζονται τα δεδομένα της εφαρμογής αυτής στην κάρτα.
Υπάρχει όμως και το θέμα της δημιουργίας και αρχικοποίησης όλων αυτών των αρχείων στην κάρτα, το οποίο δεν γίνεται με κανένα από όλα τα προγράμματα που αναφέραμε και το οποίο είναι το πρώτο στάδιο που πρέπει να εκτελεσθεί. Γι' αυτό το σκοπό, έχουν δημιουργηθεί δύο ακόμα προγράμματα, το πρόγραμμα "Αρχικοποίηση" και το πρόγραμμα "Προσωποποίηση".

5.4.1 Πρόγραμμα "Αρχικοποίηση"

Το πρόγραμμα αυτό ευθύνεται για την δημιουργία όλων των απαραίτητων αρχείων στην κάρτα και την παραμετροποίηση κάποιων από αυτών για λόγους ασφαλείας. Θεωρείται ότι αυτό το πρόγραμμα εκτελείται κάτω από συνθήκες ασφαλείας και από εξουσιοδοτημένο χρήστη, τον διαχειριστή θησαυροφυλακίου όπως θα καλείται από εδώ και πέρα. Ο διαχειριστής αυτός ευθύνεται επίσης για τη δημιουργία στην κάρτα των μυστικών κλειδιών του ιδίου και των άλλων τριών διαχειριστών (διαχειριστής μισθοδοσίας, διαχειριστής πληρωμών, διαχειριστής εισιτηρίων). Όπως θα δούμε και στο επόμενο κεφάλαιο, ο διαχειριστής θησαυροφυλακίου έχει τη μεγαλύτερη ευθύνη και αρμοδιότητα όσον αφορά τις κάρτες που δημιουργούνται. Δεύτερος ιεραρχικά έρχεται ο διαχειριστής μισθοδοσίας ο οποίος είναι αυτός που εκτελεί και το πρόγραμμα "Προσωποποίηση".
Τα μυστικά κλειδιά των διαχειριστών που αποθηκεύονται κρυπτογραφημένα στην κάρτα δημιουργούνται με δύο στάδια το καθένα. Χρειάζεται η γνώση δηλαδή δύο κωδικών για τη δημιουργία του κάθε κλειδιού. Ο πρώτος κωδικός (ένας για κάθε εφαρμογή) είναι ειδικός κωδικός της εταιρίας που θεωρείται ότι παρέχει τις κάρτες και τον γνωρίζει μόνο ο διαχειριστής θησαυροφυλακίου ο οποίος θεωρείται εξουσιοδοτημένο όργανο της εταιρίας. Ο δεύτερος κωδικός για κάθε κλειδί είναι και αυτός που αποστέλλεται στον εκάστοτε διαχειριστή και ζητείται στην εκκίνηση των προγραμμάτων της εφαρμογής που κάθε ένας από αυτούς χειρίζεται. Δικαίωμα για ολοκληρωτική διαγραφή των αρχείων της κάρτας και επαναρχικοποίησή της έχει μόνο ο διαχειριστής θησαυροφυλακίου.
Κατά την εκκίνηση του προγράμματος, ο διαχειριστής θησαυροφυλακίου καλείται να διαλέξει αν επιθυμεί αρχικοποίηση μίας κάρτας ή διαγραφή του αρχείου συστήματος που συνεπάγεται και διαγραφή όλων των αρχείων της κάρτας. Αν επιλέξει το πρώτο, η παρέμβασή του ζητείται μόνο για τη δημιουργία των κλειδιών των διαχειριστών. Όλες οι άλλες διαδικασίες δημιουργίας αρχείων και παραμετροποίησης κάποιων από αυτών γίνονται αυτόματα. Αν πάλι επιλεγεί η διαγραφή του αρχείου συστήματος, το πρόγραμμα ζητάει από τον διαχειριστή να εισάγει τους δύο κωδικούς που δημιουργούν το μυστικό του κλειδί και αν η επαλήθευση επιτύχει, εκτελείται η διαγραφή αυτόματα.
Μετά τη διαδικασία αρχικοποίησης των καρτών που θεωρείται ότι γίνεται μαζικά, οι κάρτες στέλνονται στον διαχειριστή μισθοδοσίας ενώ ταυτόχρονα αποστέλλονται στους διαχειριστές και οι κωδικοί τους.

5.4.2 Πρόγραμμα "Προσωποποίηση"

Με αυτό το πρόγραμμα γίνεται η προσωποποίηση μίας κάρτας ανά κάτοχο και η αρχικοποίηση ειδικών αρχείων των εφαρμογών. Το πρόγραμμα "Προσωποποίηση" ζητάει στην εκκίνησή του τον κωδικό του διαχειριστή μισθοδοσίας, ο οποίος όπως αναφέρθηκε του έχει αποσταλεί από την εταιρία που παρέχει τις κάρτες.
Όταν ο κωδικός επαληθευτεί από την κάρτα, το πρόγραμμα ζητάει από τον διαχειριστή μισθοδοσίας να δημιουργήσει δύο νέους μυστικούς κωδικούς, έναν δικό του για να μπορεί να συμπληρώσει χειροκίνητα τις ώρες εργασίας ενός εργαζόμενου στο πρόγραμμα "Υπερωρίες"

(σημείο ελέγχου ΣΤ του σχήματος 5.1.3.1), και έναν για τον κάτοχο της κάρτας. Ο κωδικός του κατόχου μπορεί να δημιουργείται από ειδική διαδικασία ή να συμπληρώνεται από τον ίδιο τον κάτοχο αν είναι παρών στην προσωποποίηση της κάρτας.
Έπειτα το πρόγραμμα εκτελεί αυτόματα τις διαδικασίες ρύθμισης περιεχομένου και προστασίας αρχείων που αφορούν τις εφαρμογές όπως αρχεία μετρητών, κανόνων, εγγραφών. Η παρέμβαση του χρήστη ζητείται μία φορά ακόμα στο τέλος του προγράμματος, για την εγγραφή σε ειδικό αρχείο των προσωπικών στοιχείων του κατόχου (και στοιχεία μισθοδοσίας). Τα στοιχεία αυτά που καλείται ο διαχειριστής να συμπληρώσει για τον κάτοχο είναι τα ακόλουθα:

- Όνομα κατόχου (μέχρι 20 χαρακτήρες)
- Επίθετο κατόχου (μέχρι 20 χαρακτήρες)
- Όνομα Πατρός κατόχου (μέχρι 20 χαρακτήρες)
- Ημερομηνία Γέννησης
- Αριθμός Ταυτότητας
- Διεύθυνση (αποτελείται από Οδό, Αριθμό, Περιοχή και Πόλη διαμονής)
- Ωρομίσθιο κατόχου (ως εργαζόμενος)
- Ημέρες Αδείας που δικαιούται σε ένα έτος
- Όριο Υπερωριών που μπορεί να κάνει σε ένα μήνα
- Χρόνια Εργασίας στο συγκεκριμένο χώρο

Όταν πλέον συμπληρωθούν αυτά τα στοιχεία, το πρόγραμμα ειδοποιεί τον διαχειριστή ότι ολοκληρώθηκε η διαδικασία προσωποποίησης. Αν έχει συμβεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος, ο διαχειριστής ενημερώνεται για το γεγονός αυτό από το πρόγραμμα το οποίο έπειτα τερματίζει. Σε αυτή την περίπτωση ο διαχειριστής πρέπει να προσωποποιήσει νέα κάρτα για τον κάτοχο ο οποίος θα έπαιρνε την παλιά και η παλιά κάρτα να αποσταλεί στον διαχειριστή θησαυροφυλακίου, ο οποίος θα διαγράψει το αρχείο συστήματος και θα την αρχικοποιήσει εκ νέου, θέτοντάς την πάλι ενεργή.
Η ίδια διαδικασία (επανενεργοποίησης μίας κάρτας) θα ακολουθηθεί και στην περίπτωση που μία κάρτα μπλοκάρει λόγω συνεχόμενων αποτυχημένων προσπαθειών επαλήθευσης ενός κωδικού ή κλειδιού (εκ μέρους του κατόχου ή και των διαχειριστών). Έτσι αυτές οι κάρτες αποστέλλονται πίσω στον διαχειριστή θησαυροφυλακίου για να τις θέσει πάλι σε χρήση. Η μόνη περίπτωση μία κάρτα να αχρηστευθεί εντελώς είναι αυτή της εμπλοκής του κλειδιού του διαχειριστή θησαυροφυλακίου, η οποία όμως θεωρείται σπάνια.

6

Υλοποίηση Εφαρμογών

Στο κεφάλαιο αυτό θα συζητηθεί το περιεχόμενο κάθε κάρτας που έχει σχεδιασθεί για τις 3 εφαρμογές που αναλύσαμε στο προηγούμενο κεφάλαιο, όπως επίσης το περιεχόμενο των προγραμμάτων και τα εργαλεία υλοποίησης των εφαρμογών.

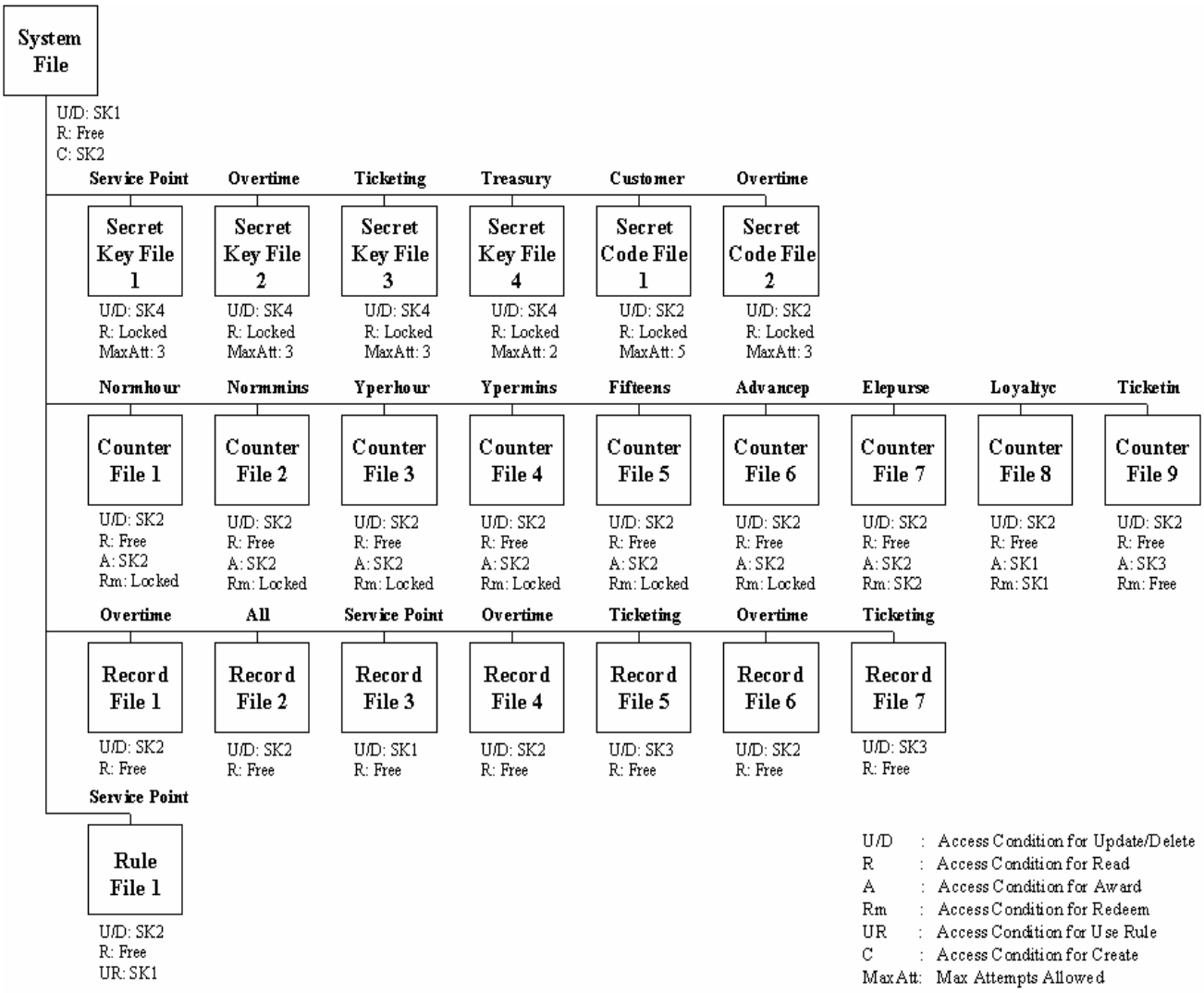
6.1 Εργαλεία Υλοποίησης

Όπως αναφέρθηκε και στο Κεφάλαιο 4, για την υλοποίηση των εφαρμογών χρησιμοποιήσαμε τις έξυπνες κάρτες GemClub και τον reader GemPC410 της GEMPLUS. Επίσης χρησιμοποιήθηκε η διεπαφή GEMPLUS API 4.30 για την επικοινωνία μεταξύ εφαρμογών και του driver του reader και η βιβλιοθήκη διεπαφής GemClub (GemClub Interface Library) η οποία παρέχει το σύνολο εντολών της GemClub μέσω του API 4.30.
Τα προγράμματα που απαρτίζουν τις εφαρμογές έχουν γραφεί στη γλώσσα προγραμματισμού C και τρέχουν σε περιβάλλον Windows ως Console Applications. Δε χρησιμοποιήθηκε κάποια Object Oriented γλώσσα προγραμματισμού για το λόγο ότι με την παρούσα μορφή τους οι εφαρμογές είναι δυνατόν με συγκεκριμένες αλλαγές να μεταφερθούν σε ένα embedded σύστημα, γεγονός που ενδιαφέρει το εργαστήριο.
Κάθε πρόγραμμα περιλαμβάνει κάποια header αρχεία και κάποιες libraries τα οποία είναι απαραίτητα για τις λειτουργίες της βιβλιοθήκης διεπαφής GemClub και της βιβλιοθήκης επαφής των GemCore-Based Readers.
Συγκεκριμένα τα header αρχεία που χρησιμοποιούνται είναι το wxxstar4.h (διεπαφή εντολών GemClub), gemplus.h (ορισμοί τύπων δεδομένων όπως "BYTE" και "G_FAR", ορισμοί μακροεντολών για τη C, ορισμοί κωδικών λαθών της Gemplus), το gemgcr.h (ορισμοί δομών, ορισμοί συναρτήσεων του API 4.30, προκαθορισμένες τιμές), το gemrdr.h (προκαθορισμένες τιμές για τους Gemplus readers), gemcard.h (προκαθορισμένες τιμές για τις κάρτες που χρησιμοποιούνται με τους Gemplus readers). Επίσης χρησιμοποιούνται τα header αρχεία des.h (περιέχει τον ορισμό της συνάρτησης crypto για την εκτέλεση του κρυπτογραφικού αλγορίθμου 3DES) και το αρχείο nasia.h (περιέχει συναρτήσεις που είναι κοινές στα προγράμματα καθώς και προκαθορισμένες τιμές που θα εξετάσουμε αργότερα).
Τέλος οι libraries που περιλαμβάνονται στις εφαρμογές είναι η w32star4.lib (βιβλιοθήκη GemClub) και η w32gcr40.lib (βιβλιοθήκη GCR400) ενώ απαραίτητο για την εγκατάσταση των βιβλιοθηκών είναι και το αρχείο w32star4.dll.

6.2 Αρχεία Κάρτας Εφαρμογών

Πριν γίνει ανάλυση των περιεχομένων της κάρτας που χρησιμοποιείται για την υλοποίηση των 3 εφαρμογών, θα γίνει μία αντιστοίχιση των ονομάτων των προγραμμάτων τα οποία αναφέρθηκαν στο Κεφάλαιο 5 με αυτά που χρησιμοποιήθηκαν στην υλοποίηση των εφαρμογών.
Έτσι, η εφαρμογή "Μισθοδοσία" καλείται "Overtime Application", ενώ τα προγράμματα "Υπερωρίες" και "Εξαργύρωση" που την αποτελούν, καλούνται αντίστοιχα "Overtime" και "Redemption".

Το πρόγραμμα "Σημείο Εξυπηρέτησης" της εφαρμογής "Ηλεκτρονικό Πορτοφόλι - Πρόγραμμα Εμπιστοσύνης" καλείται "ServPoint". Το άλλο κομμάτι της εφαρμογής είναι όπως έχει ειπωθεί μέρος του προγράμματος "Εξαργύρωση" (Redemption).
Η εφαρμογή Εισιτηρίων καλείται "Ticketing" και αποτελείται από το "TicketCenter" ("Κέντρο Εισιτηρίων") και το πρόγραμμα "Transportation" (πρόγραμμα "Μετακίνηση").
Τέλος τα προγράμματα "Αρχικοποίηση" και "Προσωποποίηση" που αναφέραμε στην ενότητα 5.4 καλούνται αντίστοιχα "Inistep" και "Personover".
Στο Σχήμα 6.2 που φαίνεται παρακάτω (περιέχεται και στο Παράρτημα Γ σε μεγαλύτερο μέγεθος) φαίνονται συνοπτικά τα αρχεία από τα οποία αποτελείται η κάρτα που σχεδιάστηκε για τις 3 εφαρμογές της διπλωματικής αυτής.
Κάτω από κάθε αρχείο παρατίθενται περιληπτικά οι ιδιότητές του (κυρίως τα δικαιώματα πρόσβασης) ενώ πάνω από το αρχείο υπάρχει το όνομα της εφαρμογής που χρησιμοποιεί το αρχείο αυτό ή στην περίπτωση των αρχείων μετρητών, η ετικέτα κάθε μετρητή.



Σχήμα 6.2 - Αρχεία κάρτας

Τα περιεχόμενα των αρχείων αυτών μαζί με σημαντικά στοιχεία θα αναλύσουμε ακολούθως.

6.2.1 System File

Το αρχείο συστήματος έχει πάντα SFI 01h και είναι το πρώτο που δημιουργείται στην κάρτα. Διαγραφή ή ανανέωση του αρχείου αυτού γίνεται μόνο με Secure Messaging (SM) με τη χρήση του μυστικού κλειδιού του διαχειριστή θησαυροφυλακίου, δηλαδή του Secret Key 4 (SK4). Η ανάγνωση του αρχείου είναι ελεύθερη ενώ έχει οριστεί ότι δημιουργία αρχείων μπορεί να γίνει μόνο με Secure Messaging (SM) με τη χρήση του μυστικού κλειδιού του διαχειριστή μισθοδοσίας. Το κλειδί αυτό είναι το Secret Key 2 (SK2). Πρώτα όμως δημιουργούνται όλα τα αρχεία στην κάρτα και μετά ρυθμίζονται τα AC (Access Conditions) του system file.

6.2.2 Secret Key Files

Στην κάρτα δημιουργούνται 4 αρχεία μυστικών κλειδιών. Το πρώτο είναι το SK1 και ανήκει στον διαχειριστή πωλήσεων. Το δεύτερο είναι το SK2 και ανήκει στον διαχειριστή μισθοδοσίας. Το τρίτο μυστικό κλειδί ανήκει στον διαχειριστή εισιτηρίων και είναι το SK3. Τελευταίο και πιο σημαντικό είναι το SK4 που ανήκει στον διαχ/στή θησαυροφυλακίου και το οποίο χρησιμοποιείται για τις λειτουργίες αυτές που χρειάζονται μέγιστη ασφάλεια, όπως η διαγραφή του αρχείου συστήματος. Κάθε ένα από αυτά τα αρχεία μυστικών κλειδιών μπορούν να ανανεωθούν ή να διαγραφούν μόνο με τη χρήση SM (Secure Messaging) με τη χρήση του SK4 και είναι κλειδωμένα για ανάγνωση. Ο μέγιστος αριθμός συνεχόμενων προσπαθειών επαλήθευσης για τα κλειδιά SK1, SK2 και SK3 είναι το 3, ενώ στην περίπτωση του SK4 ο αριθμός είναι το 2 (για λόγους ασφαλείας προφανώς).
Η διαδικασία δημιουργίας ενός μυστικού κλειδιού για αυτή τη κάρτα (όπως έχει αναφερθεί στην ενότητα 5.4.1) εκτελείται από το πρόγραμμα "Inistep" και συνίσταται σε 2 βήματα. Αρχικά ζητείται από το χρήστη του προγράμματος (διαχειριστής θησαυροφυλακίου) να εισάγει τον κωδικό των 8bits του διαχειριστή στον οποίο θα αντιστοιχεί το κλειδί. Τον κωδικό αυτό ονομάζουμε Pin του εκάστοτε διαχειριστή και είναι αυτός που του αποστέλλεται. Έπειτα ζητείται από τον χρήστη να εισάγει τον κωδικό (8bits) που έχει δημιουργήσει η εταιρία (company key) για τον κάθε διαχειριστή. Οι δύο αυτοί κωδικοί χρησιμοποιούνται για να παράγουν με τον αλγόριθμο 3DES το κρυπτογραφημένο κλειδί που αποθηκεύεται στην κάρτα. Τα 3 company keys αποθηκεύονται στο header αρχείο nasia.h και χρησιμοποιούνται για τη δημιουργία των exe αρχείων των προγραμμάτων των εφαρμογών. Το company key που χρησιμοποιείται για τη δημιουργία του SK4 δεν αποθηκεύεται πουθενά και αν τυχόν χρειαστεί ο διαχειριστής θησαυροφυλακίου να το χρησιμοποιήσει για κάποια λειτουργία πρέπει να το εισάγει μαζί με το Pin του.

6.2.3 Secret Code Files

Δύο αρχεία μυστικών κωδικών υπάρχουν στην κάρτα και οι ιδιότητές τους ορίζονται από τον διαχειριστή μισθοδοσίας κατά τη διαδικασία προσωποποίησης της κάρτας. Το πρώτο είναι το Secret Code 1 (SC1) και αντιστοιχεί στον κωδικό του κατόχου της κάρτας (8bits). Ο μέγιστος αριθμός συνεχόμενων προσπαθειών επαλήθευσης για τον κωδικό αυτό ορίζεται επίσης από τον διαχειριστή. Στις sample cards που έχουν δημιουργηθεί για τον έλεγχο της σωστής λειτουργίας των εφαρμογών της διπλωματικής, ο αριθμός αυτός έχει οριστεί να είναι το 5. Διαγραφή ή ανανέωση των στοιχείων του αρχείου του SC1 μπορεί να γίνει μόνο με SM με τη χρήση του SK2.
Ο δεύτερος κωδικός προορίζεται για τον διαχειριστή μισθοδοσίας για την περίπτωση που πρέπει να συμπληρώσει χειροκίνητα τις ώρες εργασίας ενός εργαζόμενου στα πλαίσια του προγράμματος "Overtime" ("Υπερωρίες"). Τη τιμή του ορίζει ο ίδιος ο διαχειριστής και ο μέγιστος αριθμός συνεχόμενων προσπαθειών επαλήθευσής του στις sample cards έχει οριστεί στις 3. Διαγραφή ή ανανέωση των στοιχείων του αρχείου του SC2 μπορεί πάλι να γίνει με SM με τη χρήση του SK2.

6.2.4 Counter Files

Τα αρχεία μετρητών είναι από τα πλέον σημαντικά στην υλοποίηση οποιασδήποτε εφαρμογής. Στην περίπτωσή μας, έχουν δημιουργηθεί στην κάρτα 9 μετρητές για διαφορετικές χρήσεις. Οι ιδιότητές τους ρυθμίζονται από τον διαχειριστή μισθοδοσίας.

- Counter 1: ο πρώτος μετρητής καλείται Normhour και χρησιμοποιείται από την εφαρμογή μισθοδοσίας για την αποθήκευση του συνόλου των ωρών κανονικής εργασίας ενός εργαζόμενου μέσα σε ένα μήνα. Για κάθε 1 ώρα κανονικής εργασίας προστίθενται στον μετρητή 10 πόντοι.
- Counter 2: ο δεύτερος μετρητής ονομάζεται Normmins και χρησιμοποιείται επίσης από την εφαρμογή μισθοδοσίας για την αποθήκευση του συνόλου των λεπτών που συμπληρώνουν την κανονική εργασία ενός εργαζόμενου μέσα σε ένα μήνα. Για κάθε λεπτό κανονικής εργασίας προστίθενται 10 πόντοι στο μετρητή αυτό. Ο μετρητής Normmins είναι συμπληρωματικός του Normhour και μαζί οι τιμές τους χρησιμοποιούνται για να βρεθεί το σύνολο των ωρών κανονικής εργασίας του εργαζόμενου μέσα στο μήνα.
  Για να γίνει αντιληπτή η χρήση τους, θα χρησιμοποιήσουμε ένα παράδειγμα. Έστω λοιπόν ότι την $1^{η}$ μέρα εργασίας για το νέο μήνα, ο εργαζόμενος κάνει 6 ώρες και 40 λεπτά κανονικής εργασίας και φεύγει. Στους μετρητές Normhour και Normmins αποθηκεύονται 60 και 400 πόντοι αντίστοιχα. Τη $2^{η}$ μέρα ο εργαζόμενος δουλεύει 7 ώρες και 15 λεπτά (κανονική εργασία) και αποχωρεί. Στους μετρητές Normhour και Normmins προστίθενται 70 και 150 πόντοι αντίστοιχα. Τέλος, την $3^{η}$ μέρα ο εργαζόμενος εργάζεται 9 ώρες και 20 λεπτά. Προφανώς τη μέρα αυτή έχει κάνει 1 ώρα και 20 λεπτά υπερωρία, ενώ η κανονική του εργασία ανέρχεται στις 8 ώρες. Έτσι 80 πόντοι προστίθενται στον μετρητή Normhour και 0 στον μετρητή Normmins. Στο τέλος της $3^{ης}$ μέρας λοιπόν οι δύο αυτοί μετρητές έχουν τιμές:
  Normhour = 60 + 70 + 80 = 210 $\Rightarrow$ 21 ώρες,
  Normmins = 400 + 150 + 0 = 550 $\Rightarrow$ 55 λεπτά.
  Δηλαδή μετά από 3 μέρες ο εργαζόμενος έχει συμπληρώσει 21 ώρες και 55 λεπτά κανονικής εργασίας.
- Counter 3: ο επόμενος μετρητής είναι ο Yperhour και χρησιμοποιείται μόνο από την εφαρμογή της μισθοδοσίας για να αποθηκεύει τις ώρες υπερωριακής εργασίας του εργαζόμενου μέσα σε ένα μήνα. Για κάθε 1 ώρα υπερωριακής εργασίας προστίθενται στον μετρητή 15 πόντοι. Αυτό συμβαίνει γιατί οι ώρες υπερωριακής εργασίας αμείβονται με προσαύξηση 50% επί του ωρομισθίου και έτσι 1 ώρα εργασίας με ωρομίσθιο X€/ώρα αμείβεται με:
  $$1_{ΩΡΑ} \bullet X_{€/ΩΡΑ} + 1_{ΩΡΑ} \bullet 0{,}5 \bullet X_{€/ΩΡΑ} = 1_{ΩΡΑ} \bullet 1{,}5 \bullet X_{€/ΩΡΑ} = (1{,}5_{ΩΡΕΣ} \bullet X_{€/ΩΡΑ})€.$$
  Δηλαδή η 1 ώρα υπερωριακής εργασίας υπολογίζεται ως 1,5 ώρα (κατά τον υπολογισμό του μισθού), γι' αυτό και προστίθενται 15 πόντοι στο μετρητή.
- Counter 4: ο μετρητής Ypermins είναι ο συμπληρωματικός για τις υπερωρίες μετρητής του Yperhour. Σε αυτόν αποθηκεύονται λεπτά υπερωριακής εργασίας. Επειδή σύμφωνα με την πολιτική υπερωριών που αναφέραμε στην ενότητα 5.1.3.3 τα πρώτα 15 λεπτά υπερωρίας αμείβονται με προσαύξηση 30% επί του ωρομισθίου ενώ τα επόμενα αμείβονται με προσαύξηση 50% επί του ωρομισθίου, για κάθε λεπτό υπερωρίας μέχρι τα πρώτα 15 προστίθενται στον μετρητή 13 πόντοι, ενώ για κάθε ένα από τα επόμενα προστίθενται 15 πόντοι. Η λογική αντιστοίχισης των πόντων εξηγήθηκε πριν.
- Counter 5: ο μετρητής Fifteens που ανήκει επίσης στην εφαρμογή της μισθοδοσίας, χρησιμοποιείται για την αποθήκευση των λεπτών υπερωρίας που αμείβονται με προσαύξηση 30% επί του ωρομισθίου. Για κάθε τέτοιο λεπτό αποθηκεύεται ένας πόντος στον μετρητή.

Διαγραφή ή ανανέωση των αρχείων των πέντε παραπάνω μετρητών ή απόδοση πόντων σε αυτούς μπορεί να γίνει μόνο με SM με τη χρήση του SK2. Ανάγνωση του αριθμού των πόντων που είναι αποθηκευμένοι στους μετρητές αυτούς γίνεται ελεύθερα ενώ αφαίρεση πόντων δεν επιτρέπεται. Όταν ο εργαζόμενος εξαργυρώνει τις ώρες εργασίας του, γίνεται ανανέωση των πόντων των μετρητών στη τιμή 0 για να χρησιμοποιηθούν για το νέο μήνα.

- Counter 6: ο μετρητής Advancep χρησιμοποιείται από την εφαρμογή μισθοδοσίας και συγκεκριμένα από το πρόγραμμα "Εξαργύρωση" για να αποθηκεύει την εκάστοτε προκαταβολή που παίρνει ο κάτοχος της κάρτας είτε αυτή είναι κανονική προκαταβολή είτε υπερωριακή (ενότητα 5.1.3.4). Τα ποσά που αποθηκεύονται στο μετρητή αυτό είναι δεκαδικής μορφής με δύο δεκαδικά ψηφία και πριν αποθηκευτούν πολλαπλασιάζονται με το 100. Έτσι το ποσό των 123,4€ θα αποθηκευτεί ως 12340 πόντοι και το ποσό των 25,31€ θα αποθηκευτεί ως 2531 πόντοι. Ο μετρητής αυτός προστατεύεται με Secure Messaging με το κλειδί SK2 από τις διαδικασίες ανανέωσης ή διαγραφής του καθώς και από την διαδικασία απόδοσης πόντων. Η ανάγνωσή του γίνεται ελεύθερα ενώ αφαίρεση πόντων δεν επιτρέπεται.
  Μετά την πληρωμή του εργαζόμενου ο μετρητής αυτός μηδενίζεται μόνο αν το ορίσει το πρόγραμμα "Εξαργύρωση". Συγκεκριμένα, αν μέσα στο μήνα ο εργαζόμενος έχει πάρει κάποια κανονική προκαταβολή, αυτή μέχρι τη στιγμή της πληρωμής του είναι αποθηκευμένη στο μετρητή και μετά την πληρωμή πρέπει να αφαιρεθεί. Αν πάλι ο εργαζόμενος κατά την πληρωμή του πάρει κάποια υπερωριακή προκαταβολή (λόγω υπέρβασης του ορίου υπερωριών για ένα μήνα), αυτή αποθηκεύεται στον μετρητή Advancep και πρέπει να είναι γνωστή τον επόμενο μήνα στο σύστημα. Έτσι σε αυτή την περίπτωση ο μετρητής δεν μπορεί να μηδενισθεί μετά την πληρωμή του εργαζόμενου αλλά πρέπει να διατηρεί τη τιμή του.
  Διαγραφή ή ανανέωση του μετρητή, καθώς και απόδοση πόντων σε αυτόν μπορεί να γίνει μόνο με SM με τη χρήση του SK2. Ανάγνωση του αριθμού των πόντων του μετρητή γίνεται ελεύθερα ενώ αφαίρεση πόντων δεν επιτρέπεται.
- Counter 7: ο Elepurse είναι ο μετρητής που υλοποιεί το ηλεκτρονικό πορτοφόλι και χρησιμοποιείται από το πρόγραμμα "Εξαργύρωση" και την εφαρμογή Ηλεκτρονικό Πορτοφόλι – Πρόγραμμα Εμπιστοσύνης. Αποθηκεύει αξία και όπως και ο προηγούμενος μετρητής για να υποστηρίξει ποσά με δύο δεκαδικά ψηφία, τα πολλαπλασιάζει επί 100 πριν την αποθήκευσή τους. Έτσι τα 50€ αποθηκεύονται στον μετρητή ως 5000 πόντοι, ενώ τα 23,54€ αποθηκεύονται ως 2354 πόντοι.
  Διαγραφή ή ανανέωση του μετρητή, απόδοση ή αφαίρεση πόντων από αυτόν μπορούν να γίνουν μόνο με SM με τη χρήση του SK2. Ανάγνωση του αριθμού των πόντων του μετρητή γίνεται ελεύθερα. Η αφαίρεση πόντων γίνεται και έμμεσα με τη χρήση του κανόνα 1 που θα αναλύσουμε παρακάτω.
- Counter 8: ο μετρητής Loyaltyc είναι ο μετρητής που χρησιμοποιείται από την εφαρμογή Ηλεκτρονικό Πορτοφόλι - Πρόγραμμα Εμπιστοσύνης για να αποθηκεύει τους πόντους που έχει κερδίσει ο εργαζόμενος με τις αγορές που έχει κάνει. Αυτός ο μετρητής είναι προσβάσιμος κυρίως με τη χρήση του κανόνα 1. Ανανέωση ή διαγραφή του αρχείου του μετρητή αυτού μπορεί να γίνει μόνο με SM με τη χρήση του SK2 ενώ η ανάγνωση του περιεχομένου του γίνεται ελεύθερα. Οποιαδήποτε όμως πράξη απόδοσης πόντων στον μετρητή ή αφαίρεση πόντων από αυτόν γίνεται μόνο με SM με τη χρήση του κλειδιού του διαχειριστή πωλήσεων SK1. Το πρόγραμμα "Εξαργύρωση" προσφέρει ως λειτουργία μόνο την ανάγνωση των πόντων αυτού του μετρητή.
- Counter 9: ο τελευταίος μετρητής καλείται Ticketin και χρησιμοποιείται από την εφαρμογή "Εισιτήρια" για την αποθήκευση αξίας εισιτηρίων. Όπως και οι άλλοι μετρητές οι οποίοι αποθηκεύουν αξία (Elepurse, Advancep) πολλαπλασιάζουν την αξία (σε €) επί 100. Ανανέωση ή διαγραφή του αρχείου του Ticketin μετρητή μπορεί να γίνει μόνο με SM με τη χρήση του SK2 ενώ η ανάγνωση του περιεχομένου του γίνεται ελεύθερα. Απόδοση πόντων στον μετρητή (πρόσθεση αξίας) γίνεται μόνο με

SM με τη χρήση του κλειδιού SK3 του διαχειριστή εισιτηρίων ενώ η αφαίρεση πόντων δεν δεσμεύεται από κάποια συνθήκη πρόσβασης.

6.2.5 Record Files

Στις κάρτες που σχεδιάστηκαν, υπάρχουν 7 αρχεία εγγραφών (record files). Μόνο τα δύο από αυτά προσωποποιούνται από το πρόγραμμα "Προσωποποίηση" ενώ τα υπόλοιπα ορίζονται στο πρόγραμμα "Αρχικοποίηση". Τα αρχεία αυτά αναλύονται ακολούθως.

- Record File 1: το αρχείο αυτό χρησιμοποιείται αποκλειστικά από το πρόγραμμα "Υπερωρίες" της εφαρμογής μισθοδοσίας. Έχει μήκος 24 bytes και αποτελείται από ένα record μήκους 12 byte που αντιστοιχίζει τα bytes αυτά με μεταβλητές του προγράμματος. Διαγραφή ή ανανέωση του αρχείου μπορεί να γίνει μόνο με SM με τη χρήση του SK2 ενώ η ανάγνωση του αρχείου γίνεται ελεύθερα.
  Το πρώτο byte ($byte_0$ μετρώντας από τα αριστερά προς τα δεξιά) αντιστοιχίζεται στη μεταβλητή "dayin" του προγράμματος η οποία δείχνει αν πρόκειται για είσοδο ή έξοδο του εργαζόμενου από το χώρο εργασίας. Το byte έχει τη τιμή 0 όταν έχει προηγηθεί έξοδος του χρήστη και πληροφορεί την κάρτα ότι εκείνη τη στιγμή που το πρόγραμμα διαβάζει το record, ο εργαζόμενος πραγματοποιεί είσοδο στο χώρο εργασίας. Αντίστοιχα το $byte_0$ έχει τη τιμή 1 όταν έχει προηγηθεί είσοδος του εργαζόμενου στο χώρο και αντιστοιχίζεται σε έξοδο του εργαζόμενου από το χώρο εργασίας. Το $byte_0$ (και αντίστοιχα η μεταβλητή dayin) μπορεί να πάρει μόνο αυτές τις δύο τιμές (0 και 1) και η default τιμή (τιμή αρχικοποίησης) είναι το 0 (για να αντιστοιχηθεί η πρώτη επόμενη εισαγωγή της κάρτας με είσοδο του κατόχου).
  Το δεύτερο byte ($byte_1$) αντιστοιχίζεται στη μεταβλητή "weekday" του προγράμματος και δείχνει ποια μέρα της εβδομάδας έχει κάνει τελευταία είσοδο ο εργαζόμενος. Η τιμή του αλλάζει κάθε φορά που κάνει πρώτη είσοδο ο χρήστης μέσα σε μία ημέρα. Μέσα στο πρόγραμμα παίρνει τιμές από το 0 ως το 6, δείχνοντας πόσες μέρες έχουν περάσει από την Κυριακή. Έτσι το 0 σημαίνει ημέρα Κυριακή και το 6 σημαίνει Σάββατο. Χρησιμεύει για να επισημάνει αν ο εργαζόμενος δουλεύει Σάββατο ή Κυριακή για να χρεωθεί σωστά τις υπερωρίες του. Η τιμή αρχικοποίησής του είναι το 7 για να μη μπερδέψει το πρόγραμμα στην πρώτη ανάγνωσή του.
  Τα δύο επόμενα bytes ($byte_2$ και $byte_3$) αντιστοιχίζονται στη μεταβλητή "yearday" του προγράμματος και δείχνουν ποια μέρα του χρόνου έχει κάνει τελευταία είσοδο ο εργαζόμενος. Η μεταβλητή yearday παίρνει τιμές από το 0 ως το 365 μετρώντας πόσες ημέρες έχουν περάσει από την $1^η$ Ιανουαρίου. Η τιμή της μεταβλητής αυτής (και των $byte_2$ και $byte_3$ που την αποθηκεύουν στη κάρτα) αλλάζει κάθε φορά με την πρώτη είσοδο του εργαζόμενου στο χώρο εργασίας μέσα σε μία μέρα. Χρησιμοποιείται για να υποστηρίξει την ιδιότητα πολλαπλών εισόδων / εξόδων ενός εργαζόμενου μέσα σε μία μέρα. Συγκεκριμένα, χρησιμοποιείται για να ελέγξει σε κάθε είσοδο του χρήστη αν είναι η πρώτη μέσα στην ημέρα ή όχι και σε κάθε έξοδο αν είναι μέσα στην ίδια μέρα με την προηγούμενη είσοδο ή όχι. Η τιμή αρχικοποίησης των byte αυτών στην κάρτα είναι το $366_{<10>}$ δηλαδή $byte_3$ = 6Eh και $byte_2$ = 01h.
  Το $byte_4$ που ακολουθεί αντιστοιχεί στην μεταβλητή "nhour" και χρησιμοποιείται από το πρόγραμμα για να αποθηκεύει τις ώρες κανονικής εργασίας του εργαζόμενου μέσα σε μία ημέρα. Σε κάθε έξοδο του εργαζόμενου από το χώρο εργασίας μέσα σε μία ημέρα η τιμή της nhour και συνεπακόλουθα του $byte_4$ ανανεώνεται. Στην πρώτη είσοδο του εργαζόμενου η τιμή της μηδενίζεται για να αρχίσει να μετράει ώρες για τη νέα ημέρα. Αυτή η μεταβλητή είναι που υποδεικνύει πόσοι πόντοι θα αποθηκευτούν στον μετρητή Normhour σε κάθε έξοδο του εργαζόμενου. Η τιμή της δηλαδή πολλαπλασιάζεται επί 10 και προστίθεται σε κάθε έξοδο στον μετρητή. Η default τιμή της κατά την διαδικασία προσωποποίησης της κάρτας είναι το 0.

Το $byte_5$ αντιστοιχεί στην μεταβλητή "nmin" και χρησιμοποιείται από το πρόγραμμα για να αποθηκεύει τα λεπτά κανονικής εργασίας του εργαζόμενου μέσα σε μία ημέρα. Όπως και η nmin, ανανεώνεται σε κάθε έξοδο του εργαζόμενου και μηδενίζεται στην πρώτη είσοδο του εργαζόμενου σε μία ημέρα. Η τιμή της nmin πολλαπλασιασμένη επί 10 είναι ο αριθμός των πόντων που προστίθενται σε κάθε έξοδο του εργαζόμενου στον μετρητή Normmins. Η default τιμή του $byte_5$ που ορίζεται κατά την διαδικασία προσωποποίησης της κάρτας είναι το 0.
Το επόμενο byte του record είναι το $byte_6$ και αντιστοιχεί στην μεταβλητή "yhour". Χρησιμοποιείται από το πρόγραμμα "Υπερωρίες" για να αποθηκεύει τις ώρες υπερωριακής εργασίας ενός εργαζόμενου μέσα σε μία ημέρα. Η τιμή της και αντίστοιχα η τιμή του $byte_6$ το οποίο ενημερώνει, ανανεώνεται σε κάθε έξοδο του εργαζόμενου και μηδενίζεται στην πρώτη είσοδο του εργαζόμενου σε μία ημέρα. Ο μετρητής ο οποίος χρησιμοποιεί τη τιμή της για την ενημέρωση του περιεχομένου του είναι ο Yperhour. Όπως είδαμε και στην ενότητα 6.2.4, οι ώρες υπερωριακής εργασίας του εργαζόμενου πολλαπλασιάζονται επί 15 πριν προστεθούν στον μετρητή. Η τιμή αρχικοποίησής του byte αυτού είναι το 0.
Το $byte_7$ που ακολουθεί χρησιμοποιείται για να αποθηκεύει τα λεπτά υπερωριακής εργασίας του εργαζόμενου μέσα σε μία ημέρα και αντιστοιχεί στην μεταβλητή "ymin" του προγράμματος. Όπως και τα 3 προηγούμενα bytes ανανεώνει τη τιμή του σε κάθε έξοδο του εργαζόμενου από το χώρο εργασίας του και μηδενίζεται στην πρώτη είσοδό του κάθε μέρα. Η τιμή του χρησιμοποιείται από τον μετρητή Ypermins για να μετράει τα λεπτά υπερωριακής εργασίας του κατόχου της κάρτας. Σε κάθε έξοδο η τιμή του $byte_7$ πολλαπλασιάζεται με το 13 ή το 15 (η ανάλυση δίνεται στην ενότητα 6.2.4) και προστίθεται στον μετρητή. Η default τιμή του $byte_7$ που ορίζεται κατά την διαδικασία προσωποποίησης της κάρτας είναι το επίσης το 0.
Τα τελευταία 4 bytes ($byte_8$, $byte_9$, $byte_{10}$ και $byte_{11}$) χρησιμοποιούνται από την εφαρμογή για την αποθήκευση της μεταβλητής "timein" η οποία περιέχει την ημερολογιακή ώρα της πιο πρόσφατης εισόδου του εργαζόμενου στο χώρο εργασίας. Η μεταβλητή αυτή χρειάζεται 4 bytes για να παρασταθεί και έτσι με ειδική συνάρτηση σπάει από ακέραιος αριθμός σε 4 δεκαεξαδικά bytes για να αποθηκευτεί στην κάρτα ενώ άλλη ειδική συνάρτηση συνθέτει από τα 4 αυτά bytes τον ακέραιο αριθμό της ημερολογιακής ώρας για να γίνει αντιληπτός από το πρόγραμμα κατά την ανάγνωση του record αυτού. Σε κάθε έξοδο του εργαζόμενου το πρόγραμμα λαμβάνει την τρέχουσα ημερολογιακή ώρα και την συγκρίνει με την αποθηκευμένη στα bytes αυτά ημερολογιακή ώρα για να υπολογίσει τη χρονική διαφορά μεταξύ εισόδου και εξόδου του εργαζόμενου. Σε κάθε νέα είσοδο του εργαζόμενου η μεταβλητή timein και τα 4 bytes αντίστοιχα λαμβάνουν νέα τιμή. Ο ακέραιος αριθμός αρχικοποίησης των bytes αυτών κατά την προσωποποίηση της κάρτας είναι το 0.

- Record File 2: το δεύτερο αρχείο εγγραφών είναι αυτό που περιέχει τα προσωπικά στοιχεία του κατόχου της κάρτας. Έχει μέγεθος 200 bytes και αποτελείται από 13 records. Διαγραφή ή ανανέωση του αρχείου γίνεται μόνο με SM με τη χρήση του SK2 ενώ η ανάγνωση του αρχείου είναι ελεύθερη.
  Το πρώτο record έχει μέγεθος 20 bytes και αποθηκεύει το όνομα του κατόχου της κάρτας (μέχρι 20 χαρακτήρες προφανώς). Το δεύτερο και το τρίτο record έχουν επίσης μέγεθος 20 bytes το καθένα και αποθηκεύουν αντίστοιχα το επίθετο και το όνομα πατρός του κατόχου της κάρτας.
  Το record 4 έχει μήκος 6 bytes και αποθηκεύει την ημερομηνία γέννησης του κατόχου με τη μορφή HH-MM-EEEE (Η: ημέρα, Μ: μήνας, Ε: έτος). Η ημέρα και ο μήνας χρειάζονται από 1 byte ενώ το έτος χρειάζεται 2 bytes για να αποθηκευτεί. Αποθηκεύονται και οι παύλες ανάμεσα στα στοιχεία και η κάθε μία καταλαμβάνει 1 byte.
  Το επόμενο record είναι το πέμπτο και χρησιμοποιείται για την αποθήκευση του αριθμού Αστυνομικής Ταυτότητας του κατόχου της κάρτας, ο οποίος εισάγεται με τη μορφή ΓΝΝΝΝΝΝ (Γ: γράμμα, Ν: νούμερο). Έχει μέγεθος 7 bytes και αποθηκεύει το κάθε στοιχείο (γράμμα ή αριθμό) σε 1 byte.

Τα επόμενα 4 records χρησιμοποιούνται για την αποθήκευση της διεύθυνσης διαμονής του κατόχου της κάρτας. Το record 6 αποθηκεύει την οδό διαμονής και έχει μήκος 20 χαρακτήρες (bytes). Το record 7 έχει μήκος 5 bytes, περιέχει τον αριθμό της οδού και μπορεί να απεικονίσει και την απλή μορφή αριθμού οδού (π.χ. 157) καθώς και την σύνθετη (π.χ. 454 – 458). Χρησιμοποιεί 2 bytes για την αποθήκευση κάθε αριθμού και έτσι μπορεί να παραστήσει αριθμό μέχρι και το 65.535 ενώ χρησιμοποιεί 1 byte για την αποθήκευση της παύλας. Τα records 8 και 9 έχουν μέγεθος 20 bytes το καθένα και χρησιμοποιούνται για την αποθήκευση της περιοχής και της πόλης διαμονής αντίστοιχα.
Το record 10 που ακολουθεί έχει μήκος 3 bytes και είναι αυτό που αποθηκεύει το ωρομίσθιο του κατόχου – εργαζόμενου. Μπορεί να αποθηκεύσει ακέραιο αριθμό αλλά και δεκαδικό ο οποίος χρησιμοποιεί την τελεία για διαχωρισμό ακέραιου και δεκαδικού μέρους. Χρησιμοποιεί από 1 byte για το ακέραιο και το δεκαδικό μέρος και 1 byte για την τελεία.
Τα records 11 και 12 έχουν μήκος 1 byte το καθένα και αποθηκεύουν τον αριθμό των ημερών αδείας που δικαιούται ο εργαζόμενος μέσα στο χρόνο και τις ώρες υπερωριακής εργασίας που επιτρέπεται να κάνει μέσα σε ένα μήνα αντίστοιχα. Τέλος το record 13 έχει και αυτό μήκος 1 byte και αποθηκεύει τα χρόνια εργασίας του εργαζόμενου στο συγκεκριμένο τομέα. Ένα τέτοιο στοιχείο θα μπορούσε να χρησιμοποιηθεί από τον εργοδότη για τον υπολογισμό των ημερών αδείας που δικαιούται ένας εργαζόμενος καθώς και για στατιστικούς λόγους.

- Record Files 3, 4, και 5: τα αρχεία αυτά έχουν μήκος 12 bytes το καθένα και αποτελούνται από ένα record μήκους 1 byte. Χρησιμοποιούνται αποκλειστικά για την πιστοποίηση της ταυτότητας των διαχειριστών των 3 εφαρμογών. Το record file 3 χρησιμοποιείται για τον διαχειριστή πωλήσεων, το record file 4 για τον διαχειριστή μισθοδοσίας και το record file 5 για τον διαχειριστή εισιτηρίων. Η ανάγνωση κάθε αρχείου είναι ελεύθερη αλλά η ανανέωση ή διαγραφή του περιεχομένου του γίνεται μόνο με SM με τη χρήση του Secret Key του αντίστοιχου διαχειριστή (SK1 για το record file 3, SK2 για το record file 4 και SK3 για το record file 5). Κάθε record περιέχει τον χαρακτήρα «*».
  Η ταυτοποίηση ενός διαχειριστή γίνεται με τον εξής τρόπο: το πρόγραμμα ζητάει από τον διαχειριστή τον κωδικό του (αυτόν που του έχει αποσταλεί από την εταιρία που παρέχει τις κάρτες). Όταν ο εκάστοτε διαχειριστής τον εισάγει, το πρόγραμμα τον χρησιμοποιεί μαζί με το αντίστοιχο company key για τη συγκεκριμένη εφαρμογή (το οποίο είναι αποθηκευμένο στο header file "nasia.h") και παράγει με τη χρήση του αλγορίθμου 3DES ένα κρυπτόγραμμα. Αν ο κωδικός που εισάγει ο διαχειριστής είναι σωστός, το κρυπτόγραμμα που παράγεται πρέπει να είναι ταυτόσημο με το Secret Key που αντιστοιχεί στον διαχειριστή και το οποίο είναι αποθηκευμένο στην κάρτα. Απευθείας σύγκριση δεν μπορεί να γίνει οπότε χρησιμοποιείται άλλος τρόπος, πιο ασφαλής. Συγκεκριμένα, το πρόγραμμα χρησιμοποιεί το κρυπτόγραμμα για να ανανεώσει τη τιμή του record (θέτοντας πάλι το περιεχόμενό του να είναι ο χαρακτήρας «*») με Secure Messaging. Αν η προσπάθεια αποτύχει σημαίνει ότι το κρυπτόγραμμα δεν είναι ίδιο με το Secret Key που αντιστοιχεί στον διαχειριστή και συνεπακόλουθα ο κωδικός που εισήχθη δεν είναι ο σωστός.
- Record File 6: αυτό το αρχείο έχει μήκος 12 bytes και περιέχει ένα record του 1 byte. Χρησιμοποιείται από το πρόγραμμα "Εξαργύρωση" της εφαρμογής μισθοδοσίας για να υποδεικνύει στο πρόγραμμα αν στον μετρητή Advancep είναι αποθηκευμένο ποσό που έχει προέλθει από υπερωριακή προκαταβολή. Αντιστοιχεί στη μεταβλητή "flag" του προγράμματος. Όταν η τιμή του record είναι 0 σημαίνει ότι αν υπάρχει κάποιο ποσό στον μετρητή, αυτό προέρχεται από κανονική προκαταβολή. Αν πάλι η τιμή του record είναι 1, σημαίνει ότι το ποσό που είναι αποθηκευμένο στον μετρητή προέρχεται από υπερωριακή προκαταβολή (ενότητα 5.1.3.4). Η default τιμή του record αυτού κατά την αρχικοποίησή του είναι το 0. Διαγραφή ή ανανέωση του αρχείου γίνεται με SM με τη χρήση του κλειδιού SK2 ενώ η ανάγνωσή του είναι ελεύθερη.

- Record File 7: το τελευταίο αρχείο της κάρτας έχει επίσης μήκος 12 bytes και αποτελείται και αυτό από ένα record με μέγεθος 1 byte. Χρησιμοποιείται από την εφαρμογή εισιτηρίων για να δηλώσει το είδος (κανονικό / μειωμένο) των εισιτηρίων που χρησιμοποιεί ο κάτοχος της κάρτας. Μπορεί να πάρει τιμή 0 ή 1, όπου το 0 αντιστοιχεί σε κανονικό εισιτήριο ενώ το 1 αντιστοιχεί σε μειωμένο. Στα προγράμματα ανταποκρίνεται στη μεταβλητή "half". Η default τιμή του record αυτού είναι το 0. Το αρχείο αυτό μπορεί να ανανεωθεί ή να διαγραφεί μόνο με SM με τη χρήση του κλειδιού SK3 του διαχειριστή εισιτηρίων ενώ η ανάγνωσή του γίνεται ελεύθερα.

6.2.6 Rule File

Στην κάρτα υπάρχει μόνο ένα αρχείο κανόνα. Ο κανόνας (Rule 1) περιέχει δύο μακροεντολές. Η πρώτη μακροεντολή ορίζει ότι κάθε φορά που θα εκτελείται η εντολή Use Rule για κάποιο ποσό (amount) θα αφαιρούνται από τον μετρητή Elepurse (από το ηλεκτρονικό πορτοφόλι δηλαδή) τόσοι πόντοι όσο είναι και το amount. Η δεύτερη μακροεντολή ορίζει ότι κατά την εκτέλεση της εντολής Use Rule, για κάθε φορά που το amount περιέχει το 2000, θα αποδίδονται 5 πόντοι στο μετρητή Loyaltyc.
Η εντολή Use Rule εκτελείται από το πρόγραμμα "Σημείο Εξυπηρέτησης" κάθε φορά που ο πελάτης πραγματοποιεί μία αγορά με χρήση του ηλεκτρονικού πορτοφολιού του. Amount είναι η αξία της αγοράς πολλαπλασιασμένη επί 100 για να αφαιρεθεί από τον μετρητή Elepurse. Όπως έχει ειπωθεί στην ενότητα 6.2.4, ο μετρητής Elepurse, ο οποίος υλοποιεί το ηλεκτρονικό πορτοφόλι, για να μπορεί να αποθηκεύσει και δεκαδική αξία πολλαπλασιάζει κάθε ποσό επί 100 πριν το προσθέσει στο ήδη υπάρχον ποσό του. Έτσι και για να αφαιρεθεί κάποια αξία από αυτόν πρέπει πρώτα να πολλαπλασιαστεί με το 100. Ταυτόχρονα τώρα με την αφαίρεση της αξίας από το ηλεκτρονικό πορτοφόλι υπολογίζονται με τον κανόνα οι πόντοι που θα κερδίσει ο πελάτης και θα αποδοθούν στον μετρητή Loyaltyc της κάρτας του. Έχει οριστεί ότι για κάθε 20€ που πληρώνει ο πελάτης, κερδίζει 5 πόντους.
Έτσι, αν ο πελάτης κάνει μία αγορά των 56€, η εντολή Use Rule για τον Rule 1 θα έχει ως amount το 5600 και με την $1^{η}$ μακροεντολή θα αφαιρέσει από τον μετρητή Elepurse 5600 πόντους, ενώ με τη $2^{η}$ μακροεντολή θα αποδώσει στον μετρητή Loyaltyc 10 πόντους.
Το αρχείο επιτρέπει ανανέωση ή διαγραφή του μόνο μέσω SM με τη χρήση του SK2 και ελεύθερη ανάγνωση. Η χρησιμοποίηση του κανόνα όμως, η εκτέλεση δηλαδή της εντολής Use Rule για αυτόν τον κανόνα, επιτρέπεται να γίνει μόνο με SM με το κλειδί SK1 του διαχειριστή πωλήσεων.

6.3 Ανάλυση Συναρτήσεων

Σε αυτή την ενότητα θα περιγραφούν οι συναρτήσεις, οι διαδικασίες και οι μεταβλητές που χρησιμοποιήθηκαν στα 7 προγράμματα (Inistep, Personover, Overtime, Redemption, ServPoint, TicketCenter, Transportation) που δημιουργήθηκαν για την υλοποίηση των 3 εφαρμογών της διπλωματικής αυτής. Όλος ο κώδικας των προγραμμάτων βρίσκεται στο Παράρτημα Α.

6.3.1 Αρχείο "nasia.h"

Όπως ειπώθηκε και νωρίτερα, πολλές κοινές συναρτήσεις των προγραμμάτων καθώς και δηλώσεις κοινών σταθερών και γενικών μεταβλητών αποθηκεύτηκαν στο header αρχείο "nasia.h", του οποίου το περιεχόμενο θα εξεταστεί σε αυτή την ενότητα. Όλο το αρχείο παρατίθεται στις σελίδες 3 – 10 του Παραρτήματος Α.

6.3.1.1 Δηλώσεις Σταθερών και Γενικών Μεταβλητών

Οι σταθερές USERMAXT, OPERMAXT, TREASMAXT αντιπροσωπεύουν το μέγιστο αριθμό συνεχόμενων προσπαθειών παρουσίασης κωδικού για τον κάτοχο της κάρτας, τους 3 διαχειριστές των εφαρμογών (διαχειριστής μισθοδοσίας, διαχειριστής πωλήσεων, διαχειριστής πωλήσεων) και τον διαχειριστή θησαυροφυλακίου.
Οι σταθερές SKEYFILE, SCODEFILE, RULEFILE, COUNTFILE, RECORDFILE και SYSTEMFILE δηλώνουν το τύπο ενός αρχείου στην κάρτα και χρησιμοποιούνται κατά τη δημιουργία ενός νέου αρχείου.
Οι σταθερές SMKEY1, SMKEY2, SMKEY3 και SMKEY4 δηλώνουν τη τιμή ενός access condition που ορίζει προστασία Secure Messaging με τα SK1, SK2, SK3 και SK4 αντίστοιχα, ενώ η FREE δηλώνει access condition ελεύθερης πρόσβασης και η LOCKED κλειδωμένη πρόσβαση.
Οι σταθερές COMPKEY1, COMPKEY2 και COMPKEY3 είναι τα 3 company keys που η εταιρία έχει δημιουργήσει για τις 3 εφαρμογές.
Η μεταβλητή TerminalData περιέχει τα στοιχεία τερματικού και έχει μέγεθος 8 bytes. Η τιμή που έχει δοθεί είναι μία default τιμή και στην περίπτωση πραγματικής υλοποίησης θα είχε διαφορετική τιμή για το κάθε τερματικό στο οποίο θα έτρεχε το εκάστοτε πρόγραμμα.
Η παράμετρος Zeros είναι ένας (δεκαεξαδικός) πίνακας μηδενικών που χρησιμοποιείται για την εκτέλεση κάποιων εντολών οι οποίες δεν προστατεύονται ούτε από Secret Code ούτε από Secret Key και έτσι η παράμετρός τους Secret πρέπει να αποτελείται από μηδενικά στοιχεία.
Η παράμετρος Secret που υπάρχει στις περισσότερες συναρτήσεις της βιβλιοθήκης, περιέχει τον μυστικό κωδικό ή το μυστικό κλειδί από το οποίο προστατεύεται η εντολή που ζητείται να εκτελεστεί. Αν, όπως είπαμε, η εντολή δεν προστατεύεται η παράμετρος πρέπει να έχει μηδενική τιμή. Έχει μήκος 8 ή 16 bytes ανάλογα με το αν υπάρχει προστασία ή όχι από SM (Secure Messaging).
Η μεταβλητή TermLen δηλώνει το μέγεθος της μεταβλητής TerminalData, η ObjAttribLen δηλώνει το μέγεθος της μεταβλητής ObjAttrib που χρησιμοποιείται για να υποδείξει τις ιδιότητες ενός νέου αρχείου κατά τη δημιουργία του και η ProofLen δηλώνει το μήκος της μεταβλητής Proof που αποθηκεύει την απόδειξη συναλλαγής (Transaction Proof) όταν αυτή επιστρέφεται από την κάρτα.
Η μεταβλητή Class αντιστοιχεί στην τάξη-τύπο μίας εντολής (ενότητα 4.2.6), η μεταβλητή ObjectType δηλώνει τον τύπο ενός αρχείου, η ObjectId αποθηκεύει το SFI ενός αρχείου ενώ η μεταβλητή Element περιέχει το tag ενός στοιχείου αρχείου.
Η μεταβλητή Channel αποθηκεύει τη δομή που περιέχει τις παραμέτρους επικοινωνίας με τον reader σε ένα κανάλι επικοινωνίας ενώ η Session αντιστοιχεί στη δομή που περιέχει τις παραμέτρους επικοινωνίας με την κάρτα κατά τη διάρκεια μίας συνόδου.
Τέλος η μεταβλητή file είναι ένας περιγραφέας αρχείου που αντιστοιχίζεται σε ένα αρχείο όταν αυτό ανοίγει για λόγους αναγνώρισης.

6.3.1.2 Συναρτήσεις - Διαδικασίες

Στο header αρχείο περιέχονται οι ακόλουθες συναρτήσεις που χρησιμοποιούνται από ένα ή παραπάνω προγράμματα.

- OpenChannel: ορίζει τις παραμέτρους επικοινωνίας με τον reader, ανοίγει ένα κανάλι επικοινωνίας μεταξύ host συστήματος και reader, αποθηκεύει στη μεταβλητή ChanNb τον λογικό αριθμό του καναλιού που άνοιξε και τον επιστρέφει στο πρόγραμμα που έχει καλέσει τη συνάρτηση. Σε περίπτωση λάθους τυπώνει τον κωδικό λάθους που λαμβάνει από τον reader και τον επιστρέφει στο πρόγραμμα.
- OpenSession: ορίζει τις παραμέτρους συνόδου, ανοίγει μία σύνοδο με την κάρτα πάνω σε ήδη υπάρχον κανάλι, αποδίδει τον λογικό αριθμό της νέας συνόδου στη μεταβλητή και την επιστρέφει στο πρόγραμμα. Σε περίπτωση λάθους τυπώνει τον κωδικό λάθους που λαμβάνει από την κάρτα και τον επιστρέφει στο πρόγραμμα.

- WaitForCard: η συνάρτηση αυτή ελέγχει συνεχώς πάνω στο κανάλι την υποδοχή του reader μέχρι να εντοπίσει κάρτα σε αυτήν. Ο έλεγχος γίνεται με συνεχείς προσπάθειες να ανοίξει σύνοδο με την κάρτα. Όσο οι κλήσεις επιστρέφουν τον κωδικό –1 (ο οποίος επιστρέφεται από την OpenSession όταν η κάρτα είναι απούσα από τον reader) η συνάρτηση συνεχίζει τις προσπάθειες. Σε περίπτωση επιστροφής άλλου κωδικού λάθους ή επιτυχούς κωδικού από την OpenSession η συνάρτηση τερματίζει και επιστρέφει στο πρόγραμμα τον κωδικό αυτό.
- WaitForExit: η διαδικασία αυτή ελέγχει συνεχώς την υποδοχή του reader μέχρι να εξασφαλίσει ότι δεν υπάρχει κάρτα σε αυτήν. Ο έλεγχος γίνεται με συνεχείς προσπάθειες να διαβάσει τον CTC μετρητή της κάρτας. Όσο η εντολή ανάγνωσης αυτού του μετρητή G_ReadParam επιστρέφει τιμή διάφορη του –1 (που επιστρέφεται μετά από την κλήση μίας εντολής όταν δεν υπάρχει κάρτα στον reader) η διαδικασία συνεχίζει τις προσπάθειες ανάγνωσης. Μόλις η G_ReadParam επιστρέψει τον κωδικό –1 η διαδικασία τερματίζει. Η μεταβλητή Element υποδεικνύει ότι η παράμετρος που θα διαβαστεί από την G_ReadParam είναι ο CTC δηλώνοντας το tag του. Η παράμετρος SecretLen δηλώνει το μήκος της παραμέτρου Secret που αναφέραμε στην προηγούμενη ενότητα, η DataResponse αποθηκεύει τα δεδομένα της απάντησης της κάρτας στην εντολή Read Parameter και η LenExp περιέχει το μήκος της απάντησης.
- Closeall: η διαδικασία αυτή κλείνει την υπάρχουσα σύνοδο με την κάρτα και στη συνέχεια κλείνει το κανάλι επικοινωνίας με τον reader.
- ReadCTC: η συνάρτηση αυτή διαβάζει με τη χρήση της συνάρτησης βιβλιοθήκης G_ReadParam τον μετρητή CTC της κάρτας, αποθηκεύει τη τιμή του με κατάλληλο τρόπο στη μεταβλητή CTCValue και την επιστρέφει στο πρόγραμμα που την έχει καλέσει. Σε περίπτωση λάθους επιστρέφει τον κωδικό λάθους στο πρόγραμμα.
- UpdateCTC: ανανεώνει την τιμή του μετρητή CTC της κάρτας με τη χρήση της συνάρτησης βιβλιοθήκης G_UpdateParamEnh. Έχει ως όρισμα την παράμετρο NewValue η οποία περιέχει την τιμή που το πρόγραμμα επιθυμεί να αποδώσει στο μετρητή και την παράμετρο SM που υποδηλώνει αν θα χρησιμοποιηθεί Secure Messaging κατά την εκτέλεση της συνάρτησης ή όχι (η τιμή 1 σημαίνει ότι χρησιμοποιείται SM). Η παράμετρος ParamLen δείχνει το μήκος της παραμέτρου που επιθυμεί να ανανεώσει η συνάρτηση G_UpdateParamEnh (εδώ του CTC), ενώ οι MacIn και MacOut αντιστοιχούν στα MAC_IN και MAC_OUT που ανταλλάσσονται κατά τη λειτουργία SM που τυχόν προστατεύει την ανανέωση του CTC. Σε περίπτωση λάθους επιστρέφει ένα ειδικό κωδικό λάθους στο πρόγραμμα, αλλιώς επιστρέφει το 0.
- ReadCountBalance: διαβάζει την τιμή του μετρητή που καθορίζει η παράμετρος SFI της συνάρτησης. Τη τιμή αυτή αποθηκεύει με κατάλληλο τρόπο στην μεταβλητή Balance την οποία επιστρέφει στο πρόγραμμα. Σε περίπτωση λάθους επιστρέφει τον κωδικό λάθους στο πρόγραμμα.
- UpdateCountBal: μηδενίζει την τιμή του μετρητή που καθορίζει η παράμετρος SFI. Η παράμετρος SM, όπως ειπώθηκε και πριν, χρησιμεύει για να υποδείξει στη συνάρτηση τη χρήση ή όχι Secure Messaging. Απαραίτητη επίσης είναι η γνώση στη συνάρτηση της τιμής του CTC (παράμετρος CTCValue) η οποία ανανεώνεται μετά τη διαδικασία SM. Το κλειδί ή ο κωδικός που μπορεί να προστατεύει την ανανέωση της τιμής του μετρητή αυτού, περιέχεται στην παράμετρο Secret. Η ανανέωση της τιμής του μετρητή γίνεται με τη συνάρτηση βιβλιοθήκης G_UpdateParamEnh. Αν ο μηδενισμός της τιμής του μετρητή αποτύχει, επιστρέφεται ο κωδικός λάθους στο πρόγραμμα, αλλιώς επιστρέφεται το 0.
- VerifySCode: συγκρίνει τον κωδικό που περιέχεται στην παράμετρο Secret με τον κωδικό που είναι αποθηκευμένος στο Secret Code αρχείο που υποδηλώνει η παράμετρος SFI. Αν η επαλήθευση είναι επιτυχής τυπώνει μήνυμα επιτυχίας και επιστρέφει στο πρόγραμμα τη τιμή 0. Αν όχι, τυπώνει μήνυμα λάθους και επιστρέφει στο πρόγραμμα την τιμή του κωδικού λάθους που επιστρέφει η κάρτα.

- ReadRecord: διαβάζει με την G_ReadRecord από το record file που ορίζει η παράμετρος SFI το record νούμερο RecNum και το αποτέλεσμα της ανάγνωσης αποθηκεύεται στη παράμετρο RecVal και έχει μήκος RecLen. Η παράμετρος SFI δεν περιέχει το ίδιο το SFI του αρχείου αλλά μία κωδικοποιημένη μορφή του. Συγκεκριμένα έχει τη δυαδική μορφή xxxxx100 όπου τα 5 bits από αριστερά προς τα δεξιά περιέχουν το πραγματικό SFI του record file. Έτσι αν θέλουμε να διαβάσουμε το record file με SFI 05h, η παράμετρος SFI θα έχει τη τιμή 2Ch = 0010 $1100_{<2>}$. Η μεταβλητή TrueSFI υπολογίζει το πραγματικό SFI του αρχείου. Σε περίπτωση μη επιτυχούς ανάγνωσης η συνάρτηση τυπώνει το μήνυμα λάθους και το επιστρέφει στο πρόγραμμα ενώ στην αντίθετη περίπτωση επιστρέφει το 0.
- UpdateRecParam: ρυθμίζει τις access conditions για ανανέωση / διαγραφή και ανάγνωση του record file που ορίζει η παράμετρος SFI. Η τιμή των access conditions ορίζεται στην παράμετρο ParamVal μήκους (ParamLen) 2 bytes. Σε περίπτωση λάθους τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα, αλλιώς τυπώνει μήνυμα επιτυχίας και επιστρέφει το 0.
- UpdateRec: ανανεώνει με την G_UpdateRecord το περιεχόμενο του record νούμερο RecNum που βρίσκεται στο record file που ορίζει η παράμετρος SFI. Το record έχει μήκος RecLen και παίρνει τη τιμή που ορίζει η παράμετρος RecVal. Αυτή η συνάρτηση χρησιμοποιείται όταν το record file έχει ελεύθερη πρόσβαση. Όταν η ανανέωση είναι επιτυχής επιστρέφει στο πρόγραμμα το 0 αλλιώς επιστρέφει στο πρόγραμμα τον κωδικό λάθους που έχει στείλει η κάρτα.
- UpdateRecSM: ανανεώνει με την G_SMUpdateRecordEnh το περιεχόμενο του record νούμερο RecNum στο record file που ορίζει η παράμετρος SFI. Το record έχει μήκος RecLen και παίρνει τη τιμή που ορίζει η παράμετρος RecVal. Η συνάρτηση αυτή χρησιμοποιείται όταν το record file προστατεύεται με SM. Το SK που χρειάζεται η διαδικασία SM βρίσκεται στην παράμετρο Secret και μετά την εκτέλεση της ανανέωσης οι μεταβλητές MacIn και MacOut περιέχουν αντίστοιχα τα MAC_IN και MAC_OUT που ανταλλάχθηκαν μεταξύ συστήματος και κάρτας. Η παράμετρος SFI περιέχει το SFI του record file στο οποίο αναφέρεται, με ίδιο τρόπο όπως και στην ReadRecord συνάρτηση. Όταν η ανανέωση είναι επιτυχής επιστρέφει στο πρόγραμμα το 0 αλλιώς επιστρέφει στο πρόγραμμα τον κωδικό λάθους που έχει στείλει η κάρτα.
- Award: η συνάρτηση αυτή αποδίδει στον μετρητή που υποδεικνύει η παράμετρος SFI τόσους πόντους όσους περιέχει η παράμετρος Amount χρησιμοποιώντας την συνάρτηση βιβλιοθήκης G_AwardEnh. Η απόδοση πόντων μπορεί να γίνει είτε με SM είτε με επαλήθευση μυστικού κωδικού ή και χωρίς προστασία. Η παράμετρος SM της συνάρτησης ενημερώνει αν θα χρησιμοποιηθεί Secure Messaging. Η παράμετρος Secret, όπως έχει ειπωθεί και νωρίτερα, περιέχει τον μυστικό κωδικό ή το μυστικό κλειδί που τυχόν χρειάζεται να παρουσιαστεί για να γίνει η απόδοση των πόντων στο συγκεκριμένο μετρητή. Αν ο μετρητής επιθυμεί τον υπολογισμό απόδειξης συναλλαγής (Transaction Proof) αυτή αποθηκεύεται μετά την εκτέλεση της απόδοσης πόντων στη μεταβλητή Proof και το μήκος της αποθηκεύεται στη μεταβλητή ProofLen. Η μεταβλητή Balance περιέχει μετά την εκτέλεση της απόδοσης πόντων τη νέα τιμή του μετρητή, την οποία και η συνάρτηση τυπώνει. Σε περίπτωση επιτυχίας επιστρέφει στο πρόγραμμα τη τιμή 0 ενώ αν παρουσιαστεί σφάλμα, τυπώνει και επιστρέφει στο πρόγραμμα τον κωδικό λάθους.
- Redeem: λειτουργεί ακριβώς όπως και η Award, μόνο που εκτελεί αφαίρεση πόντων από ένα μετρητή. Η νέα τιμή του μετρητή μετά την αφαίρεση των πόντων αποθηκεύεται στη μεταβλητή Balance. Αν η αφαίρεση πόντων είναι επιτυχής επιστρέφει στο πρόγραμμα τη τιμή 0 ενώ αν παρουσιαστεί σφάλμα τυπώνει και επιστρέφει στο πρόγραμμα τον κωδικό λάθους.
- CreateorAppendRec: δημιουργεί και προσθέτει ένα νέο record στο record file που ορίζει η παράμετρος SFI, με χρήση της συνάρτησης βιβλιοθήκης G_AppendRecord. Το περιεχόμενο της νέας εγγραφής και το μήκος της περιέχονται αντίστοιχα στις

παραμέτρους RecVal και RecLen. Η παράμετρος SFI περιέχει το SFI του αρχείου με τη δυαδική μορφή xxxxx000, όπου στα bits xxxxx περιέχεται το πραγματικό SFI. Αν η εντολή εκτελεστεί σωστά, η συνάρτηση τυπώνει μήνυμα επιτυχίας και επιστρέφει το 0 ενώ αν είναι ανεπιτυχής τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα.

- Nameof: αυτή η συνάρτηση είναι βοηθητική και χρησιμοποιείται για να διαβάσει από το Record File 2 το record με νούμερο RecNum. Το περιεχόμενό του αποθηκεύεται στη παράμετρο RecVal και έχει μήκος RecLen και εκτελείται με κλήση της ReadRecordReadRecordReadRecord. Κυρίως χρησιμοποιείται για να διαβάσει κάποιο από τα 3 πρώτα records αυτού του αρχείου που περιέχουν το όνομα, το επίθετο και το όνομα πατρός του κατόχου αντίστοιχα.
- Salute: η συνάρτηση αυτή χρησιμοποιείται για να ελέγξει την ταυτότητα του κατόχου της κάρτας ή την ταυτότητα του διαχειριστή μισθοδοσίας όταν αυτός επιχειρήσει να εισάγει χειροκίνητα τις ώρες εργασίας του κατόχου της κάρτας στο πρόγραμμα Overtime. Η παράμετρος who προσδιορίζει ποιον θα πιστοποιήσει η συνάρτηση (τον κάτοχο της κάρτας ή το διαχειριστή) ενώ η maxt προσδιορίζει ποιος είναι ο μέγιστος αριθμός συνεχόμενων προσπαθειών παρουσίασης του κωδικού πριν αυτός κλειδωθεί και η κάρτα απορριφθεί. Στην περίπτωση που θα ελέγξει τον κάτοχο της κάρτας, η συνάρτηση διαβάζει με την Nameof το όνομα και το επίθετο του κατόχου, τον καλωσορίζει και του ζητάει να παρουσιάσει τον κωδικό του ενώ στην άλλη περίπτωση καλωσορίζει το διαχειριστή και πάλι ζητάει την παρουσίαση του κωδικού. Μέχρι να δοθεί ο σωστός κωδικός και να εξαντληθούν οι maxt προσπάθειες, η συνάρτηση ζητάει τον κωδικό αυτό, ο οποίος κάθε φορά αποθηκεύεται στη μεταβλητή Pin και μετά εκτελείται επαλήθευση με την χρήση της VerifySCode. Η επαλήθευση μπορεί επίσης να αποτύχει λόγω απουσίας της κάρτας από τον reader. Μόλις ο κύκλος επαληθεύσεων τερματίσει, η συνάρτηση επιστρέφει στο πρόγραμμα τον κωδικό που έχει επιστρέψει η VerifySCode.
- LoadPurse: η συνάρτηση αυτή διαβάζει το περιεχόμενο του μετρητή Elepurse με χρήση της ReadCountBalance. Αν η ανάγνωση γίνει επιτυχώς η τιμή του αποθηκεύεται στην παράμετρο countpurse. Η συνάρτηση επιστρέφει στο πρόγραμμα τον κωδικό που επιστρέφει η ReadCountBalance (0 για επιτυχή ανάγνωση).
- LoadLoyal: η συνάρτηση διαβάζει με χρήση της ReadCountBalance το περιεχόμενο του μετρητή Loyaltyc. Αν η ανάγνωση επιτύχει η τιμή του αποθηκεύεται στην παράμετρο countloyal. Έπειτα επιστρέφει στο πρόγραμμα τον κωδικό που επιστρέφει η ReadCountBalance (0 στην επιτυχή ανάγνωση).
- Divide4bytes: υπολογίζει την δεκαεξαδική τιμή μήκους 4 bytes του ακέραιου αριθμού που περιέχει η timein και την αποθηκεύει στη παράμετρο RecVal. Χρησιμοποιείται για να αποθηκεύσει τη μεταβλητή timein του προγράμματος Overtime στην κάρτα.
- Divide2bytes: υπολογίζει την δεκαεξαδική τιμή μήκους 2 bytes του ακέραιου αριθμού που περιέχει η παράμετρος var και την αποθηκεύει στην παράμετρο RecVal.
- Compile2bytes: υπολογίζει τον ακέραιο αριθμό που περιέχεται σε δεκαεξαδική μορφή μήκους 2 bytes στην παράμετρο RecVal και τον επιστρέφει στο πρόγραμμα.
- Compile4bytes: υπολογίζει τον ακέραιο αριθμό που περιέχεται σε δεκαεξαδική μορφή μήκους 4 bytes στην παράμετρο RecVal και τον επιστρέφει στο πρόγραμμα. Χρησιμοποιείται για να μπορεί να διαβαστεί από το πρόγραμμα σε μορφή ημερολογιακής ώρας η μεταβλητή timein που στην κάρτα είναι αποθηκευμένη στο record 1 του record file 1 στα 4 τελευταία bytes του ($byte_8$, $byte_9$, $byte_{10}$ και $byte_{11}$).
- Debstrfl: η συνάρτηση αυτή αποκωδικοποιεί ένα string που έχει δοθεί στην είσοδο από τον χρήστη του προγράμματος και το οποίο αντιστοιχεί σε δεκαδικό αριθμό και επιστρέφει την αξία του πολλαπλασιασμένη με το 100. Ο χρήστης μπορεί να εισάγει είτε ένα ακέραιο αριθμό είτε ένα δεκαδικό αριθμό με την τελεία να διαχωρίζει το ακέραιο από το δεκαδικό μέρος. Ο αριθμός που δίνει ο χρήστης διαβάζεται από το

πρόγραμμα ως μία συμβολοακολουθία και γι' αυτό δίνεται στην συνάρτηση debstrfl για να τον μετατρέψει σε κανονικό αριθμό. Η συνάρτηση χρησιμοποιεί τη μεταβλητή valuea για να αποθηκεύσει το ακέραιο μέρος του αριθμού και την μεταβλητή valueb για το δεκαδικό μέρος (αν αυτό υπάρχει). Αρχικά ελέγχει τη συμβολοακολουθία που υπάρχει στη παράμετρο string και αν δεν συναντήσει το χαρακτήρα της τελείας συμπεραίνει πως το string παριστάνει ακέραιο αριθμό τον οποίο υπολογίζει, αποθηκεύει στην valuea, τον πολλαπλασιάζει επί 100 και τον επιστρέφει στο πρόγραμμα. Αλλιώς υπολογίζει ξεχωριστά το ακέραιο και το δεκαδικό μέρος και αφού τα πολλαπλασιάζει κατάλληλα με το 100 επιστρέφει πάλι έναν ακέραιο αριθμό. Αν δηλαδή το string είναι το 32.05 επιστρέφει στο πρόγραμμα τον αριθμό 3205. Οι πολλαπλασιασμοί επί 100 γίνονται γιατί αυτή η συνάρτηση χρησιμοποιείται για να μετατρέψει μία χρηματική αξία σε πόντους που θα προστεθούν σε μετρητή (όπως αναφέραμε στην περίπτωση των μετρητών Advancep, Elepurse και Ticketin). Η συνάρτηση αυτή, όπως και η επόμενη, ελέγχουν την ορθότητα της εισόδου που δίνει ο χρήστης (π.χ. δεν μπορούν να περιλαμβάνονται στο string χαρακτήρες γραμμάτων) και αν αυτή δεν είναι σωστή τυπώνει και επιστρέφει μήνυμα λάθους.

- Debstrint: αυτή η συνάρτηση λειτουργεί σχεδόν όπως και η debstrint με τη διαφορά ότι περιμένει η είσοδος από το χρήστη που βρίσκεται στο string να είναι ακέραιος αριθμός τον οποίο έπειτα υπολογίζει και επιστρέφει ως έχει (χωρίς δηλαδή να τον πολλαπλασιάσει επί 100). Χρησιμοποιείται περισσότερο για να διαβάσει το πρόγραμμα από την είσοδο την επιλογή του διαχειριστή, όταν αυτός καλείται να επιλέξει ανάμεσα σε διάφορες λειτουργίες (π.χ. στο πρόγραμμα Redemption ή στο πρόγραμμα ServPoint).

6.3.2 Πρόγραμμα Inistep (Αρχικοποίηση)

Η δομή και οι λειτουργίες αυτού του προγράμματος έχουν αναλυθεί στην ενότητα 5.4.1. Σε αυτή την ενότητα θα περιγράψουμε τις συναρτήσεις που χρησιμοποιεί και κάποια σημαντικά στοιχεία. Όλο το πρόγραμμα παρατίθεται στις σελίδες 11 – 20 του Παραρτήματος Α.

6.3.2.1 Επισημάνσεις Παραμέτρων

Όπως αναλύθηκε και στις συναρτήσεις που περιγράφηκαν στην προηγούμενη ενότητα, υπάρχουν κάποιες παράμετροι – μεταβλητές που έχουν σε όλες τις συναρτήσεις τον ίδιο ρόλο. Θα τις αναφέρουμε ξανά για ευκολία κατανόησης των επόμενων συναρτήσεων.
Η ChanNb περιέχει τον λογικό αριθμό του καναλιού που έχει ανοίξει μεταξύ host συστήματος και reader.
Η Secret περιέχει τον μυστικό κωδικό ή το μυστικό κλειδί από το οποίο μπορεί να προστατεύεται η εντολή που ζητείται να εκτελεστεί και έχει μήκος SecretLen.
Η SM δηλώνει αν θα χρησιμοποιηθεί Secure Messaging ή όχι.
Η TermLen δηλώνει το μέγεθος της μεταβλητής TerminalData που είναι τα χαρακτηριστικά του τερματικού.
Η ObjAttrib ορίζει τις ιδιότητες ενός νέου αρχείου και έχει μήκος ObjAttribLen.
Οι μεταβλητές MacIn και MacOut περιέχουν αντίστοιχα τα MAC_IN και MAC_OUT που ανταλλάσσονται μεταξύ συστήματος και κάρτας όταν πραγματοποιείται Secure Messaging.
Η ProofLen δηλώνει το μήκος της μεταβλητής Proof που αποθηκεύει την απόδειξη συναλλαγής (Transaction Proof) όταν αυτή επιστρέφεται από την κάρτα.
Η μεταβλητή Class αντιστοιχεί στην τάξη μίας εντολής, η μεταβλητή ObjectType δηλώνει τον τύπο ενός αρχείου, η ObjectId αποθηκεύει το SFI ενός αρχείου ενώ η μεταβλητή Element περιέχει το tag μίας παραμέτρου ενός αρχείου.
Η μεταβλητή CTCValue περιέχει τη τιμή του μετρητή CTC της κάρτας, η Balance αποθηκεύει τη τιμή ενός μετρητή και η RenNum προσδιορίζει το νούμερο ενός record.

6.3.2.2 Συναρτήσεις - Διαδικασίες

- CreateSystemFile: η συνάρτηση αυτή δημιουργεί στην κάρτα το αρχείο συστήματος χρησιμοποιώντας τις συναρτήσεις βιβλιοθήκης G_BuildCreateObjectDataIn και G_CreateObject. Σε περίπτωση σφάλματος τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα ενώ αν είναι επιτυχής επιστρέφει το 0.
- UpdateSysParam: ορίζει μέσω της G_UpdateParamEnh τις παραμέτρους του αρχείου συστήματος, δηλαδή τα access conditions του αρχείου και τις αναφορές για τα αρχεία PIN EMV-DIR, σύμφωνα με το περιεχόμενο της παραμέτρου ParamVal. Οι ιδιότητες του αρχείου αυτού έχουν αναλυθεί στην ενότητα 6.2.1 Αν εκτελεστεί επιτυχώς επιστρέφει στο πρόγραμμα το 0 αλλιώς τυπώνει και επιστρέφει τον κωδικό σφάλματος.
- DeleteSystemFile: διαγράφει το αρχείο συστήματος με χρήση της συνάρτησης βιβλιοθήκης G_DeleteObjectEnh. Σε περίπτωση σφάλματος τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα ενώ αν είναι επιτυχής επιστρέφει το 0.
- CreateSKeyFile: δημιουργεί ένα αρχείο μυστικού κλειδιού (Secret Key File) χρησιμοποιώντας την συνάρτηση βιβλιοθήκης G_CreateObjectEnh. Αν είναι επιτυχής επιστρέφει και αυτή το 0 αλλιώς επιστρέφει στο πρόγραμμα τον κωδικό σφάλματος.
- CreateSCodeFile: δημιουργεί ένα αρχείο μυστικού κωδικού (Secret Code File) μέσω της G_CreateObjectEnh. Αν είναι επιτυχής επιστρέφει στο πρόγραμμα το 0 αλλιώς επιστρέφει τον κωδικό σφάλματος.
- CreateCounterFile: η συνάρτηση αυτή δημιουργεί ένα αρχείο μετρητή στην κάρτα χρησιμοποιώντας επίσης την G_CreateObjectEnh. Οι μετρητές που δημιουργούνται έχουν ενεργοποιημένες τις παραμέτρους Cumulative Balance, Visit Counter, Rules και Label. Σε περίπτωση σφάλματος τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα ενώ αν είναι επιτυχής επιστρέφει το 0.
- CreateRuleFile: η συνάρτηση δημιουργεί ένα αρχείο κανόνα στην κάρτα μέσω της G_CreateObjectEnh. Επειδή ο μόνος κανόνας που θα υπάρχει στην κάρτα θα περιέχει 2 μακροεντολές, στη δημιουργία του αρχείου ορίζεται ότι μπορεί να περιέχει μέχρι 2 μακροεντολές. Αν το αρχείο δημιουργηθεί επιτυχώς επιστρέφεται στο πρόγραμμα η τιμή 0 αλλιώς επιστρέφεται ο κωδικός λάθους που έχει προκύψει.
- CreateRecFile: δημιουργεί στην κάρτα μέσω της G_CreateObjectEnh ένα record file του οποίου το μέγεθος ορίζεται από την παράμετρο RecSize. Αν η δημιουργία επιτύχει επιστρέφει στο πρόγραμμα την τιμή 0 αλλιώς επιστρέφει κωδικό λάθους.
- SetKeyValue: ρυθμίζει τις παραμέτρους του μυστικού κλειδιού που ορίζει η παράμετρος SFI με διαδοχικές χρήσεις της συνάρτησης βιβλιοθήκης G_UpdateParamEnh. Συγκεκριμένα, η παράμετρος KeyMSB ορίζει τη μισή τιμή του κλειδιού (τα 8 αριστερά bytes), η KeyLSB ορίζει την άλλη μισή τιμή του κλειδιού (τα 8 δεξιά bytes) και η παράμετρος max ορίζει τον μέγιστο αριθμό συνεχόμενων προσπαθειών παρουσίασης του κλειδιού. Επίσης ορίζονται τα access conditions του αρχείου. Οι παράμετροι αυτοί για κάθε ένα από τα 4 κλειδιά που θα δημιουργηθούν στην κάρτα ορίζονται σύμφωνα με όσα αναφέρθηκαν στην ενότητα 6.2.2. Η συνάρτηση σε περίπτωση σφάλματος τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα ενώ αν είναι επιτυχής επιστρέφει το 0.
- Producekey: η διαδικασία αυτή χρησιμοποιείται για να επαληθεύσει το πρόγραμμα το μυστικό κλειδί του διαχειριστή θησαυροφυλακίου. Λειτουργεί παρόμοια με την συνάρτηση Salute με τη διαφορά ότι γνωρίζει σε ποιον απευθύνεται. Ο διαχειριστής μετά από προτροπή της διαδικασίας εισάγει και τον κωδικό του (αυτόν που δημιούργησε η εταιρία) και το αντίστοιχο company key και η διαδικασία υπολογίζει το κρυπτόγραμμα που προκύπτει και το αποθηκεύει στην παράμετρο skey.
- Produce4key: η διαδικασία αυτή είναι υπεύθυνη για τη δημιουργία των 4 μυστικών κλειδιών των διαχειριστών με τη διαδικασία που έχουμε αναλύσει στην ενότητα 6.2.2. Τα 4 κρυπτογραφημένα κλειδιά που δημιουργούνται με αυτή τη διαδικασία

αποθηκεύονται αντίστοιχα στις παραμέτρους skey1, skey2, skey3 και skey4 για να αποθηκευτούν αργότερα από το πρόγραμμα στα αντίστοιχα αρχεία κλειδιών.

- Initial: η συνάρτηση αυτή καλείται από το κύριο πρόγραμμα για να πραγματοποιήσει την αρχικοποίηση της κάρτας, το οποίο κάνει με κατάλληλες κλήσεις των συναρτήσεων που αναφέρθηκαν παραπάνω. Έτσι ανοίγει κανάλι και σύνοδο επικοινωνίας, δημιουργεί το αρχείο συστήματος, 2 αρχεία μυστικών κωδικών, 4 αρχεία μυστικών κλειδιών, 9 αρχεία μετρητών, 1 αρχείο κανόνα και 7 αρχεία εγγραφών. Στα αρχεία εγγραφών 3 ως και 7 δημιουργεί και αρχικοποιεί τα records τους και ρυθμίζει τις παραμέτρους τους. Έπειτα με την produce4key δημιουργεί τα 4 κλειδιά των διαχειριστών και ανανεώνει τις παραμέτρους των αρχείων κλειδιών με τις τιμές αυτές. Τέλος μηδενίζει τον μετρητή CTC και με την UpdateSysParam ανανεώνει τις παραμέτρους του συστήματος. Αν σε οποιαδήποτε κλήση συνάρτησης επιστραφεί κωδικός λάθους, η Initial σταματάει ενώ επιστρέφει κατευθείαν τον κωδικό αυτό στο κύριο πρόγραμμα. Αν όλες οι συναρτήσεις εκτελεστούν κανονικά, κλείνει το κανάλι και τη σύνοδο επικοινωνίας και επιστρέφει στο κύριο πρόγραμμα τη τιμή 0.
- Delete: η συνάρτηση Delete καλείται από το κύριο πρόγραμμα για να διαγράψει το αρχείο συστήματος της κάρτας. Η διαγραφή του αρχείου συστήματος, όπως ειπώθηκε στην ενότητα 6.2.1, γίνεται μόνο με SM με τη χρήση του SK4 του διαχειριστή θησαυροφυλακίου. Το κλειδί δημιουργείται με την συνάρτηση producekey που αναφέρθηκε προηγουμένως και αν δεν είναι σωστό δεν επιτυγχάνει η διαγραφή του αρχείου συστήματος. Σε περίπτωση αποτυχίας της διαδικασίας, η συνάρτηση επιστρέφει στο κύριο πρόγραμμα ένα κωδικό λάθους αλλιώς επιστρέφει την τιμή 0.

6.3.3 Πρόγραμμα Personover (Προσωποποίηση)

Η δομή και οι λειτουργίες του προγράμματος “Προσωποποίηση” έχουν αναλυθεί στην ενότητα 5.4.2. Σε αυτή την ενότητα παρατίθενται οι συναρτήσεις και οι διαδικασίες που χρησιμοποιεί. Ισχύουν οι ίδιες επισημάνσεις παραμέτρων με αυτές της ενότητας 6.3.2.1. Ολόκληρο το πρόγραμμα παρατίθεται στις σελίδες 21 - 32 του Παραρτήματος Α.

6.3.3.1 Συναρτήσεις - Διαδικασίες

- SetCodeValue: η συνάρτηση ρυθμίζει τις παραμέτρους του αρχείου μυστικού κωδικού που ορίζει η παράμετρος SFI. Συγκεκριμένα η παράμετρος CodeVal ορίζει τη τιμή του κωδικού και η παράμετρος max ορίζει τον μέγιστο αριθμό συνεχόμενων προσπαθειών παρουσίασης του κωδικού. Χρησιμοποιεί την συνάρτηση βιβλιοθήκης G_UpdateParamEnh 2 φορές. Ρυθμίζονται επίσης και τα access conditions του αρχείου, σύμφωνα με όσα έχουν αναλυθεί στην ενότητα 6.2.3. Σε περίπτωση οποιουδήποτε σφάλματος τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα ενώ αν είναι επιτυχής επιστρέφει το 0.
- UpdateCountParamA: με την συνάρτηση αυτή ρυθμίζονται κάποιες από τις παραμέτρους του μετρητή τον οποίο υποδεικνύει η παράμετρος SFI. Με διαδοχικές κλήσεις της G_UpdateParamEnh ρυθμίζεται, σύμφωνα με τις τιμές των παραμέτρων AllowRul και Label αντίστοιχα, η παράμετρος του μετρητή που δηλώνει ποιοι κανόνες επιτρέπεται να επιδράσουν στον μετρητή αυτό καθώς επίσης και η ετικέτα που θα έχει ο μετρητής. Οποιοδήποτε σφάλμα προκύψει επιστρέφεται στο κύριο πρόγραμμα ενώ αν η εκτέλεση της συνάρτησης είναι επιτυχής, επιστρέφεται στο πρόγραμμα το 0.
- UpdateCountParamB: είναι συμπληρωματική της UpdateCountParamA συνάρτησης και ρυθμίζει τα access conditions του μετρητή που δηλώνει η παράμετρος SFI καθώς και το κλειδί που χρησιμοποιεί ο μετρητής αυτός για transaction proof. Οι τιμές

όλων των παραμέτρων για κάθε μετρητή δηλώνονται στην ενότητα 6.2.4. Για την ανανέωση των παραμέτρων του μετρητή χρησιμοποιείται και εδώ η συνάρτηση βιβλιοθήκης G_UpdateParamEnh. Σε περίπτωση σφάλματος τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα ενώ αν είναι επιτυχής επιστρέφει το 0.

- UpdateRuleMacro: ορίζει την μακροεντολή νούμερο MacroNum του αρχείου κανόνα SFI. Το περιεχόμενο της μακροεντολής περιέχεται στην παράμετρο MacroVal και ανανεώνεται με χρήση της G_UpdateParamEnh. Αν η ανανέωση γίνει με επιτυχία επιστρέφει στο κύριο πρόγραμμα το 0 αλλιώς επιστρέφει τον κωδικό σφάλματος.
- UpdateRuleParam: ρυθμίζει τις παραμέτρους του αρχείου κανόνα που δηλώνει η παράμετρος SFI χρησιμοποιώντας και αυτή την συνάρτηση βιβλιοθήκης G_UpdateParamEnh. Συγκεκριμένα ορίζει τις access conditions, το κλειδί που χρησιμοποιείται για transaction proof και την έκδοση του αρχείου κανόνα. Οι τιμές των παραμέτρων αυτών έχουν συμπληρωθεί σύμφωνα με όσα αναφέρθηκαν στην ενότητα 6.2.6. Αν παρουσιαστεί σφάλμα στην εκτέλεση, ο κωδικός του επιστρέφεται στο κύριο πρόγραμμα ενώ αν η συνάρτηση εκτελεστεί επιτυχώς, επιστρέφει στο πρόγραμμα το 0.
- Producekey: η λειτουργία αυτή ζητάει από τον διαχειριστή μισθοδοσίας να εισάγει τον κωδικό του και αφού τον αποθηκεύσει στη μεταβλητή Pin, τον χρησιμοποιεί μαζί με το company key που είναι αποθηκευμένο στην παράμετρο ckey για να δημιουργήσει με τον αλγόριθμο 3DES ένα κρυπτόγραμμα το οποίο αποθηκεύεται στην παράμετρο skey. Αν ο κωδικός που έχει δώσει ο διαχειριστής είναι σωστός, το κρυπτόγραμμα θα είναι ίδιο με το Secret Key (SK2) του διαχειριστή μισθοδοσίας, αυτό είναι κάτι όμως που το ελέγχει το πρόγραμμα.
- Debdate: η συνάρτηση αυτή "αποκωδικοποιεί" το string που περιέχει την ημερομηνία γέννησης του κατόχου και το οποίο έχει εισάγει ο διαχειριστής υπό τη μορφή HH-MM-EEEE, σε μορφή κατάλληλη για να αποθηκευτεί στο Record 4 του Record File 2 της κάρτας. Το αποκωδικοποιημένο string αποθηκεύεται στη παράμετρο RecVal. Αν ο διαχειριστής δεν έχει εισάγει την ημ/νία γέννησης υπό τη σωστή μορφή, η debdate τυπώνει μήνυμα λάθους και επιστρέφει στο πρόγραμμα τη τιμή –1, αλλιώς επιστρέφει μηδενική η θετική τιμή.
- Debsal: αυτή η συνάρτηση "αποκωδικοποιεί" το string που αντιστοιχεί στο ωρομίσθιο του εργαζόμενου το οποίο ο διαχειριστής συμπληρώνει είτε σε ακέραιη είτε σε δεκαδική μορφή. Η αποκωδικοποιημένη τιμή που έχει κατάλληλη μορφή για να αποθηκευτεί στην κάρτα (Record 10 - Record File 2), αποθηκεύεται στη παράμετρο RecVal. Η συνάρτηση ελέγχει ότι το ωρομίσθιο έχει εισαχθεί με σωστή μορφή και αν όχι τυπώνει μήνυμα λάθους και επιστρέφει στο πρόγραμμα τη τιμή –1, αλλιώς επιστρέφει το 0.
- Person: η συνάρτηση αυτή καλείται από το κύριο πρόγραμμα για να πραγματοποιήσει την προσωποποίηση μίας κάρτας, το οποίο γίνεται με κατάλληλες κλήσεις των συναρτήσεων που αναφέρθηκαν. Αρχικά ανοίγει κανάλι και σύνοδο επικοινωνίας με τον reader και την κάρτα αντίστοιχα. Έπειτα ελέγχει την ταυτότητα του διαχειριστή χρησιμοποιώντας την συνάρτηση producekey και τη μέθοδο που έχει αναλυθεί στην ενότητα 6.2.5, στο τμήμα που αναφέρεται στο Record File 4. Μετά ανανεώνει τις παραμέτρους των δύο αρχείων μυστικών κωδικών (τις τιμές των κωδικών τις εισάγει ο ίδιος) και όλων των μετρητών. Στη συνέχεια ορίζονται οι δύο μακροεντολές και οι παράμετροι του αρχείου κανόνα και τέλος συμπληρώνονται τα records των Record Files 1 και 2, ανανεώνονται οι παράμετροι των δύο αυτών αρχείων και κλείνει η επικοινωνία με τον reader και την κάρτα. Αν στη διαδικασία προσωποποίησης κάποια συνάρτηση επιστρέψει κωδικό λάθους, η Person τον επιστρέφει στο κύριο πρόγραμμα. Αν πάλι όλες οι συναρτήσεις εκτελεσθούν επιτυχώς, η Person επιστρέφει τη τιμή 0 στο κύριο πρόγραμμα.

6.3.4 Πρόγραμμα Overtime (Υπερωρίες)

Το πρόγραμμα αυτό έχει αναλυθεί ως προς τη δομή του και τις λειτουργίες στην ενότητα 5.1.3.1. Ισχύουν και σε αυτό το πρόγραμμα οι επισημάνσεις παραμέτρων που παρουσιάστηκαν στην ενότητα 6.3.2.1. Στις παραμέτρους αυτές θα προστεθεί η γενική μεταβλητή award που χρησιμοποιείται για να αποθηκεύει προσωρινά τους πόντους που πρέπει να προστεθούν στους μετρητές 1 ως και 5. Επίσης χρησιμοποιείται η μεταβλητή Key στην οποία αποθηκεύεται (για όσο το πρόγραμμα τρέχει) το μυστικό κλειδί του διαχειριστή, το οποίο χρησιμοποιείται για τις διαδικασίες Secure Messaging που θα εκτελεστούν. Ο κώδικας όλου του προγράμματος βρίσκεται στο Παράρτημα Α, στις σελίδες 33 – 40.

6.3.4.1 Συναρτήσεις - Διαδικασίες

- Producekey: η λειτουργία αυτή είναι ίδια με την Producekey του προγράμματος Personover. Χρησιμεύει για να επαληθεύει το πρόγραμμα τη ταυτότητα του διαχειριστή. Δημιουργεί απλά ένα κρυπτόγραμμα το οποίο πρέπει να ταιριάζει με το μυστικό κλειδί του διαχειριστή.
- LoadCounters: η συνάρτηση αυτή διαβάζει με χρήση της συνάρτησης ReadCountBalance τις τιμές των μετρητών 1, 2, 3, 4 και 5 (Normhour, Normmins, Yperhour, Ypermins και Fifteens) και τις αποθηκεύει αντίστοιχα στις παραμέτρους countnh, countnm, countyh, countym και countfif. Αν έστω και μία από τις αναγνώσεις αποτύχει, ο κωδικός σφάλματος επιστρέφεται στο πρόγραμμα ενώ στην αντίθετη περίπτωση επιστρέφεται κωδικός επιτυχίας.
- LoadRecords: με την συνάρτηση αυτή διαβάζονται μέσω της ReadRecord τα στοιχεία του Record 1 του Record File 1 και αποθηκεύονται με κατάλληλο τρόπο στις παραμέτρους dayin, weekday, yearday, nhour, nmin, yhour, ymin και timein. Οι αντιστοιχίες μεταξύ των bytes του record και των παραμέτρων αυτών έχουν εξηγηθεί στην ενότητα 6.2.5, στο τμήμα που αναφέρεται στο Record File 1. Αν η ανάγνωση του αρχείου αποτύχει η LoadRecords επιστρέφει τον κωδικό λάθους στο πρόγραμμα, αλλιώς επιστρέφει κωδικό επιτυχίας (0).
- UpdateRecords: ανανεώνει με κατάλληλο τρόπο τα στοιχεία του Record 1 του Record File 1 σύμφωνα με τις τιμές των παραμέτρων dayin, weekday, yearday, nhour, nmin, yhour, ymin και timein. Οι αντιστοιχίες μεταξύ των bytes του record και των παραμέτρων έχουν εξηγηθεί στην ενότητα 6.2.5, στο τμήμα που αναφέρεται στο Record File 1. Αν η ανανέωση του αρχείου αποτύχει η LoadRecords επιστρέφει τον κωδικό λάθους στο πρόγραμμα, αλλιώς επιστρέφει τον κωδικό επιτυχίας 0.
- Awardcount: η συνάρτηση αυτή αποδίδει στους μετρητές 1, 2, 3, 4 και 5 (Normhour, Normmins, Yperhour, Ypermins και Fifteens) τους πόντους που είναι αποθηκευμένοι στην μεταβλητή award (award[1], award[2], award[3], award[4] και award[5] αντίστοιχα). Οι αποδόσεις των πόντων γίνονται με τη χρήση της συνάρτησης Award. Αν σε κάποια απόδοση πόντων παρουσιαστεί σφάλμα η συνάρτηση επιστρέφει τη τιμή –8 στο πρόγραμμα, αλλιώς επιστρέφει τη τιμή 0.
- CalcWork: η διαδικασία αυτή υλοποιεί την πολιτική υπερωριών που αναφέρθηκε στην ενότητα 5.1.3.3. Υπολογίζει για το χρονικό διάστημα μεταξύ εισόδου και εξόδου του εργαζόμενου πόσες ώρες κανονικής εργασίας και πόσες υπερωριακής εργάστηκε καθώς επίσης και τους πόντους που πρέπει να προστεθούν σε κάθε μετρητή που χρησιμοποιείται σε αυτό το πρόγραμμα. Για να γίνει αυτό πρέπει να υπάρχει η γνώση των ήδη υπάρχουσων ωρών εργασίας του εργαζόμενου μέσα στην ίδια μέρα, πληροφορία την οποία περιέχουν οι παράμετροι nhour, nmin, yhour και ymin. Το χρονικό διάστημα μεταξύ εισόδου και εξόδου του εργαζόμενου περιέχεται στις παραμέτρους temph και tempm με τη μορφή ωρών και λεπτών αντίστοιχα που έχουν περάσει. Η παράμετρος time αντιστοιχεί στη μεταβλητή weekday που έχει αναφερθεί και σε προηγούμενη ενότητα, και χρησιμεύει για να δηλώσει αν η τρέχουσα μέρα εργασίας είναι μέσα σε Σαββατοκύριακο. Οι παράμετροι nhour,

nmin, yhour και ymin ανανεώνονται μετά τους υπολογισμούς με τις νέες τιμές ωρών και λεπτών εργασίας του εργαζόμενου μέσα στην ημέρα. Οι παράμετροι countnh, countnm, countyh, countym και countfif ενημερώνονται με τις νέες τιμές που θα πρέπει μετά τους υπολογισμούς να έχουν οι μετρητές 1, 2, 3, 4 και 5 και οι μεταβλητές award[1]…award[5] περιέχουν τον αριθμό των πόντων που θα πρέπει αντίστοιχα να προστεθούν στους μετρητές αυτούς για να συμπεριλάβουν τις νέες ώρες εργασίας που συμπλήρωσε ο εργαζόμενος. Οι πόντοι που θα πρέπει να προστεθούν σε κάθε μετρητή υπολογίζονται σύμφωνα με όσα αναφέρθηκαν στην ενότητα 6.2.4.

- Appli: η συνάρτηση αυτή καλείται από το κύριο πρόγραμμα για να υλοποιήσει το κύριο σώμα του προγράμματος Υπερωριών, αυτό δηλαδή που αφορά τον έλεγχο εισόδου και εξόδου του εργαζόμενου. Η συνάρτηση αυτή, την πρώτη φορά που καλείται, ανοίγει το αρχείο OVERLOGFILE.txt ως log αρχείο, ανοίγει κανάλι επικοινωνίας με τον reader, πιστοποιεί την ταυτότητα του διαχειριστή και έπειτα αναμένει είσοδο κάρτας στον reader. Σε κάθε εισαγωγή κάρτας πιστοποιεί την ταυτότητα του κατόχου της κάρτας, διαβάζει τα απαραίτητα στοιχεία από την κάρτα με τις LoadCounters και LoadRecords και έπειτα ελέγχει αν πρόκειται για είσοδο ή έξοδο του χρήστη τυπώνοντας τις κατάλληλες πληροφορίες (στην οθόνη και το αρχείο) και κάνοντας τις απαραίτητες ενημερώσεις στα αρχεία της κάρτας (UpdateRecords). Αν πρόκειται για έξοδο υπολογίζει με την CalcWork τις ώρες και το είδος εργασίας του εργαζόμενου για το διάστημα που μεσολάβησε από την τελευταία είσοδο, τυπώνει πληροφορίες στην οθόνη και το αρχείο και αποθηκεύει τους σωστούς πόντους στους μετρητές με χρήση της συνάρτησης Award και της μεταβλητής award. Κατά την έξοδο υποστηρίζεται και η περίπτωση που ο εργαζόμενος βγαίνει από το χώρο εργασίας διαφορετική μέρα από αυτή που μπήκε και τότε ο διαχειριστής καλείται να συμπληρώσει χειροκίνητα τις ώρες εργασίας του εργαζόμενου, αν είναι παρών. Μετά την χειροκίνητη συμπλήρωση των ωρών εργασίας καλείται και πάλι η CalcWork και ακολουθούν οι συνηθισμένες διαδικασίες. Τέλος περιμένει αφαίρεση της κάρτας από τον reader. Ανάλογα με τα σφάλματα που μπορεί να προκληθούν επιστρέφεται στο κύριο πρόγραμμα ειδικός κωδικός λάθους αλλιώς επιστρέφεται η τιμή 0. Η συνάρτηση αυτή καλείται συνέχεια από το κύριο πρόγραμμα μέσα σε ένα βρόχο.

6.3.5 Πρόγραμμα Redemption (Εξαργύρωση)

Το πρόγραμμα Redemption εξυπηρετεί όπως έχει ειπωθεί λειτουργίες δύο εφαρμογών, της εφαρμογής μισθοδοσίας και της εφαρμογής ηλεκτρονικού πορτοφολιού. Η δομή του και οι λειτουργίες που υποστηρίζει έχουν εξεταστεί αναλυτικά στις ενότητες 5.1.3.2 και 5.2.3.2. Οι ακριβείς συναρτήσεις που έχουν χρησιμοποιηθεί παρατίθενται παρακάτω. Για τις παραμέτρους που χρησιμοποιούνται ισχύουν οι επισημάνσεις που αναφέρθηκαν στην ενότητα 6.3.2.1. Συμπληρωματικά δηλώνεται σε αυτό το πρόγραμμα η γενική μεταβλητή award που χρησιμοποιείται για να αποθηκεύει τους πόντους που πρέπει να προστεθούν στους μετρητές, η γενική μεταβλητή over για τα ποσά προκαταβολής που πρέπει να περνούν στον μετρητή Advancep και ο περιγραφέας αρχείου clf για το αρχείο που θα χρησιμοποιεί ο εργαζόμενος ως απόδειξη για τις συναλλαγές του. Όλο το πρόγραμμα βρίσκεται στο Παράρτημα Α, στις σελίδες 41 - 56.

6.3.5.1 Συναρτήσεις - Διαδικασίες

- Producekey: είναι ίδια με την Producekey του προγράμματος Personover. Χρησιμεύει για να επαληθεύει το πρόγραμμα τη ταυτότητα του διαχειριστή μισθοδοσίας. Δημιουργεί ένα κρυπτόγραμμα το οποίο πρέπει να ταιριάζει με το μυστικό κλειδί του διαχειριστή.

- LoadCounters: διαβάζει με χρήση της συνάρτησης ReadCountBalance τις τιμές των μετρητών 1, 2, 3, 4, 5 και 6 (Normhour, Normmins, Yperhour, Ypermins, Fifteens και Advancep) και τις αποθηκεύει αντίστοιχα στις παραμέτρους countnh, countnm, countyh, countym, countfif και countadv. Αν έστω και μία από τις αναγνώσεις των μετρητών αποτύχει, ο κωδικός σφάλματος επιστρέφεται στο πρόγραμμα. Σε περίπτωση επιτυχίας η συνάρτηση επιστρέφει τη τιμή 0.
- WorkInfo: με την συνάρτηση αυτή διαβάζονται μέσω της ReadRecord τα Records 10, 11 και 12 του Record File 2 τα οποία περιέχουν το ωρομίσθιο του εργαζόμενου, τις ημέρες αδείας που δικαιούται μέχρι το τέλος του χρόνου και το όριο των υπερωριών που μπορεί να κάνει μέσα σε ένα μήνα. Τα περιεχόμενα των records αυτών αποθηκεύονται με κατάλληλο τρόπο στις παραμέτρους salary, holim και ylim. Αν κάποια από τις αναγνώσεις των αρχείων αποτύχει η WorkInfo επιστρέφει τον κωδικό λάθους στο πρόγραμμα, αλλιώς επιστρέφει κωδικό επιτυχίας (0).
- PersInfo: η συνάρτηση αυτή διαβάζει από το Record File 2 τις προσωπικές πληροφορίες του εργαζόμενου, δηλαδή το ονοματεπώνυμο, το όνομα πατρός, τη διεύθυνση διαμονής, την ημερομηνία γέννησης, τον αριθμό ταυτότητας και τον αριθμό των ετών που ο εργαζόμενος έχει στη συγκεκριμένη εργασία. Τα στοιχεία αυτά εκτυπώνονται με κατάλληλη στοίχιση στην οθόνη και στο αρχείο που αφορά τον εργαζόμενο. Για την ανάγνωση των εγγραφών χρησιμοποιείται η ReadRecord. Αν κάποια από τις αναγνώσεις αποτύχει, ο κωδικός σφάλματος που επιστρέφει περνάει από την PersInfo στο πρόγραμμα που την έχει καλέσει. Αν πάλι όλες οι αναγνώσεις εκτελεστούν επιτυχώς, η συνάρτηση επιστρέφει στο πρόγραμμα το 0.
- Awardcountm: με την συνάρτηση αυτή εκτελείται η λειτουργία της χειροκίνητης ενημέρωσης μετρητών της κάρτας που αναφέραμε στην ενότητα 5.1.3.4. Οι μετρητές που ενημερώνονται με τους πόντους που εισάγει ο διαχειριστής είναι οι 5 που αφορούν το πρόγραμμα Overtime, δηλαδή οι Normhour, Normmins, Yperhour, Ypermins και Fifteens. Χρησιμοποιούνται κυρίως οι συναρτήσεις Award και debstrint. Αν κάποια απόδοση πόντων αποτύχει, η Awardcountm διακόπτεται και επιστρέφει στο πρόγραμμα τον κωδικό λάθους –17 (ο οποίος χειρίζεται μετά από το κύριο πρόγραμμα) ενώ αν όλες οι συναρτήσεις εκτελεστούν επιτυχώς, επιστρέφεται στο πρόγραμμα η τιμή επιτυχίας 0.
- AwPurse: πραγματοποιεί την απόδοση αξίας – πόντων στο ηλεκτρονικό πορτοφόλι, σύμφωνα με όσα έχουν αναφερθεί στην ενότητα 5.2.3.2 όπου αναλύεται η παρούσα λειτουργία. Η συνάρτηση διαβάζει το περιεχόμενο του πορτοφολιού, εκτελεί κάποιους ελέγχους και αν επιτρέπεται απόδοση αξίας στο πορτοφόλι προτρέπει τον διαχειριστή για την εισαγωγή του ποσού. Το ποσό ελέγχεται για την εγκυρότητά του και με τις κατάλληλες μετατροπές (βλ.ενότητα 6.2.4, σελ.83) πραγματοποιεί την προσθήκη αξίας στον μετρητή Elepurse και τυπώνει τις πληροφορίες της συναλλαγής στην οθόνη και τα αρχεία file και clf. Αν παρουσιαστεί σφάλμα στην εκτέλεση της συνάρτησης, αυτό επιστρέφεται στο πρόγραμμα, αλλιώς η συνάρτηση επιστρέφει τον κωδικό επιτυχίας 0.
- RedPurse: η συνάρτηση υλοποιεί την εξαργύρωση ολόκληρου του ηλεκτρονικού πορτοφολιού του εργαζόμενου. Διαβάζει αρχικά το περιεχόμενό του και αν δεν είναι μηδενικό περνάει στην αφαίρεση όλης της αξίας του. Στο τέλος πληροφορεί για το ποσό που αφαιρέθηκε από τον μετρητή Elepurse (μετατρέπει τους πόντους που αφαιρέθηκαν σε αντίστοιχη χρηματική αξία). Οι πληροφορίες αυτές τυπώνονται στην οθόνη και τα δύο αρχεία. Αν παρουσιαστεί σφάλμα, η συνάρτηση επιστρέφει στο πρόγραμμα τον κωδικό του αλλιώς επιστρέφει τη τιμή 0.
- Payment: η συνάρτηση αυτή ευθύνεται για τον υπολογισμό του τελικού μισθού που θα λάβει ο εργαζόμενος όταν επιθυμήσει την εξαργύρωση των ωρών εργασίας του στο τέλος του μήνα. Αρχικά διαβάζει το περιεχόμενο των μετρητών που την αφορούν με την συνάρτηση LoadCounters ενώ με την WorkInfo διαβάζει απαραίτητες πληροφορίες που αφορούν τον εργαζόμενο. Επίσης διαβάζει το Record 1 του Record File 6 που περιέχει την πληροφορία αν στον μετρητή Advancep είναι

αποθηκευμένο ποσό που έχει προέλθει από υπερωριακή προκαταβολή (μεταβλητή "flag"). Ανάλογα με τις ώρες εργασίας που διαβάζει από τους μετρητές, με τα ποσά προκαταβολής που βρίσκονται στον μετρητή Advancep και με τις ημέρες αδείας που έχει χρησιμοποιήσει ο εργαζόμενος, υπολογίζει και τυπώνει σε οθόνη και αρχείο το τελικό μισθό που δικαιούται ο εργαζόμενος. Στην περίπτωση υπέρβασης των ωρών υπερωρίας ισχύουν οι λειτουργίες που έχουν αναλυθεί στην ενότητα 5.1.3.4. Η συνάρτηση Payment μπορεί να κληθεί είτε για απλό υπολογισμό του μισθού του εργαζόμενου για τις ώρες εργασίας που είναι καταγεγραμμένες στην κάρτα του, είτε για υπολογισμό και εξαργύρωση του μισθού του, είτε τέλος για εκτέλεση της λειτουργίας κανονικής προκαταβολής. Στις δύο πρώτες περιπτώσεις η συνάρτηση επιστρέφει τον συνολικό μισθό που έχει υπολογίσει για τον εργαζόμενο. Στη τρίτη περίπτωση όμως επιστρέφει το μέγιστο ποσό που μπορεί να λάβει ο εργαζόμενος ως προκαταβολή. Το ποσό αυτό εξαρτάται από την τυχούσα προκαταβολή που έχει ήδη πάρει και από τις ώρες κανονικής εργασίας που έχει συμπληρώσει μέχρι τότε μέσα στο μήνα. Κατά τον υπολογισμό του συνολικού μισθού του εργαζόμενου χρησιμοποιούνται μεταβλητές όπως οι pay, payn, payy, pay5, payh και paya που αποθηκεύουν τα ποσά που αντιστοιχούν στο σύνολο της εργασίας του, στην κανονική εργασία, στην υπερωριακή εργασία, στην υπερωριακή εργασία που αμείβεται με προσαύξηση 50%, στις ημέρες αδείας και στο ποσό προκαταβολής που έχει τυχόν λάβει. Επίσης σημαντική είναι η χρήση της γενικής μεταβλητής over που πληροφορεί το πρόγραμμα για το ποσό υπερωριακής προκαταβολής το οποίο έχει λάβει ο εργαζόμενος και το οποίο πρέπει να προστεθεί στον μετρητή Advancep κατά τη διαδικασία αρχικοποίησης των μετρητών μετά την εξαργύρωση του μισθού του κατόχου της κάρτας.

- Advance: υλοποιεί την λειτουργία κανονικής προκαταβολής για τον εργαζόμενο. Η συνάρτηση καλεί την Payment για να πληροφορηθεί για το μέγιστο ποσό που μπορεί να λάβει ο εργαζόμενος ως προκαταβολή και αν αυτό δεν είναι μηδενικό, προτρέπει τον διαχειριστή για την εισαγωγή ενός ποσού, ελέγχει την ορθότητά του και τέλος προσθέτει την αξία αυτή στον μετρητή Advancep (για να αφαιρεθεί μετά από τον τελικό μισθό του εργαζόμενου) και τυπώνει τα στοιχεία της συναλλαγής. Αν η συνάρτηση εκτελεστεί επιτυχώς, επιστρέφει την τιμή 0, αλλιώς επιστρέφει τον κωδικό σφάλματος, όπου αυτός έχει προκύψει.
- ResetCard: η συνάρτηση ResetCard πραγματοποιεί την απαραίτητη αρχικοποίηση – μηδενισμό αρχείων κάρτας που πρέπει να εκτελείται πάντα μετά από τη λειτουργία υπολογισμού – εξαργύρωσης του μισθού του εργαζόμενου. Συγκεκριμένα, αρχικοποιούνται τα περιεχόμενα του Record 1 του Record File 1 (dayin, weekday, yearday, nhour, nmin, yhour, ymin και timein), μηδενίζονται οι μετρητές Normhour, Normmins, Yperhour, Ypermins και Fifteens και τέλος ο μετρητής Advancep λαμβάνει τη σωστή τιμή ανάλογα με την μεταβλητή over. Αν προκύψει κάποιο σφάλμα, η συνάρτηση επιστρέφει κατάλληλο κωδικό λάθους αλλιώς επιστρέφει 0.
- Apbody: αποτελεί το κύριο μέρος του προγράμματος. Ανοίγει κανάλι επικοινωνίας με reader και κάρτα, ανοίγει δύο αρχεία, το PAYLOGFILE ως log αρχείο και το RECEIPT για να έχει απόδειξη ο εργαζόμενος για τις συναλλαγές που έχει πραγματοποιήσει μέσω του προγράμματος Redemption (εξαργυρώσεις μισθού, προσθήκες αξίας στο ηλεκτρονικό πορτοφόλι, προκαταβολές κ.ά.), πιστοποιεί την ταυτότητα του διαχειριστή και του κατόχου της κάρτας και τυπώνει τα στοιχεία του κατόχου. Έπειτα η συνάρτηση δίνει στον διαχειριστή την δυνατότητα να διαλέξει ποια λειτουργία του προγράμματος θα εκτελέσει και ανάλογα με την επιλογή αυτή καλεί την κατάλληλη συνάρτηση από αυτές που αναφέραμε παραπάνω. Αν κάποια συνάρτηση από αυτές που καλεί επιστρέψει κωδικό σφάλματος, η Apbody τον αντιστοιχίζει με ειδικό κωδικό που επιστρέφει στο κύριο πρόγραμμα. Το κύριο πρόγραμμα με τη σειρά του αξιολογεί τον κωδικό αυτό, τυπώνει τα κατάλληλα μηνύματα σε οθόνη και αρχεία και εκτελεί ή έξοδο ή επανάκληση της Apbody. Αν η Apbody εκτελεστεί επιτυχώς επιστρέφει στο κύριο πρόγραμμα τη τιμή 0.

6.3.6 Πρόγραμμα ServPoint (Σημείο Εξυπηρέτησης)

Το πρόγραμμα SevrPoint εξυπηρετεί την εφαρμογή ηλεκτρονικού πορτοφολιού και προγράμματος εμπιστοσύνης. Η δομή που το χαρακτηρίζει και οι λειτουργίες που υποστηρίζει έχουν εξεταστεί στην ενότητα 5.2.3.3 και στην ενότητα 5.2.3.4. Οι συναρτήσεις που χρησιμοποιεί δίνονται ακολούθως. Ισχύουν οι γνωστές από την ενότητα 6.3.2.1 επισημάνσεις παραμέτρων. Ο κώδικας του προγράμματος παρατίθεται ολόκληρος στις σελίδες 57 - 62 του Παραρτήματος Α.

6.3.6.1 Συναρτήσεις - Διαδικασίες

- UseRule: η συνάρτηση αυτή εκτελεί τον κανόνα που βρίσκεται στο αρχείο κανόνα το οποίο δηλώνει η παράμετρος SFI, με ποσό συναλλαγής το περιεχόμενο της παραμέτρου Amount. Χρησιμοποιεί την συνάρτηση βιβλιοθήκης G_UseRuleEnh. Σε περίπτωση σφάλματος τυπώνει τον κωδικό λάθους και τον επιστρέφει στο πρόγραμμα ενώ αν είναι επιτυχής επιστρέφει το 0. Οι ιδιότητες του κανόνα που υπάρχει στην κάρτα της εφαρμογής περιγράφονται στην ενότητα 6.2.6.
- Producekey: ίδια με την Producekey των προγραμμάτων Personover, Overtime και Redemption. Χρησιμεύει για να επαληθεύει το πρόγραμμα τη ταυτότητα του διαχειριστή πωλήσεων.
- MakeBuy: πραγματοποιεί μία αγορά (εκ μέρους του κατόχου της κάρτας) με χρήση του ηλεκτρονικού πορτοφολιού. Αρχικά διαβάζει το περιεχόμενο του μετρητή Elepurse και αν υπάρχει διαθέσιμο ποσό προτρέπει τον διαχειριστή να εισάγει το ποσό της αγοράς. Ελέγχει την εγκυρότητα του ποσού και τη διαθεσιμότητά του σε σχέση με το ηλεκτρονικό πορτοφόλι και αν όλες οι συνθήκες ικανοποιηθούν καλεί τον κανόνα 1, ο οποίος αφαιρεί με κατάλληλο τρόπο την αξία της αγοράς από τον μετρητή Elepurse και προσθέτει τους πόντους που κερδίζει ο κάτοχος μέσα στα πλαίσια του προγράμματος εμπιστοσύνης στον μετρητή Loyaltyc. Αν παρουσιαστεί κάποιο σφάλμα η MakeBuy επιστρέφει ειδικό κωδικό λάθους, αλλιώς τυπώνει τα στοιχεία της συναλλαγής και επιστρέφει τη τιμή 0.
- RedeemPoints: η συνάρτηση αυτή εκτελεί την αφαίρεση – εξαργύρωση πόντων από τον μετρητή Loyaltyc. Αρχικά διαβάζει το περιεχόμενο του μετρητή (με την LoadLoyal) και αν υπάρχουν διαθέσιμοι πόντοι, προτρέπει τον διαχειριστή να εισάγει τον αριθμό των πόντων που ο κάτοχος επιθυμεί να εξαργυρώσει. Η συνάρτηση ελέγχει την εγκυρότητα του αριθμού, προχωρά στην αφαίρεση των πόντων, τυπώνει τις πληροφορίες της συναλλαγής αυτής και επιστρέφει τη τιμή 0. Αν έχει παρουσιαστεί κάποιο σφάλμα, η συνάρτηση επιστρέφει τον κωδικό του στο πρόγραμμα που την έχει καλέσει.
- Servpoint: η συνάρτηση Servpoint λειτουργεί όπως και η Apbody στην προηγούμενη ενότητα. Έτσι, ανοίγει κανάλι επικοινωνίας με reader και κάρτα, ανοίγει το αρχείο SERVLOGFILE ως log αρχείο και πιστοποιεί την ταυτότητα του διαχειριστή και του κατόχου της κάρτας. Στη συνέχεια δίνει στον διαχειριστή πωλήσεων την δυνατότητα να διαλέξει ποια λειτουργία του προγράμματος θα εκτελέσει και ανάλογα με την επιλογή καλεί την κατάλληλη συνάρτηση από τις προαναφερθείσες. Αν κάποια συνάρτηση επιστρέψει κωδικό σφάλματος, η Servpoint τον αντιστοιχίζει με ειδικό κωδικό που επιστρέφει στο κύριο πρόγραμμα ενώ αν η Servpoint εκτελεστεί με επιτυχία επιστρέφει στο κύριο πρόγραμμα τη τιμή 0.

6.3.7 Πρόγραμμα TicketCenter (Κέντρο Εισιτηρίων)

Η δομή και οι λειτουργίες του προγράμματος αυτού έχουν εξεταστεί στις ενότητες 5.3.3.1 και 5.3.3.2. Το πρόγραμμα αυτό περιέχει βασικά 2 συναρτήσεις που παρατίθενται ακολούθως. Ισχύουν και εδώ οι γνωστές από την ενότητα 6.3.2.1 επισημάνσεις παραμέτρων. Ολόκληρο το πρόγραμμα παρατίθεται στις σελίδες 63 - 66 του Παραρτήματος Α.

- Producekey: ίδια με την Producekey των προγραμμάτων Personover, Overtime Redemption, ServPoint. Χρησιμεύει για να επαληθεύει το πρόγραμμα τη ταυτότητα του διαχειριστή εισιτηρίων.
- Appli: η συνάρτηση αυτή καλείται από το κύριο πρόγραμμα για να υλοποιήσει το κύριο σώμα του προγράμματος Κέντρο Εισιτηρίων που αφορά την αποθήκευση αξίας εισιτηρίων στην κάρτα. Όταν ξεκινάει, ανοίγει το αρχείο OVERLOGFILE ως log αρχείο, ανοίγει κανάλι επικοινωνίας με τον reader, πιστοποιεί την ταυτότητα του διαχειριστή και έπειτα αναμένει είσοδο κάρτας στον reader. Με την εισαγωγή κάρτας καλεί τον διαχειριστή εισιτηρίων να επιλέξει ποια από τις δύο λειτουργίες του προγράμματος θα εκτελέσει. Αν ο διαχειριστής επιλέξει την αρχικοποίηση της πληροφορίας του είδους εισιτηρίου που χρησιμοποιεί ο κάτοχος, ενημερώνει με την UpdateRecSM το Record 1 του Record File 7 (μεταβλητή "half", κανονικό ή μειωμένο εισιτήριο). Αν ο διαχειριστής επιλέξει την πρόσθεση αξίας εισιτηρίων, αυτή εκτελείται σύμφωνα με όσα έχουν αναφερθεί στις ενότητες 5.3.3.2 (Λειτουργίες προγράμματος "Κέντρο Εισιτηρίων") και 6.2.4, στο τμήμα που αναφέρεται στον μετρητή Ticketin (Counter 9). Η μεταβλητή trans χρησιμεύει για να υποδεικνύει το είδος της συγκοινωνίας για το οποίο θα αγοραστούν τα εισιτήρια. Η μεταβλητή num αποθηκεύει το νούμερο των εισιτηρίων που θα αγοράσει ο κάτοχος της κάρτας. Ανάλογα με τα σφάλματα που μπορεί να προκληθούν επιστρέφεται στο κύριο πρόγραμμα ειδικός κωδικός λάθους αλλιώς επιστρέφεται η τιμή 0.

6.3.8 Πρόγραμμα Transportation (Μετακίνηση)

Το πρόγραμμα Transportation εξυπηρετεί την εφαρμογή εισιτηρίων και είναι το πιο απλό σε μορφή και λειτουργίες (ενότητα 5.3.3.3). Ολόκληρος ο κώδικας του προγράμματος βρίσκεται στις σελίδες 67 - 68 του Παραρτήματος Α.
Το πρόγραμμα χρησιμοποιεί μία μόνο συνάρτηση, την Appli, την οποία το πρόγραμμα εκτελεί συνέχεια μέσα σε βρόχο. Η συνάρτηση Appli ανοίγει το αρχείο TICKET ως απόδειξη για τον κάτοχο της κάρτας, ανοίγει κανάλι επικοινωνίας με τον reader και αναμένει είσοδο κάρτας στον reader. Όποτε εισάγεται κάρτα στον reader (προφανώς για "επικύρωση εισιτηρίου") η συνάρτηση διαβάζει το Record 1 του Record File 7 (μεταβλητή "half") όπου φαίνεται αν πρόκειται για κανονικό ή μειωμένο εισιτήριο και έπειτα διαβάζει την αποθηκευμένη αξία στον μετρητή Ticketin. Αν δεν υπάρχει κατάλληλη διαθέσιμη αξία εισιτηρίου στον μετρητή τυπώνει στην οθόνη ανάλογο μήνυμα και επιστρέφει στην αρχή. Αν πάλι υπάρχει διαθέσιμη αξία στην κάρτα, η συνάρτηση ανάλογα με το είδος του μέσου μεταφοράς στο οποίο τρέχει (μεταβλητή trans) αφαιρεί την κατάλληλη αξία από τον counter Ticketin και τυπώνει την ώρα, την ημερομηνία και το ποσό του εισιτηρίου στο αρχείο. Αν μετά την "ακύρωση" του εισιτηρίου η διαθέσιμη αξία στον μετρητή αντιστοιχεί σε ένα εισιτήριο, η συνάρτηση τυπώνει αντίστοιχο μήνυμα ενώ αν η αξία δεν αρκεί για κανένα πλέον εισιτήριο πληροφορεί τον κάτοχο ότι δεν έχει άλλα εισιτήρια στην κάρτα του για το συγκεκριμένο μέσο μεταφοράς. Αν προκληθεί κάποιο σφάλμα , για παράδειγμα αν ο κάτοχός της την αφαιρέσει πριν γίνει η αφαίρεση της αξίας του εισιτηρίου, επιστρέφεται στο κύριο πρόγραμμα κωδικός λάθους, βάση του οποίου τυπώνεται κατάλληλο μήνυμα στον κάτοχο. Έπειτα το κύριο πρόγραμμα καλεί ξανά την Appli.

7

Αποτελέσματα

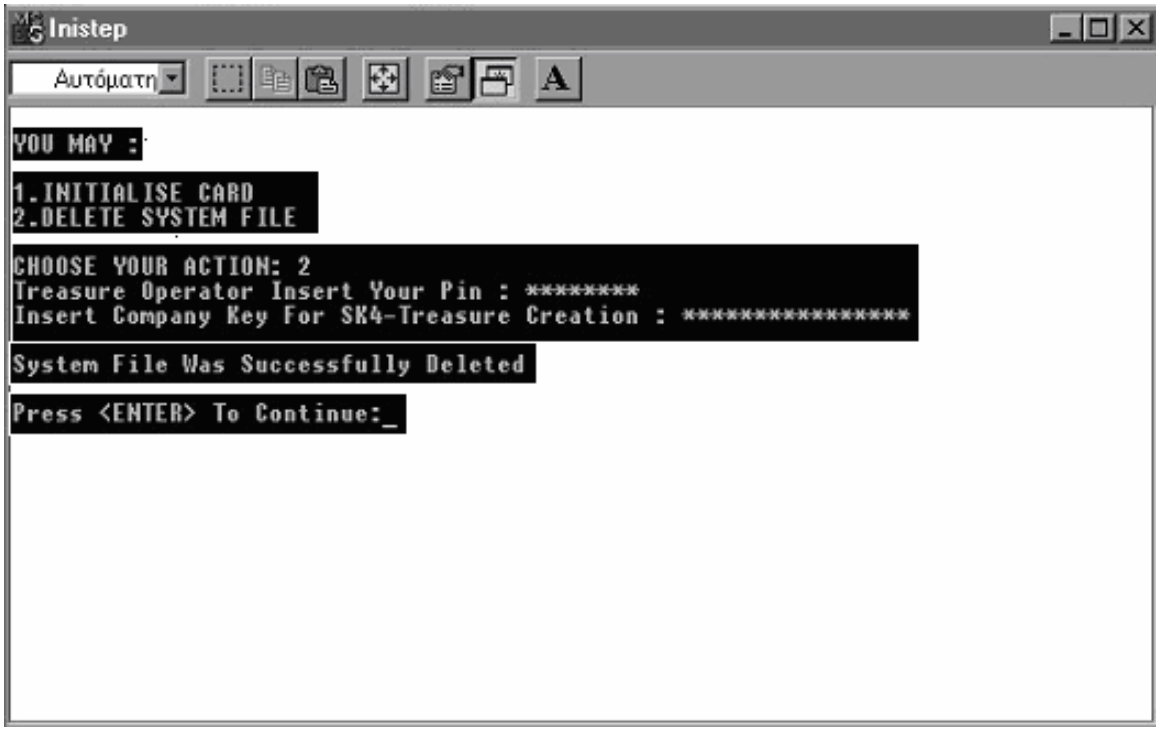
Σε αυτό το κεφάλαιο θα παρουσιάσουμε κάποια στιγμιότυπα πραγματικής εφαρμογής των προγραμμάτων που αναλύσαμε στα προηγούμενα 2 κεφάλαια, για μία καλύτερη κατανόηση των περιεχομένων και της μορφής τους.
Για τις ανάγκες των εφαρμογών, δημιουργήθηκαν 3 sample cards. Η διαδικασία που ακολουθήθηκε για τη δημιουργία τους είναι μία και αποτελείται από τα προγράμματα Inistep και Personover, όπως έχει αναφερθεί και παραπάνω.

7.1 Αρχικοποίηση – Προσωποποίηση Sample Cards

Θεωρήθηκαν 3 sample εργαζόμενοι – κάτοχοι καρτών, με τα ακόλουθα στοιχεία:

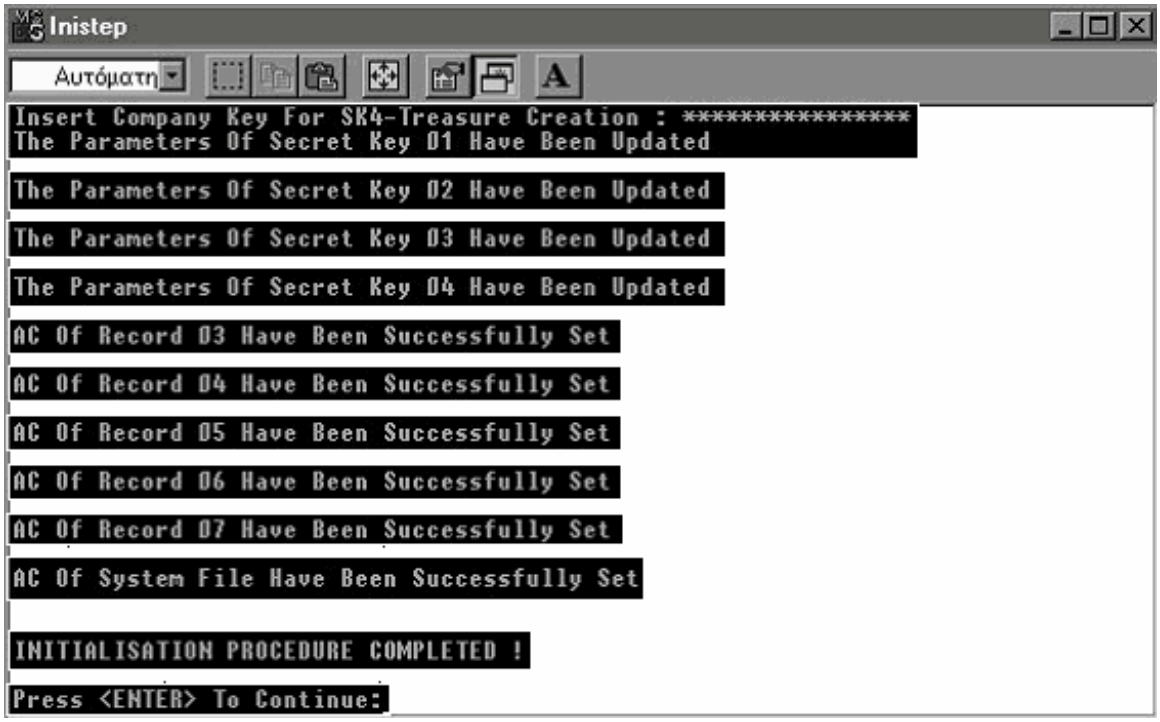
- Νατάσσα Παπαδοπούλου του Φιλίππου (ΚΑΤΟΧΟΣ 1)
  Δ/νση: Στρ.Παπάγου 158-162, Ηλιούπολη, Αθήνα
  Ημ/νία Γέννησης: 23-5-1964
  Αρ.Ταυτότητας: Α 121445
  Ωρομίσθιο: 12.5 €/ώρα
  Άδεια: 25 ημέρες
  Όριο Υπερωριών: 14 ώρες
  Χρόνια Εργασίας: 3
- Γιάννης Τασόπουλος του Πέτρου (ΚΑΤΟΧΟΣ 2)
  Δ/νση: Ευτυχίδου 12, Παγκράτι, Αθήνα
  Ημ/νία Γέννησης: 2-10-1972
  Αρ.Ταυτότητας: Ρ 121987
  Ωρομίσθιο: 31.2 €/ώρα
  Άδεια: 21 ημέρες
  Όριο Υπερωριών: 27 ώρες
  Χρόνια Εργασίας: 4
- Μάριος Κατσάνος του Κωνσταντίνου (ΚΑΤΟΧΟΣ 3)
  Δ/νση: Απόλλωνος 45, Κηφισιά, Αθήνα
  Ημ/νία Γέννησης: 24-12-1959
  Αρ.Ταυτότητας: Χ 223112
  Ωρομίσθιο: 9.21 €/ώρα
  Άδεια: 23 ημέρες
  Όριο Υπερωριών: 24 ώρες
  Χρόνια Εργασίας: 8

Θα εξετάσουμε την δημιουργία της κάρτας της κατόχου 1. Θεωρήθηκε πως η κάρτα δεν ήταν κενή και για το λόγο αυτό έπρεπε το πρόγραμμα Inistep πρώτα να τη διαγράψει και μετά να την αρχικοποιήσει. Ένα στιγμιότυπο αυτής της διαδικασίας φαίνεται στο Σχήμα 7.1.1 που ακολουθεί. Ο διαχειριστής θησαυροφυλακίου έχει εισάγει τους σωστούς κωδικούς και έχει μόλις διαγράψει το αρχείο συστήματος της κάρτας.



Σχήμα 7.1.1 – Διαγραφή αρχείου συστήματος

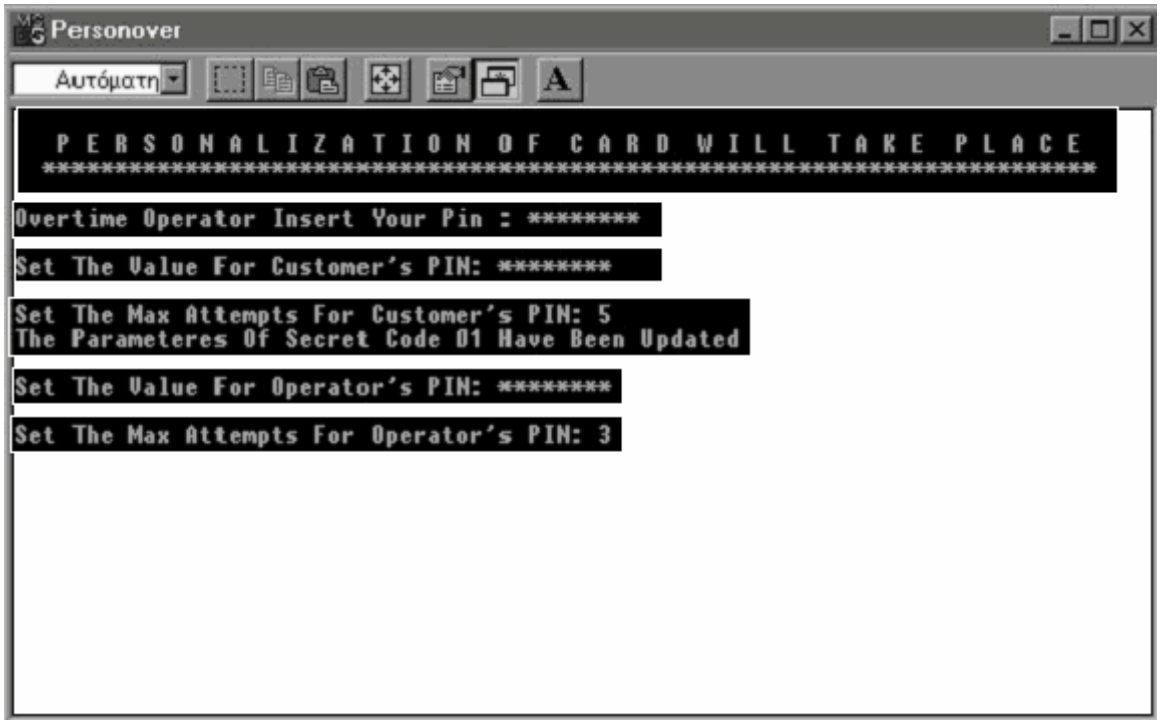
Στη συνέχεια έγινε αρχικοποίηση της κάρτας και η διαδικασία τερμάτισε κανονικά, έχοντας ρυθμίσει όλες τις παραμέτρους. Στο στιγμιότυπο που ακολουθεί φαίνονται τα τελευταία μηνύματα που τύπωσε το πρόγραμμα μαζί με το μήνυμα επιτυχούς ολοκλήρωσης της διαδικασίας.



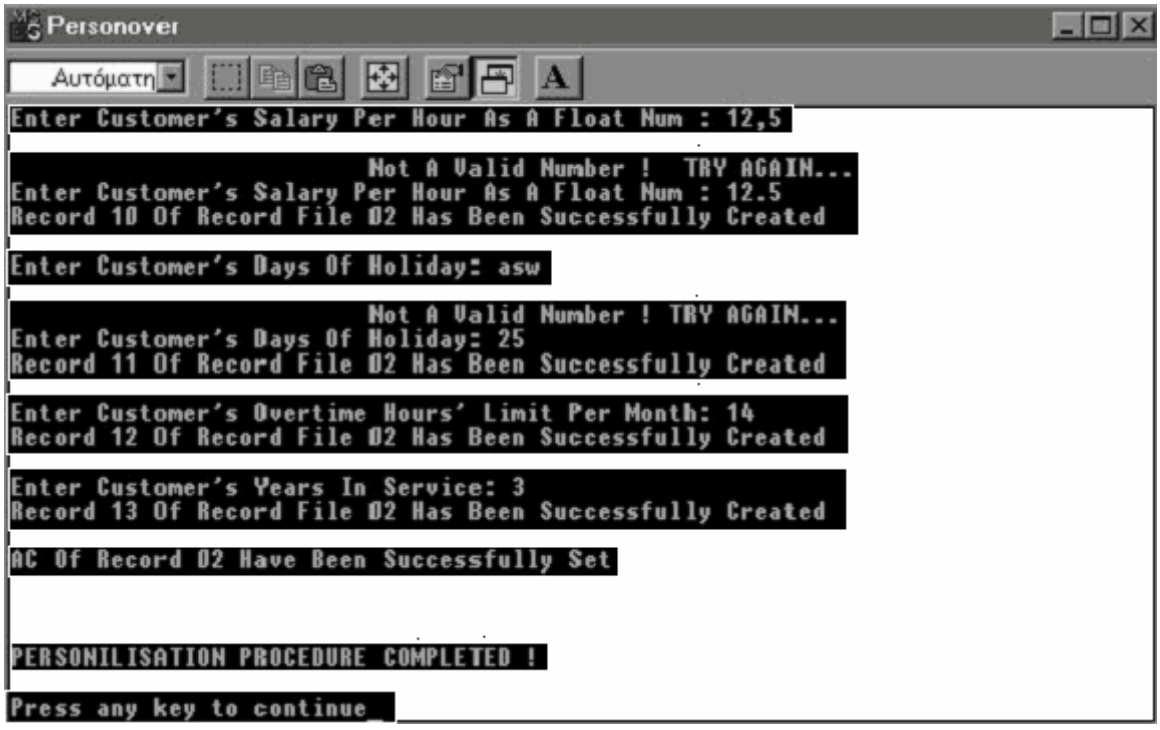
Σχήμα 7.1.2 – Τέλος διαδικασίας αρχικοποίησης

Έπειτα ακολούθησε η διαδικασία προσωποποίησης κάθε κάρτας με τη χρήση του προγράμματος Personover. Στα δύο στιγμιότυπα που ακολουθούν φαίνεται το στάδιο

εκκίνησης του προγράμματος, όπου ο διαχειριστής έχει εισάγει τον σωστό κωδικό και ρυθμίζει τις παραμέτρους των δύο μυστικών αρχείων της κάρτας, καθώς και τα τελευταία στάδια εισαγωγής δεδομένων όπου ο διαχειριστής συμπληρώνει τα προσωπικά στοιχεία του κατόχου 1. Συγκεκριμένα, στο 2ο στιγμιότυπο φαίνεται το σημείο όπου ζητείται η εισαγωγή του ωρομισθίου του εργαζόμενου και επειδή δεν χρησιμοποιείται η τελεία ως διαχωριστικό του δεκαδικού μέρους, δεν θεωρείται έγκυρο. Κάτι αντίστοιχο συμβαίνει και με τη συμπλήρωση των ημερών αδείας όπου αντί για νούμερο έχουν συμπληρωθεί γράμματα. Τέλος φαίνεται το μήνυμα επιτυχούς ολοκλήρωσης της διαδικασίας προσωποποίησης.



Σχήμα 7.1.3 – Εκκίνηση προγράμματος Personover



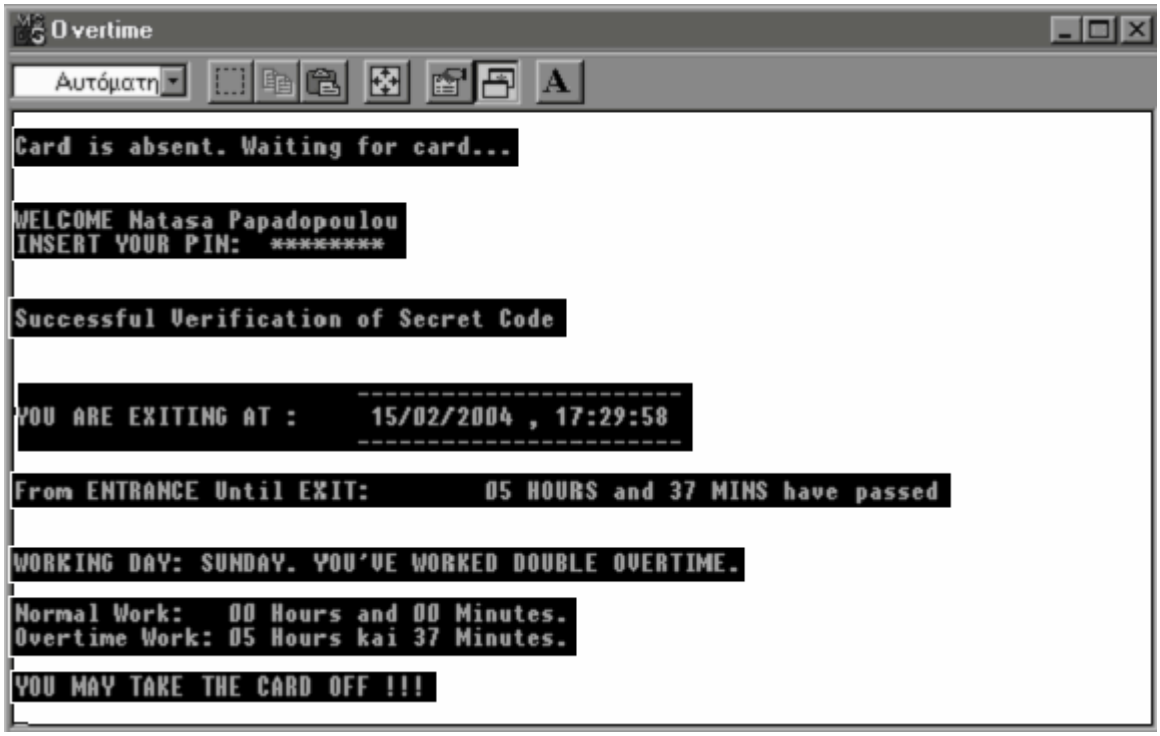
Σχήμα 7.1.4 – Τερματισμός προγράμματος Personover

7.2 Εφαρμογή Μισθοδοσίας

Στη διαδικασία εξομοίωσης της λειτουργίας της εφαρμογής μισθοδοσίας με τις sample cards, θέσαμε για λόγους ταχύτητας το πρόγραμμα να μετράει τα λεπτά ως ώρες και τα δευτερόλεπτα ως λεπτά.
Δηλαδή αν από την είσοδο ως την έξοδο ενός εργαζόμενου έχουν περάσει 9 λεπτά και 42 δευτερόλεπτα, το πρόγραμμα θεωρεί ότι έχουν περάσει 9 ώρες και 42 λεπτά.
Ακολούθως παρατίθεται το στιγμιότυπο της εισόδου της κατόχου 1 στο χώρο εργασίας. Το πρόγραμμα εκκίνησε εκείνη τη στιγμή, ο διαχειριστής εισήγαγε τον σωστό κωδικό και η κάτοχος χρειάστηκε δύο προσπάθειες για να εισάγει σωστά τον δικό της κωδικό.

Σχήμα 7.2.1 – Εκκίνηση προγράμματος Overtime και είσοδος χρήστη

Στο Σχήμα 7.2.2 που ακολουθεί, φαίνεται το στιγμιότυπο εξόδου της κατόχου 1 από το χώρο εργασίας. Επειδή η μέρα εισόδου – εξόδου είναι Κυριακή, το πρόγραμμα ειδοποιεί την κάτοχο ότι έχει δουλέψει διπλή υπερωρία και τυπώνει τη χρονική διαφορά μεταξύ εισόδου και εξόδου, το είδος της εργασίας μέσα σ' αυτό το διάστημα καθώς και το σύνολο και το είδος των ωρών εργασίας της μέσα στην ίδια ημέρα.



Σχήμα 7.2.2 – Έξοδος χρήστη

Το περιεχόμενο του αρχείου OVERLOGFILE παρατίθεται ακολούθως για να παρέχει μία εικόνα των υπολογισμών που έγιναν στην περίπτωση που περιγράφηκε:

WELCOME Natasa Papadopoulou
G_VerifySecretCode Error -4504!
YOU ARE ENTERING AT: 15/02/2004, 17:24:21

WELCOME Natasa Papadopoulou
YOU ARE EXITING AT: 15/02/2004, 17:29:58

From ENTRANCE Until EXIT: 05 HOURS and 37 MINS have passed
IT IS SUNDAY, YOU'VE WORKED DOUBLE OVERTIME.

Normal Work: 00 Hours and 00 Minutes.
Overtime Work: 05 Hours and 37 Minutes.
NORMHOUR=00, NORMMINS=00, YPERHOUR=150, YPERMINS=1110, FIFTEENS=00

0 Points Awarded To Counter 01. New Balance of Counter 01: 0 points!
0 Points Awarded To Counter 02. New Balance of Counter 02: 0 points!
150 Points Awarded To Counter 03. New Balance of Counter 03: 150 points!
1110 Points Awarded To Counter 04. New Balance of Counter 04: 1110 points!
0 Points Awarded To Counter 05. New Balance of Counter 05: 0 points!

Αφού μετρούνται τα λεπτά ως ώρες και τα δευτερόλεπτα ως λεπτά, από τις 17:24:21 ως τις 17:29:58 έχουν περάσει 5 ώρες και 37 λεπτά. Επειδή η εργασία της κατόχου 1 είναι διπλή υπερωρία θεωρείται ότι έχει εργαστεί 10 ώρες και 74 λεπτά υπερωρίας και έτσι στον μετρητή Yperhour θα αποδοθούν 10 • 15 = 150 πόντοι ενώ στον Ypermins θα αποδοθούν 74 • 15 = 1110 πόντοι.

Για καλύτερη παρατήρηση και έλεγχο της εφαρμογής μισθοδοσίας έγινε μία προσομοίωση της εφαρμογής για το μήνα Νοέμβριο για τους 3 sample εργαζόμενους. Έτσι θεωρήθηκε από την 1-11-2003 ως τις 31-11-2003 ότι οι εργαζόμενοι αυτοί εργαζόντουσαν κανονικά και κάθε μέρα, για την είσοδο και έξοδό τους χρησιμοποιούσαν τις sample κάρτες τους.
Όλα τα στοιχεία της εργασίας τους για αυτόν το μήνα περιέχονται στο OVERLOGFILE-11 που παρατίθεται στο Παράρτημα Δ..
Από αυτήν την εξομοίωση θα επισημάνουμε τα εξής συμβάντα:

Κάτοχος 1 (Νατάσσα Παπαδοπούλου):
Παίρνει άδεια τις ημέρες 11-11, 12-11, 24-11, 25-11, 26-11 (5 στο σύνολο).
Στις 16-11 εισέρχεται στο χώρο εργασίας αλλά κατά την έξοδό της ξεχνάει να χτυπήσει την κάρτα της. Στις 17-11, εισέρχεται στο χώρο εργασίας και το πρόγραμμα λαμβάνει την εισαγωγή της κάρτας ως έξοδο σε διαφορετική μέρα από την προηγούμενη είσοδό της. Ζητάει από τον διαχειριστή να εισάγει χειροκίνητα τις ώρες εργασίας της κατόχου 1 για τις 16-11. Ο διαχειριστής συμπληρώνει 9ώρες και 27 λεπτά. Η κάτοχος εισάγει ξανά την κάρτα της ως είσοδο πλέον στις 17-11.
Στις 19-11, μετά την λήξη της εργασίας της, πηγαίνει στο διαχειριστή μισθοδοσίας και ζητάει να δει την κατάσταση του μισθού της μέχρι εκείνη την ημέρα.
Από το OVERLOGFILE-11, φαίνεται πως μέχρι και εκείνη την ημέρα η κάτοχος 1 έχει συμπληρώσει 96 ώρες και 161 λεπτά κανονικής εργασίας (συνολικά 98 ώρες και 41 λεπτά), 92 λεπτά υπερωριακής εργασίας με προσαύξηση 30% (συνολικά 1 ώρα και 32 λεπτά) και 3 ώρες και 106 λεπτά υπερωριακής εργασίας με προσαύξηση 50% (συνολικά 4 ώρες και 46 λεπτά). Συνολικά η υπερωριακή της εργασία ανέρχεται στις 6 ώρες και 18 λεπτά.
Αφού το ωρομίσθιό της είναι 12,5 €/ώρα, περιμένουμε να της αντιστοιχεί μισθός:
Κανονική εργασία:

$$98_{\text{ΩΡΕΣ}} \bullet 12{,}5_{€/\text{ΩΡΑ}} + 41_{\text{ΛΕΠΤΑ}} \bullet \left(\frac{12{,}5}{60}\right)_{€/\text{ΛΕΠΤΟ}} = 1233{,}54€.$$

Υπερ/κή εργασία:

$$92_{\text{ΛΕΠΤΑ}} \bullet 1{,}3 \bullet \left(\frac{12{,}5}{60}\right)_{€/\text{ΛΕΠΤΟ}} + 4_{\text{ΩΡΕΣ}} \bullet 1{,}5 \bullet 12{,}5_{€/\text{ΩΡΑ}} + 46_{\text{ΛΕΠΤΑ}} \bullet 1{,}5 \bullet \left(\frac{12{,}5}{60}\right)_{€/\text{ΛΕΠΤΟ}} =$$

$$= 114{,}29€.$$

Ημέρες αδείας: $2_{\text{ΗΜΕΡΕΣ}} \bullet 8_{\text{ΩΡΕΣ}/\text{ΗΜΕΡΑ}} \bullet 12{,}5_{€/\text{ΩΡΑ}} = 200€.$
Δηλαδή αναμένεται ο μισθός μέχρι και τις 19-11 να ανέρχεται στα 1547,83€.
Το ποσό αυτό επαληθεύεται από τα αποτελέσματα που έχουν καταγραφεί στο αρχείο PAYLOGFILE-11 που παρατίθεται επίσης στο Παράρτημα Δ.
Στη συνέχεια στις 21-11, ημέρα Κυριακή, η κάτοχος 1 πραγματοποιεί διπλή υπερωριακή εργασία 5 ωρών και 48 λεπτών, η οποία ως διπλή θα υπολογιστεί στο μισθό ως 11 ωρών και 36 λεπτών.
Στις 31-11 τελειώνει η μισθολογική περίοδος Νοεμβρίου, κατά την οποία η κάτοχος 1, σύμφωνα με τα στοιχεία του OVERLOGFILE-11, έχει συμπληρώσει 136 ώρες και 161 λεπτά κανονικής εργασίας (συνολικά 138 ώρες και 41 λεπτά), 167 λεπτά υπερωριακής εργασίας με προσαύξηση 30% (συνολικά 2 ώρες και 47 λεπτά) και 14 ώρες και 265 λεπτά υπερωριακής εργασίας με προσαύξηση 50% (συνολικά 18 ώρες και 25 λεπτά). Συνολικά η υπερωριακή της εργασία (και με τις δύο προσαυξήσεις) ανέρχεται στις 21 ώρες και 12 λεπτά.
Σύμφωνα με αυτά τα νούμερα και με όσα έχουν αναφερθεί στην ενότητα 6.2.4 για τον τρόπο που καταγράφονται στους μετρητές, οι πόντοι των μετρητών θα έπρεπε να είναι για κάθε περίπτωση:
Normhour: $136 \bullet 10 = 1360$ πόντοι.
Normmins: $161 \bullet 10 = 1610$ πόντοι.
Yperhour: $14 \bullet 15 = 210$ πόντοι.
Ypermins: $167 \bullet 13 + 265 \bullet 15 = 6146$ πόντοι.

Fifteens: $167 \bullet 1 = 167$ πόντοι.
Αυτοί επαληθεύονται από τα καταγεγραμμένα στοιχεία της τελευταίας εξόδου της κατόχου 1, στις 31-11.
Ο συνολικός μισθός που αντιστοιχεί σε αυτές τις ώρες εργασίας υπολογίζεται:
Κανονική εργασία:

$$138_{\text{ΩΡΕΣ}} \bullet 12{,}5_{€/\text{ΩΡΑ}} + 41_{\text{ΛΕΠΤΑ}} \bullet \left(\frac{12{,}5}{60}\right)_{€/\text{ΛΕΠΤΟ}} = 1733{,}54€.$$

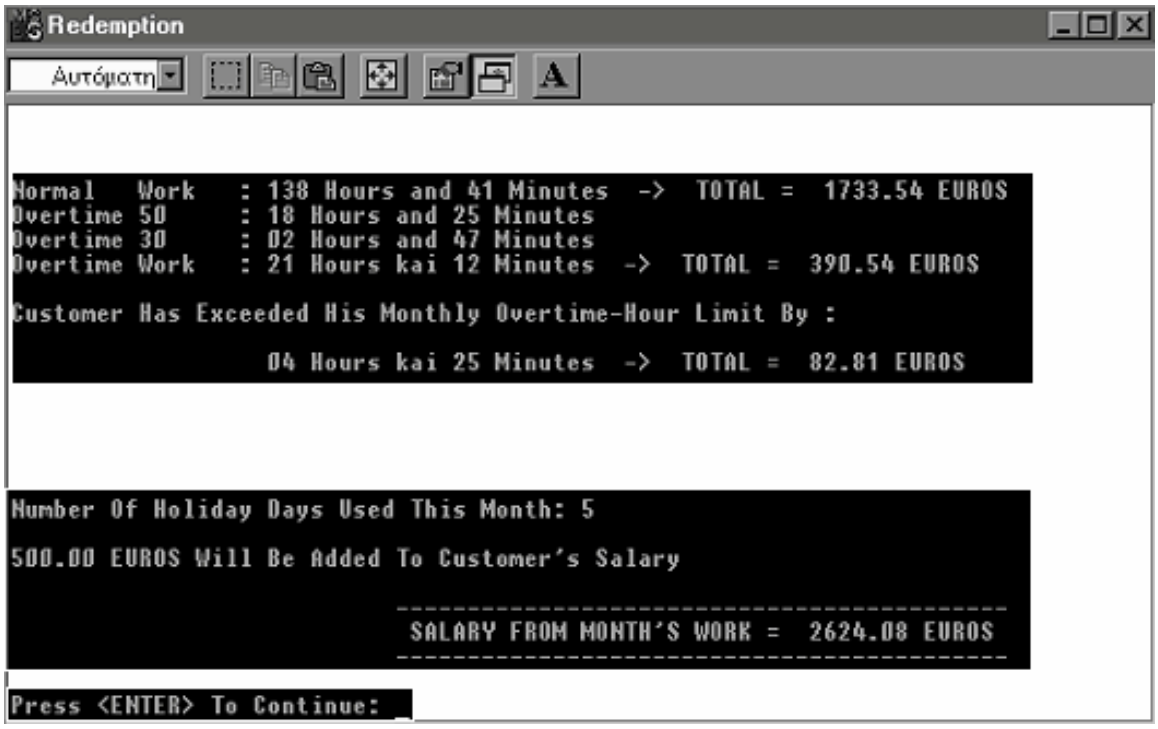
Υπερ/κή εργασία:

$$167_{\text{ΛΕΠΤΑ}} \bullet 1{,}3 \bullet \left(\frac{12{,}5}{60}\right)_{€/\text{ΛΕΠΤΟ}} + 18_{\text{ΩΡΕΣ}} \bullet 1{,}5 \bullet 12{,}5_{€/\text{ΩΡΑ}} + 25_{\text{ΛΕΠΤΑ}} \bullet 1{,}5 \bullet \left(\frac{12{,}5}{60}\right)_{€/\text{ΛΕΠΤΟ}} =$$

$$= 390{,}54€.$$

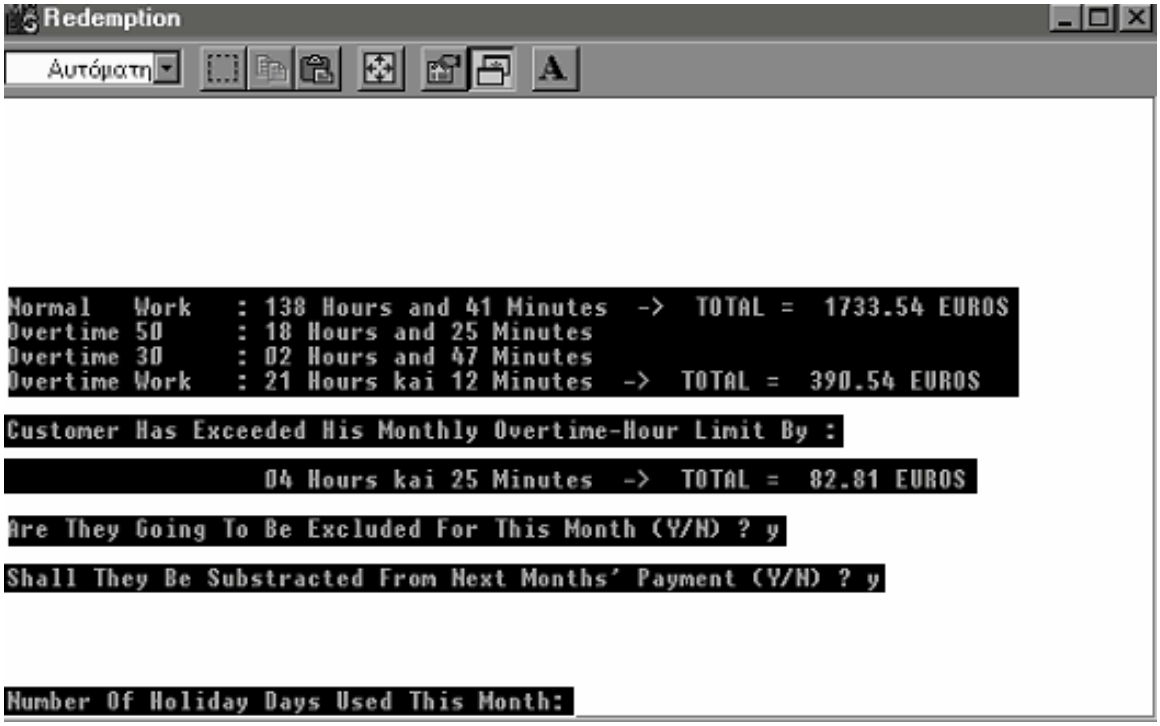
Ημέρες αδείας: $5_{\text{ΗΜΕΡΕΣ}} \bullet 8_{\text{ΩΡΕΣ/ΗΜΕΡΑ}} \bullet 12{,}5_{€/\text{ΩΡΑ}} = 500€.$
Δηλαδή αναμένεται ο μισθός μέχρι και τις 31-11 να ανέρχεται στα 2624,08€.

Αυτό επαληθεύεται στο ακόλουθο στιγμιότυπο του προγράμματος Redemption, όπου η κάτοχος 1 έχει ζητήσει τον υπολογισμό μόνο του μισθού της (επιλογή 2 στο μενού επιλογών του προγράμματος).

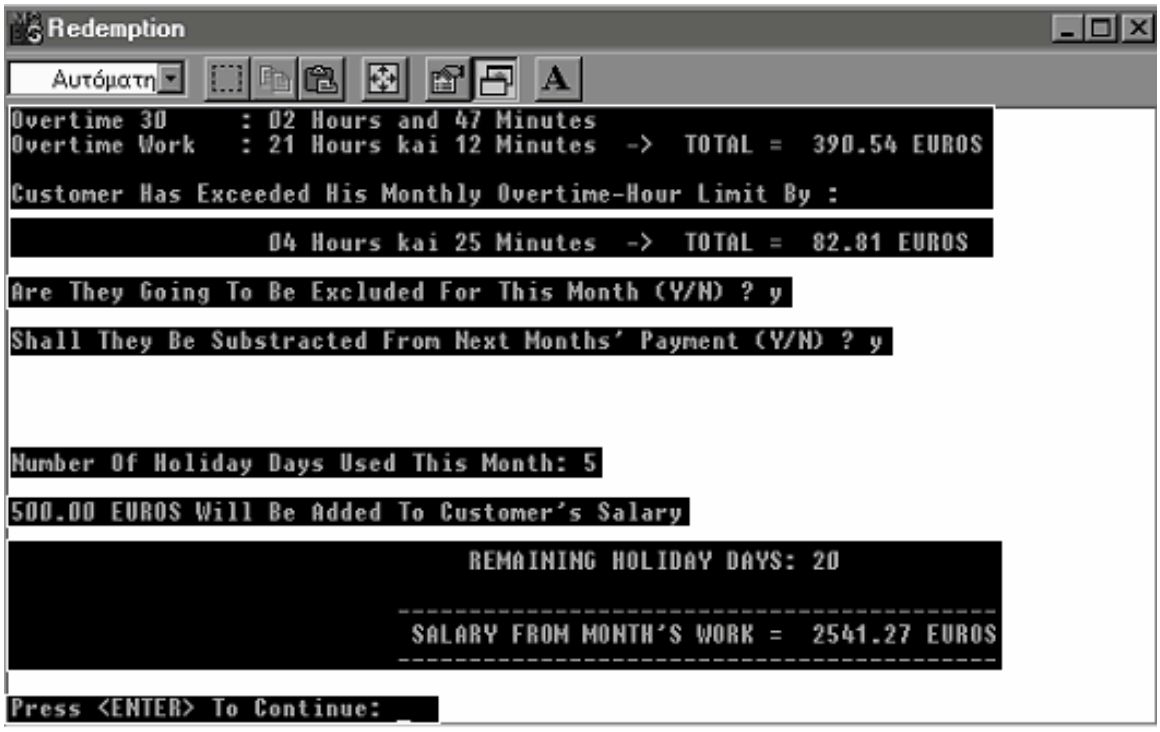


Σχήμα 7.2.3– Υπολογισμός μισθού κατόχου 1

Όταν η κάτοχος 1 ζητάει την εξαργύρωση του μισθού αυτού (επιλογή 3 στο πρόγραμμα Redemption), προκύπτουν τα ακόλουθα στιγμιότυπα που φαίνονται στα Σχήματα 7.2.4 και 7.2.5:

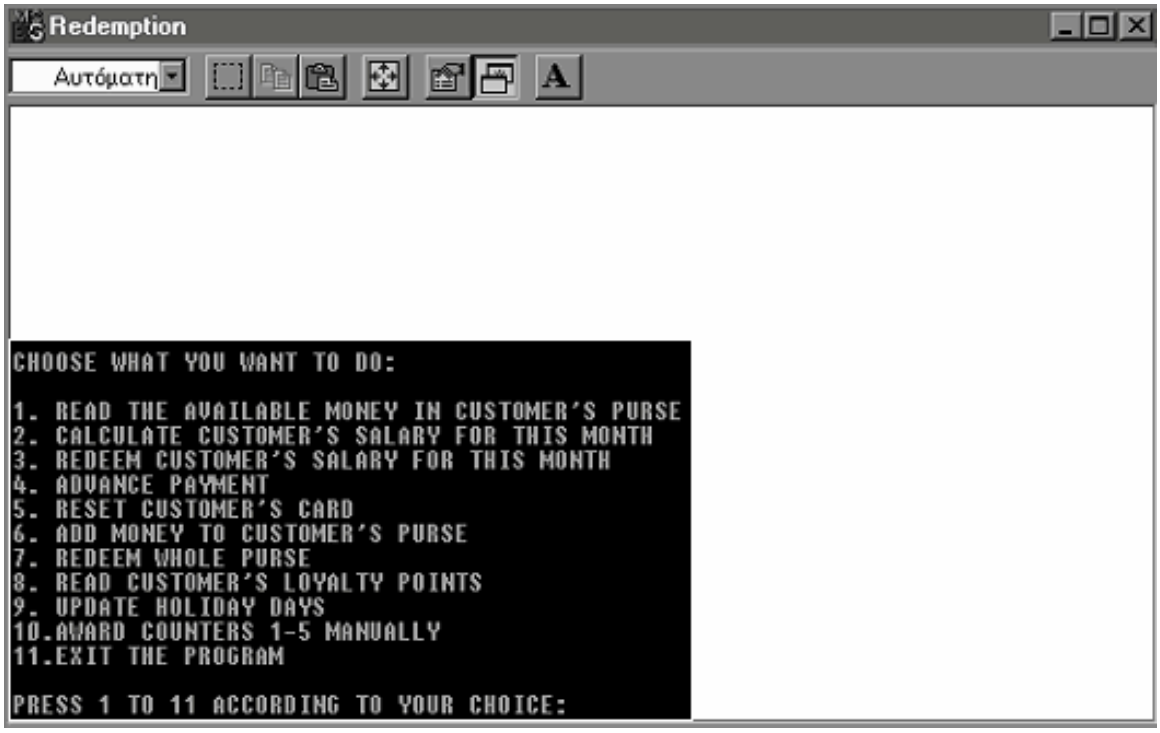


Σχήμα 7.2.4– Εξαργύρωση μισθού κατόχου 1 -α



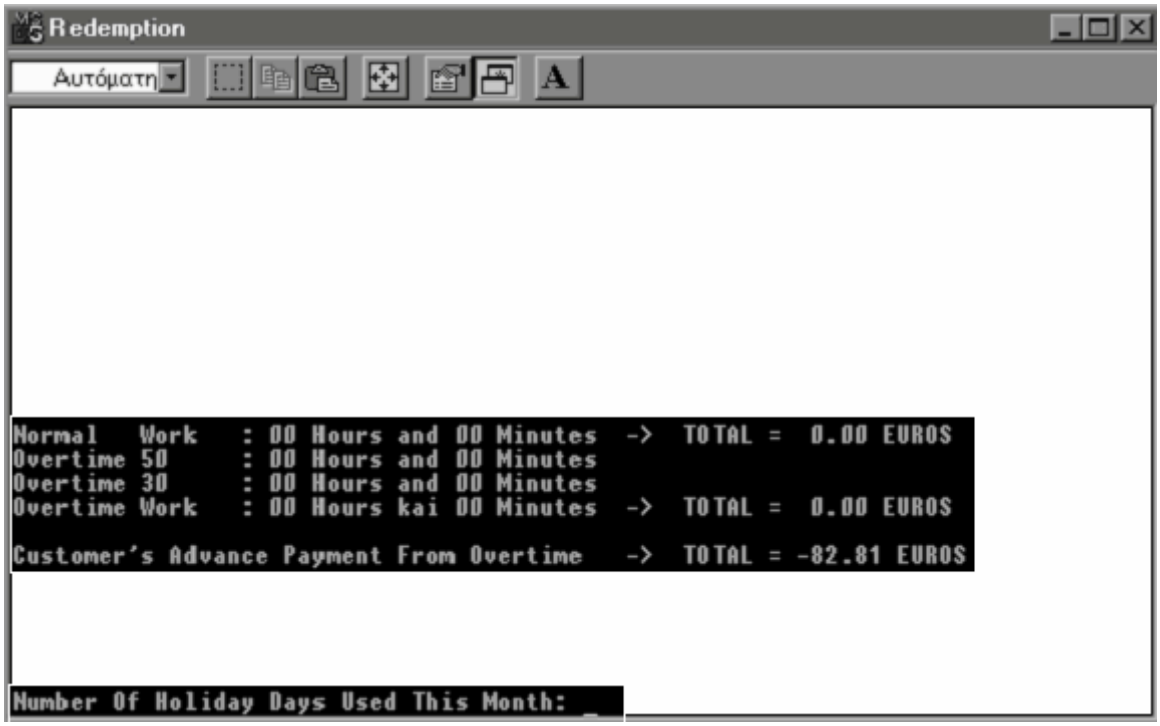
Σχήμα 7.2.5– Εξαργύρωση μισθού κατόχου 1 -β

Η κάτοχος 1 έχει όριο υπερωριακής εργασίας 14 ώρες το μήνα, και το έχει υπερβεί κατά 4 ώρες και 25 λεπτά, που αντιστοιχούν σε 82,81€. Ο διαχειριστής επιλέγει να μη τα μετρήσει στις υπερωρίες του μήνα που πέρασε, αλλά να τις υπολογίσει στον επόμενο μήνα, θεωρώντας τα υπερωριακή προκαταβολή. Έτσι ο κανονικός μισθός της κατόχου 1 για αυτό το μήνα θα είναι 2624,08 - 82,81 = 2541,27€ και τα 82,81€ θα τα λάβει ως προκαταβολή της υπερωριακής εργασίας του επόμενου μήνα.



Σχήμα 7.2.6 – Κεντρικό Μενού Redemption

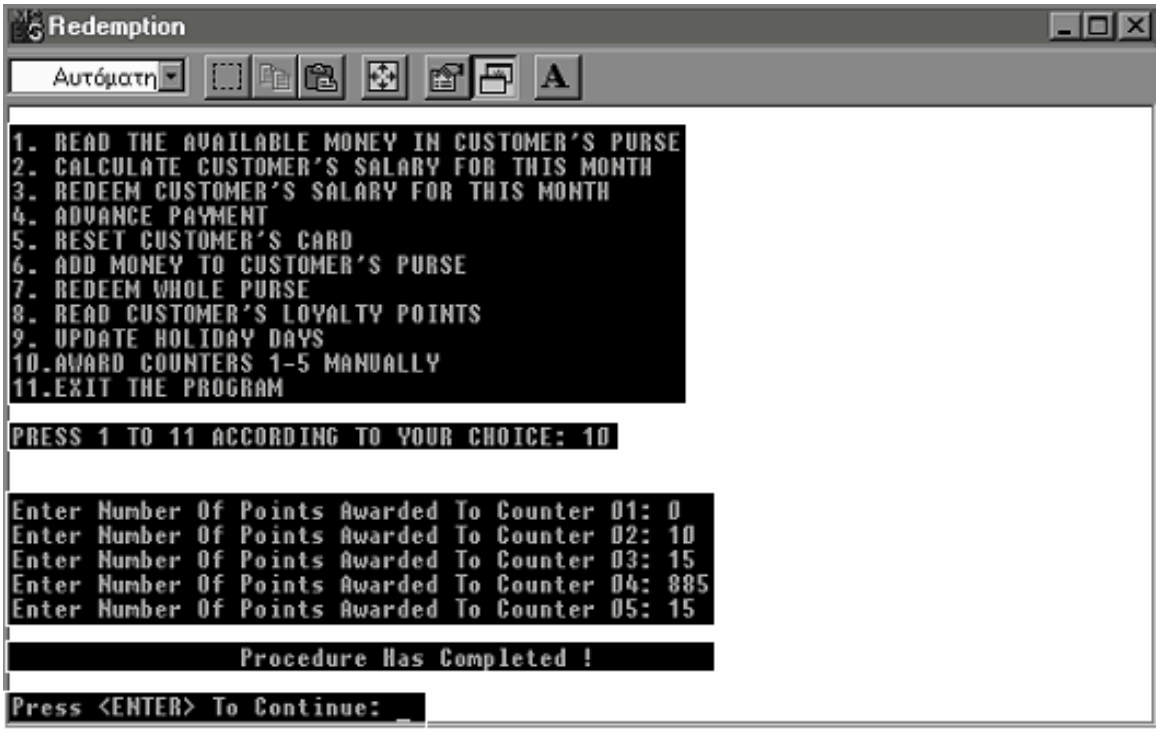
Ο διαχειριστής μετά επιλέγει από το μενού που φαίνεται στο στιγμιότυπο του Σχήματος 7.2.6 την αρχικοποίηση – μηδενισμό αρχείων της κάρτας. Μετά τη διαδικασία αυτή θα πρέπει να μείνει στον μετρητή Advancep το ποσό της υπερωριακής προκαταβολής και να φαίνεται στις αναλύσεις του μισθού της κατόχου 1 στον επόμενο μήνα.
Αυτό ακριβώς φαίνεται στο στιγμιότυπο του Σχήματος 7.2.7 που παρατίθεται παρακάτω, και που έχει προκύψει από την επιλογή 2 (υπολογισμό μισθού) ακριβώς μετά την διαδικασία αρχικοποίησης (επιλογή 5).



Σχήμα 7.2.7 – Νέος υπολογισμός μισθού κατόχου 1

Όλες αυτές οι πληροφορίες και οι πράξεις έχουν καταγραφεί και στο αρχείο PAYLOGFILE-11 (Παράρτημα Δ).

Κάτοχος 2 (Γιάννης Τασόπουλος):
Στις 18-11 ο κάτοχος 2 δουλεύει 7 ώρες και 59 λεπτά, βγαίνει από το χώρο εργασίας και αργότερα μέσα στην ημέρα εισέρχεται ξανά και εξέρχεται μετά τα μεσάνυχτα (μετά από 2 ώρες και 2 λεπτά), την επόμενη μέρα (στις 19-11 δηλαδή τα ξημερώματα). Επειδή ο διαχειριστής μισθοδοσίας δεν είναι παρών τότε, δεν συμπληρώνονται οι ώρες αυτές.
Έτσι στις 22-11 πηγαίνει στον διαχειριστή μισθοδοσίας για να συμπληρώσει αυτές τις ώρες. Οι πόντοι που πρέπει να συμπληρωθούν στους μετρητές είναι:
Normhour: $0 \bullet 10 = 0$ πόντοι,
Normmins: $1 \bullet 10 = 10$ πόντοι,
Yperhour: $1 \bullet 15 = 15$ πόντοι.
Ypermins: $15 \bullet 13 + 46 \bullet 15 = 885$ πόντοι,
Fifteens: $15 \bullet 1 = 15$ πόντοι.
Αφού ο κάτοχος 2 είχε συμπληρώσει εκείνη την ημέρα 7 ώρες και 59 λεπτά κανονικής εργασίας, από τις 2 ώρες και 2 λεπτά που συμπληρώνονται, το 1 λεπτό αντιστοιχεί σε κανονική εργασία, τα πρώτα 15 λεπτά σε υπερωριακή εργασία με προσαύξηση 30% και οι υπόλοιπες 1 ώρα 46 λεπτά αντιστοιχούν σε υπερωριακή εργασία με προσαύξηση 50%. Η πρόσθεση των πόντων γίνεται με τη λειτουργία 10 του προγράμματος Redemption, όπως φαίνεται στο στιγμιότυπο που ακολουθεί και όπως έχει καταγραφεί στο αρχείο PAYLOGFILE-11 (Παράρτημα Δ).

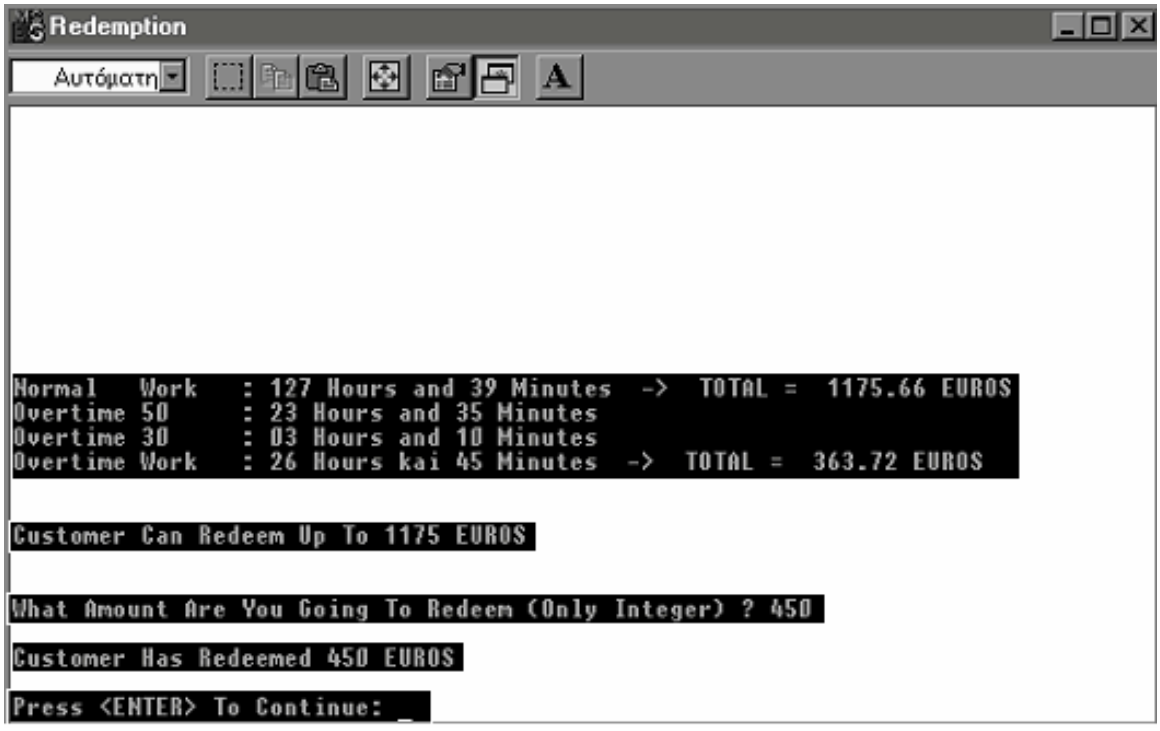


Σχήμα 7.2.8 – Χειροκίνητη προσθήκη πόντων στους μετρητές

Στις 29-11, παρουσιάζεται σφάλμα στη διαδικασία ενημέρωσης των αρχείων της κάρτας και έτσι, δεν ενημερώνονται οι μετρητές με τους νέους πόντους που θα έπρεπε να προστεθούν. Από το log εκείνης της ημέρας παρατηρούμε όμως ότι αν είχαν γίνει σωστά οι ενημερώσεις, θα έπρεπε οι μετρητές Normhour, Normmins, Yperhour, Ypermins και Fifteens να έχουν αντίστοιχα τις τιμές 1420, 880, 225, 9850 και 220. Οι τελευταίες τιμές που γνωρίζουμε ότι είχαν προκύπτουν από το log στις 24-11 όπου οι τιμές των μετρητών ήταν αντίστοιχα 1340, 880, 210, 9430 και 205. Άρα δε συμπληρώθηκαν αντίστοιχα στους μετρητές οι πόντοι 80, 0, 15, 420 και 15.

Στις 30-11, παρουσιάζεται πάλι σφάλμα κατά την έξοδο του κατόχου 2 και έτσι δεν ενημερώνονται σωστά όλοι οι μετρητές με τις νέες τιμές που πρέπει να έχουν. Από το log εκείνης της ημέρας προκύπτει ότι οι πόντοι που θα έπρεπε να αποδοθούν στους μετρητές αυτούς ήταν 80, 0, 15, 510 και 15 (η πληροφορία προκύπτει από τις μεταβλητές aw[1] ως aw[5]. Από αυτές τις αποδόσεις έγιναν μόνο οι 3 πρώτες οπότε μένουν να συμπληρωθούν στους μετρητές αντίστοιχα οι πόντοι 0, 0, 0, 510 και 15.
Συνδυάζοντας τους πόντους που πρέπει να προστεθούν από τις 29-11 και από τις 30-11 προκύπτει ότι συνολικά πρέπει να αποδοθούν στους μετρητές αντιστοίχως οι πόντοι 80, 0, 15, 930 και 30, όπως και γίνεται με την επιλογή 10 του προγράμματος Redemption στις 30-11.
Φαίνεται έτσι η βοήθεια που μπορεί να παρέχει το log αρχείο κάθε προγράμματος για την αντιμετώπιση σφαλμάτων.

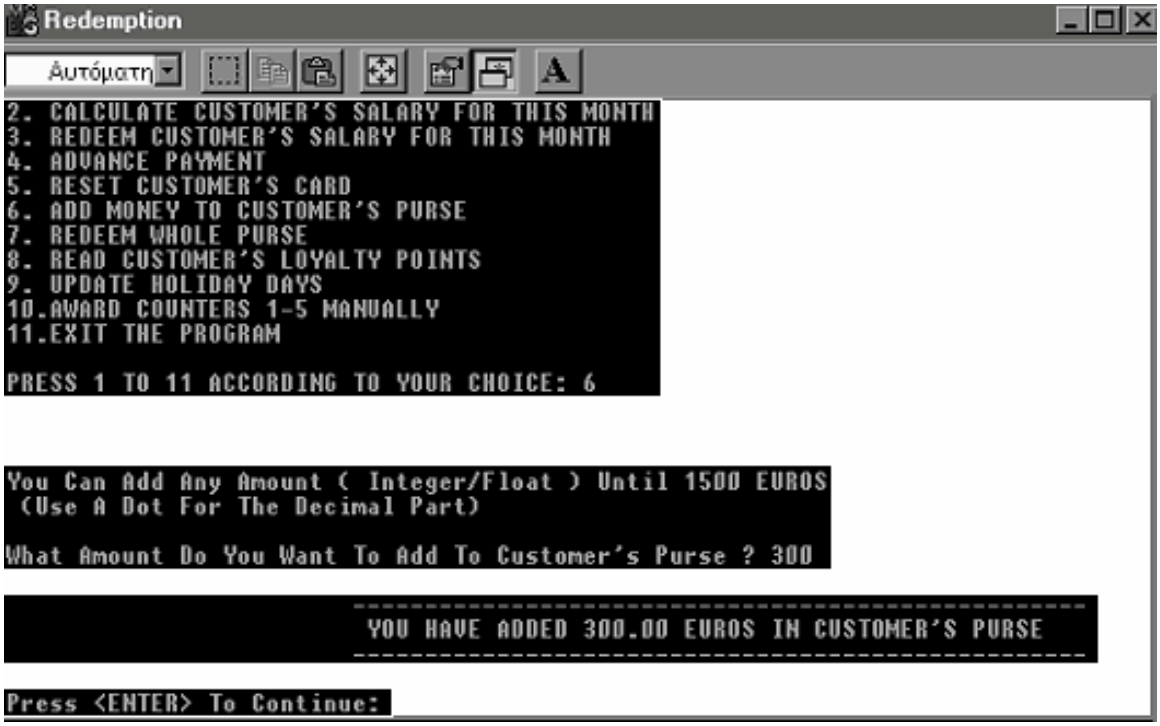
Κάτοχος 3 (Μάριος Κατσάνος):
Στις 13-11, ημέρα Σάββατο εργάζεται για 2 ώρες και 22 λεπτά, εργασία που υπολογίζεται ολόκληρη ως υπερωριακή με προσαύξηση 50%. Στις 18-11 και 19-11 παίρνει άδεια.
Στις 21-11, ημέρα Κυριακή, εργάζεται για 4 ώρες και 2 λεπτά, εργασία που θεωρείται διπλή υπερωριακή και θα υπολογιστεί στο μισθό ως υπερωριακή εργασία 8 ωρών και 4 λεπτών, με προσαύξηση 50%.
Στις 24-11 ο κάτοχος 3 πάει στο διαχειριστή για να κάνει έναν υπολογισμό του μισθού του μέχρι τότε και να πάρει αν δύναται μία κανονική προκαταβολή. Για να πάρει κανονική προκαταβολή θα πρέπει να έχει δουλέψει κάποιες ώρες κανονικής εργασίας και να μην χρωστάει υπερωριακή προκαταβολή από προηγούμενο μήνα. Σύμφωνα με τις εγγραφές στο αρχείο OVERLOGFILE-11, μέχρι και τις 24-11 έχει συμπληρώσει 127 ώρες και 39 λεπτά κανονικής εργασίας τα οποία αντιστοιχούν σε 1175,66€.
Από αυτά θα μπορεί να λάβει ως κανονική προκαταβολή μέχρι και τα 117€, όπως φαίνεται και στο ακόλουθο στιγμιότυπο του προγράμματος Redemption.



Σχήμα 7.2.9 – Κανονική προκαταβολή για κάτοχο 3

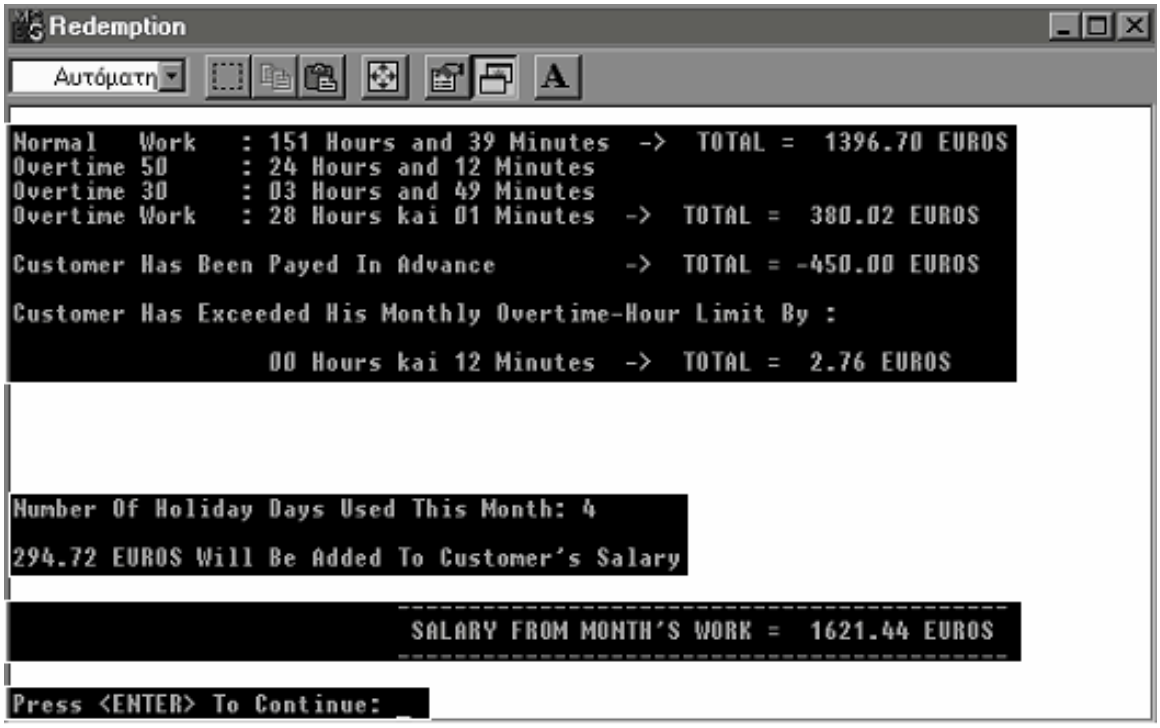
Ο κάτοχος 3 επιλέγει να λάβει ως προκαταβολή 450€ και αφού γίνει αυτή η συναλλαγή και καταγραφεί στο αρχείο RECEIPT-3 για χρήση από τον ίδιο, επιλέγει να προσθέσει 300€ από αυτά στο ηλεκτρονικό πορτοφόλι του. Το στιγμιότυπο της προσθήκης του ποσού αυτού στο

ηλεκτρονικό πορτοφόλι του κατόχου 3 φαίνεται στο Σχήμα 7.2.10 καθώς και στο αρχείο RECEIPT-3 που δημιουργήθηκε για εκείνη τη συναλλαγή.



Σχήμα 7.2.10 – Προσθήκη αξίας στο ηλεκτρονικό πορτοφόλι του κατόχου 3

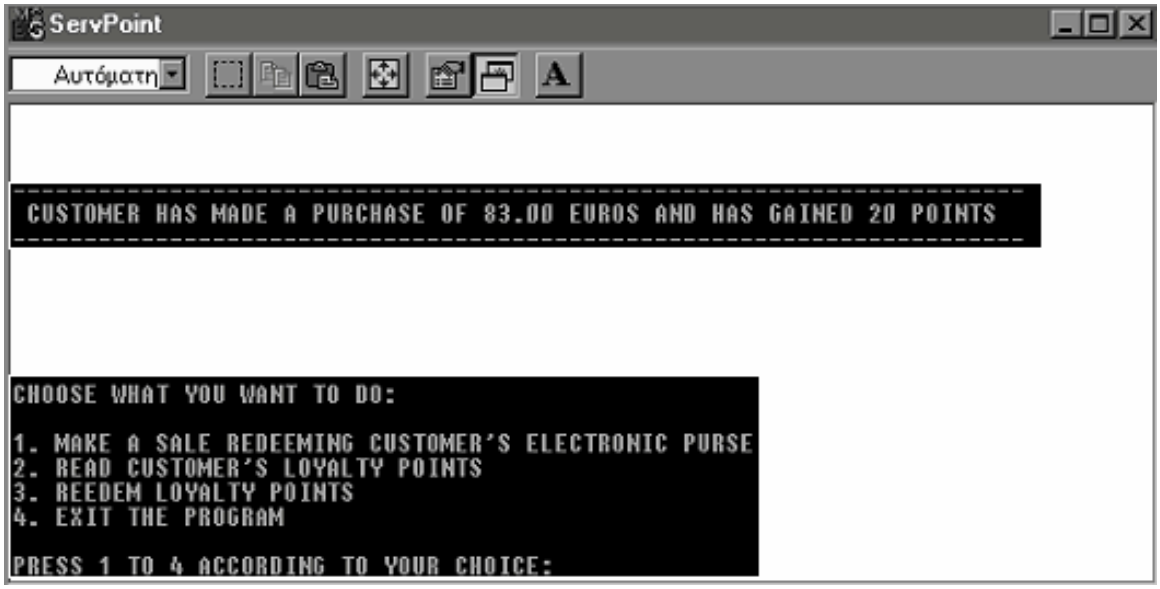
Έτσι όταν στις 31-11 ο κάτοχος 3 πάει να εισπράξει τον συνολικό μισθό του, θα πρέπει να υπολογιστεί και η προκαταβολή που έχει πάρει στις 24-11. Αυτό φαίνεται στο στιγμιότυπο του Σχήματος 7.2.11. Συνολικά έχει πάρει χρησιμοποιήσει μέχρι τότε 4 ημέρες αδείας.



Σχήμα 7.2.11– Υπολογισμός μισθού κατόχου 3

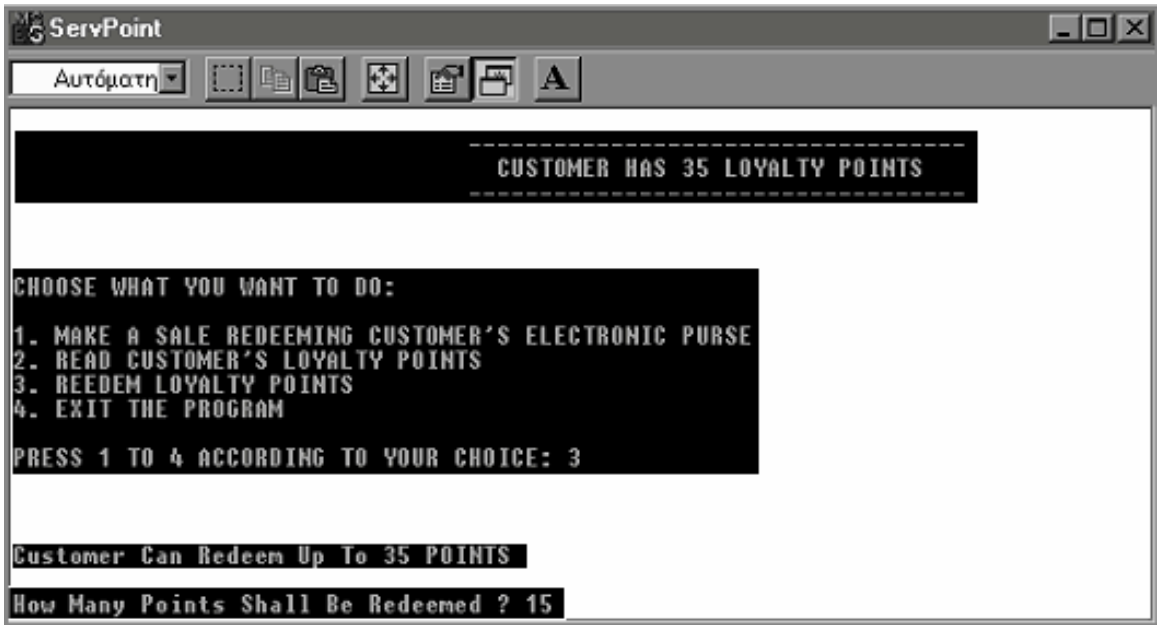
7.3 Εφαρμογή Ηλεκ/νικού Πορτοφολιού – Προγ/ματος Εμπιστοσύνης

Θα εξετάσουμε την περίπτωση του κατόχου 3, ο οποίος όπως είδαμε στην προηγούμενη ενότητα, έχει προσθέσει 300€ στο ηλεκτρονικό πορτοφόλι του. Θεωρείται ότι ο κάτοχος της κάρτας βρίσκεται σε ένα σημείο πώλησης μέσα στο χώρο του πανεπιστημίου και επιθυμεί την αγορά κάποιων προϊόντων συνολικής αξίας 83€. Η αγορά αυτή γίνεται με χρήση του προγράμματος ServPoint.

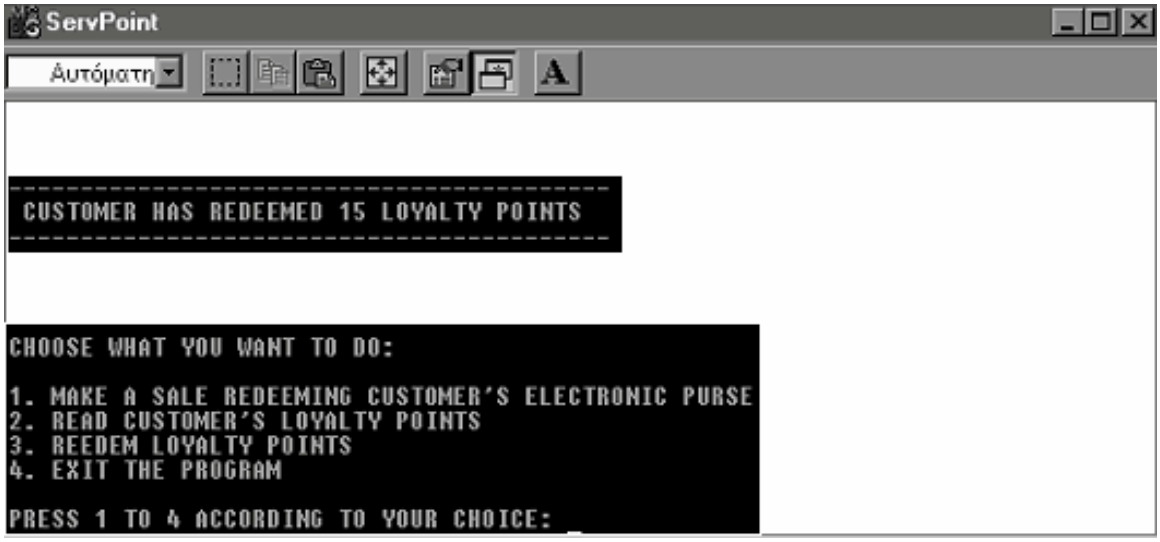


Σχήμα 7.3.1– Αγορά κατόχου 3

Μετά από κάποιο χρονικό διάστημα, και ενώ ο κάτοχος 3 έχει συγκεντρώσει από αγορές του 35 πόντους συνολικά, επιθυμεί την εξαργύρωση κάποιων αυτών για την απόκτηση ενός δώρου. Συγκεκριμένα επιθυμεί την εξαργύρωση 15 πόντων την οποία πραγματοποιεί όπως φαίνεται και στα ακόλουθα στιγμιότυπα.



Σχήμα 7.3.2 – Ανάγνωση πόντων κατόχου 3

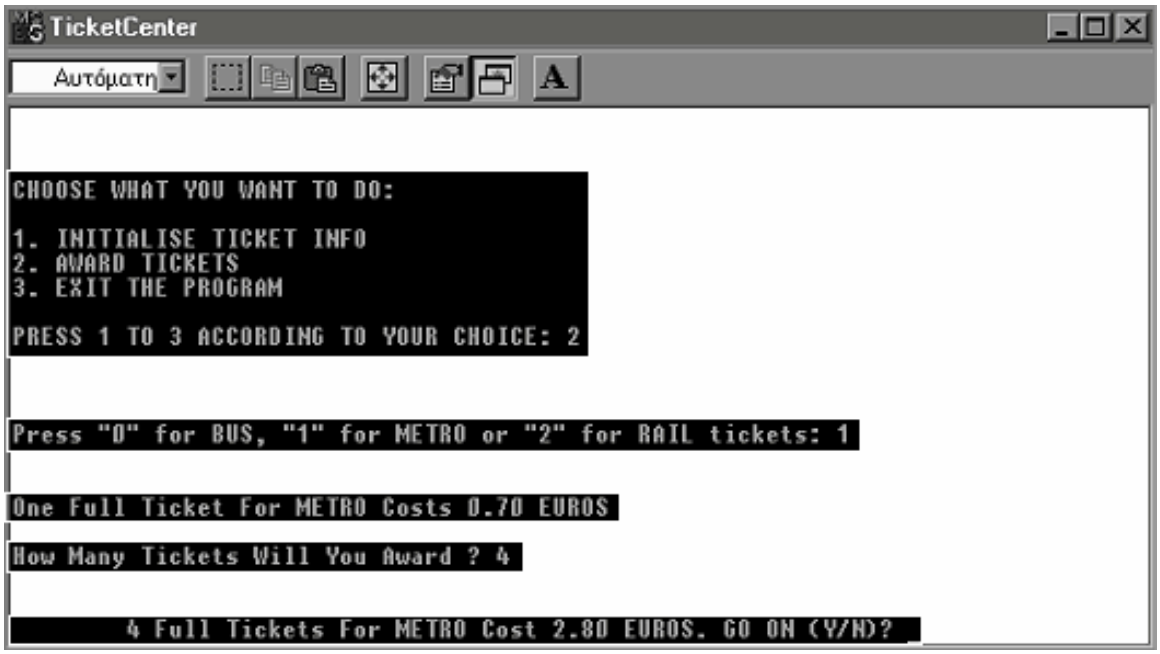


Σχήμα 7.3.3 – Εξαργύρωση πόντων κατόχου 3

Αρχικά διαβάζει τον αριθμό των πόντων που έχει διαθέσιμους ο κάτοχος στην κάρτα του και έπειτα πραγματοποιεί την εξαργύρωση των πόντων που επιθυμεί. Αν ο διαχειριστής εισήγαγε μεγαλύτερο ποσό από αυτό που επιτρέπεται, η εξαργύρωση δε θα πραγματοποιούταν.
Διαβάζοντας μετά τον αριθμό των πόντων που έχει πλέον η κάρτα του κατόχου 3 αποθηκευμένους, προκύπτει πως οι πόντοι που έχουν απομείνει στην κάρτα είναι 20.

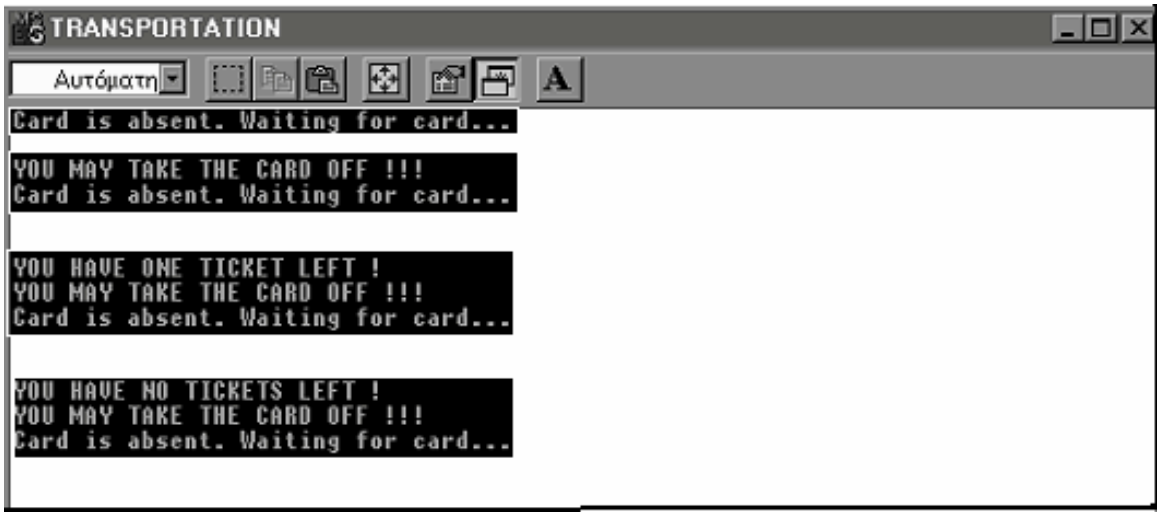
7.4 Εφαρμογή Εισιτηρίων

Θεωρείται ότι ο κάτοχος 2 επιθυμεί να αποθηκεύσει στην κάρτα του αξία που αντιστοιχεί σε 4 ολόκληρα εισιτήρια για το μετρό. Ο διαχειριστής εισιτηρίων, χρησιμοποιώντας το πρόγραμμα TicketCenter, αρχικοποιεί πρώτα την κάρτα δηλώνοντας ότι τα εισιτήρια που χρησιμοποιεί ο κάτοχός της είναι ολόκληρα. Έπειτα, όπως φαίνεται και στο στιγμιότυπο του Σχήματος 7.4.1. επιλέγει το μετρό ως μέσο μεταφοράς για το οποίο προορίζονται τα εισιτήρια



Σχήμα 7.4.1 – Προσθήκη αξίας εισιτηρίων

Το πρόγραμμα Transportation στη συνέχεια, ανάλογα με το μέσο μεταφοράς στο οποίο τρέχει η εφαρμογή, αφαιρεί με κάθε "ακύρωση" της κάρτας την κατάλληλη αξία εισιτηρίου.
Στην υλοποίηση που πραγματοποιήθηκε, με κάθε ακύρωση παράγεται και μία εγγραφή στο αρχείο TICKET και έτσι για μία "ακύρωση" εισιτηρίου της κατόχου 2 στο μετρό παράγεται η εξής εγγραφή στο αρχείο: 20/02/2004 , 20:42:07 -> 0.70 EUROS.
Επίσης, το πρόγραμμα υπολογίζει τον αριθμό των εισιτηρίων που απομένουν στην κάρτα του κατόχου και όταν αυτός ο αριθμός είναι το 1, τυπώνει ανάλογο μήνυμα στην οθόνη, προειδοποιώντας τον κάτοχο. Στο Σχήμα 7.4.2 φαίνονται 3 διαδοχικές ακυρώσεις εισιτηρίου για μία κάρτα που περιείχε αρχικά 3 εισιτήρια.



Σχήμα 7.4.2 – Διαδοχικές ακυρώσεις εισιτηρίου

8

Επίλογος

Με την διπλωματική αυτή εργασία, επιχειρήθηκε μία πρώτη γνωριμία με τη τεχνολογία των έξυπνων καρτών, μία τεχνολογία σύγχρονη, με μεγάλη εξάπλωση παγκοσμίως και με σημαντικές προοπτικές για το μέλλον.
Μέσω της τεχνογνωσίας που αποκτήθηκε κατά την εκπόνηση της εργασίας έγιναν σαφείς σημαντικές λειτουργίες των έξυπνων καρτών όπως αυτή της προηγμένης ασφάλειας δεδομένων και συναλλαγών και η δυνατότητα ασφαλούς αποθήκευσης και μεταφοράς προσωπικών δεδομένων.
Αν αναλογιστούμε το γεγονός ότι οι κάρτες που χρησιμοποιήθηκαν στο πρακτικό κομμάτι της εργασίας είναι από τις πλέον απλές κάρτες με μικροεπεξεργαστή, ότι ο reader GCR410 είναι επίσης απλός και παρέχει βασικές λειτουργίες ανάγνωσης, εγγραφής και επικοινωνίας και τέλος το ότι επιτεύχθηκε η υλοποίηση τριών διαφορετικών εφαρμογών με βάση την ίδια κάρτα, γίνεται κατανοητό πως οι δυνατότητες των εξελιγμένων έξυπνων καρτών που κυκλοφορούν στην παγκόσμια αγορά έχουν ανοίξει το δρόμο για σημαντικές εφαρμογές μεγάλης απήχησης.
Συγκεκριμένα, η κυρίαρχη τάση στην αγορά έξυπνων καρτών αφορά κάρτες ασύρματες ή Combi, προγραμματιζόμενες με γλώσσα Java, με μεγάλες αποθηκευτικές δυνατότητες και κυρίως με δυνατότητα συνύπαρξης διαφορετικών εφαρμογών πάνω στην ίδια κάρτα (multi-application cards). Όλα αυτά τα στοιχεία έχουν χρησιμοποιηθεί για την ανάπτυξη και υλοποίηση νέων προηγμένων εφαρμογών αλλά και για τη βελτιστοποίηση παλαιότερων εφαρμογών όπως αυτή του ηλεκτρονικού πορτοφολιού ή του ελέγχου πρόσβασης.
Αυτό που βασικά απομένει για την υιοθέτηση των έξυπνων καρτών ως την πιο “έξυπνη” λύση για την επίλυση ποικίλων προβλημάτων και την βελτίωση της λειτουργικότητας καθημερινών πρακτικών, είναι η επέκταση της διαλειτουργικότητας των έξυπνων καρτών και η προτυποποίηση όλων των στοιχείων που αφορούν αυτή τη τεχνολογία.
Οι έξυπνες κάρτες και οι συσκευές που τις συνοδεύουν πρέπει να είναι συμβατές με όλα τα προγραμματιστικά εργαλεία και όλων των ειδών το hardware εξοπλισμό.
Κυρίως όμως, απομένει να γίνουν οι έξυπνες κάρτες η νέα συνήθεια για τον καταναλωτή και τον τελικό χρήστη, να αποτελέσουν βασικό στοιχείο στη συνείδησή του και να υποκαταστήσουν τις παλαιότερες μεθόδους συναλλαγών και εμπορικών λειτουργιών.

9

Βιβλιογραφία

[ACB+97] Allen, Catherine a., William J. Barr, Ron Schultz. *Smart Cards: seizing strategic business opportunities*, 1997

[Eve02] Dr. David B Everett, *Smart Card Technology: Introduction To Smart Cards*, available at www.smartcardclub.co.uk, Smart Card News Ltd., 2002

[Fin03] Klaus Finkenzeller; translated by Rachel Waddington, *RFID handbook: Fundamentals and applications in contactless smart cards and identification*, 2nd edition, 2003

[Haw02] Peter Hawkes, *Opinion: "Why Is No One Trying to Sell Me An Electronic Wallet?"*, available at www.smartcardclub.co.uk, Smart Card News Ltd., 2002

[RE00] W. Rankl, W. Effing; translated by Kermeth Cox. *Smart Card Handbook,* 2nd edition, 2000

[ZM94] Jose Luis Zoreda, Jose Manuel Oton. *Smart Cards*, 1994

URLs:

www.ePaynews.com
www.cardwerk.com
www.smartcardbasics.com
www.gemplus.com
www.weethet.nl
www.scan.co.id
www.smartcardgroup.com
www.smartcardalliance.org
www.estrategy.gov
www.smartcardclub.co.uk
www.jdpdigital.com
whatis.techtarget.com
www.paceintegration.com
www.javaworld.com
www.infosyssec.org
www.citi.umich.edu
www.cardshow.com