Πανεπιστημιο Κρητησ
Σχολη Θετικων Επιστημων
Τμημα Επιστημησ Υπολογιστων

# Ανάλυση Ασφαλείας Συστημάτων Έξυπνων Καρτών

Αρτέμιος Γ. Βογιατζής

Μεταπτυχιακή Εργασία

Ηράκλειο, Σεπτέμβριος 2001

Πανεπιστημιο Κρητης
Σχολη Θετικων Επιστημων
Τμημα Επιστημης Υπολογιστων

# Ανάλυση Ασφαλείας Συστημάτων Έξυπνων Καρτών

Εργασία που υποβλήθηκε από τον
Αρτεμιο Γ. Βογιατζη
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
Μεταπτυχιακου Διπλωματος Ειδικευσης

Συγγραφέας:

Αρτέμιος Γ. Βογιατζής
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Εξεταστική Επιτροπή:

Απόστολος Τραγανίτης, Αναπληρωτής Καθηγητής
Επόπτης

Δημήτριος Ν. Σερπάνος, Αναπληρωτής Καθηγητής
Επιβλέπων

Γεώργιος Φ. Γεωργακόπουλος, Επίκουρος Καθηγητής
Μέλος

Γεώργιος Τζιρίτας, Αναπληρωτής Καθηγητής
Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Σεπτέμβριος 2001

*Σε αυτούς που,*
*σε πείσμα των καιρών,*
*τολμούν να είναι αληθινοί.*

# Ανάλυση Ασφαλείας Συστημάτων Έξυπνων Καρτών

Αρτέμιος Γ. Βογιατζής

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

## Περίληψη

Οι έξυπνες κάρτες είναι ένα ιδανικό μέσο για την αντιπροσώπευση φυσικών οντοτήτων σε συστήματα ηλεκτρονικών συναλλαγών. Το μικρό μέγεθος και η απλότητα χρήσης διευκολύνουν την μεταφορά και χρήση τους οπουδήποτε και από οποιονδήποτε. Προσφέρουν επίσης ένα υψηλό επίπεδο ασφάλειας, καθώς μπορούν να αποθηκεύσουν με ασφαλή τρόπο μυστικά κλειδιά και να εκτελέσουν κρυπτογραφικούς αλγόριθμους.

Η *κρυπτανάλυση πλάγιου καναλιού* είναι μία καινοτόμος μέθοδος για την ανάκτηση μυστικών κλειδιών που είναι αποθηκευμένα σε έξυπνες κάρτες. Η μέθοδος αυτή είναι ιδιαίτερα δημοφιλής, γιατί μπορεί να ανακτήσει τα μυστικά κλειδιά μίας κάρτας με ελάχιστη προσπάθεια.

Μελετούμε την εφαρμογή δύο ειδών κρυπτανάλυσης πλάγιου καναλιού, τις ενεργές και παθητικές επιθέσεις στο υλικό, στις τρεις γενιές έξυπνων καρτών.

Το κύριο χαρακτηριστικό των καρτών πρώτης γενιάς είναι η απλότητα του υλικού και η απουσία επεξεργαστή. Για το λόγο αυτό αναφέρονται και ως κάρτες μνήμης. Για λόγους ασφάλειας, έχει ενσωματωθεί σε αυτές λογική σε υλικό, ικανή να εκτελεί απλούς αλγόριθμους κρυπτογράφησης. Αναπτύξαμε μία βιβλιοθήκη επικοινωνίας με κάρτες μνήμης, χρησιμοποιώντας τον αναγνώστη έξυπνων καρτών Gemplus GCR-410. Με τον τρόπο αυτό παρέχουμε μία κοινή πλατφόρμα υλοποίησης παθητικών επιθέσεων για κάρτες μνήμης και έξυπνες κάρτες. Προτείνουμε επίσης ένα θεωρητικό μοντέλο κατανάλωσης ισχύος προσαρμοσμένο σε κάρτες μνήμης. Το μοντέλο αυτό επιτρέπει την ανακατασκευή των μυστικών αλγορίθμων κρυπτογράφησης που χρησιμοποιούνται σε κάρτες μνήμης.

Οι έξυπνες κάρτες διαθέτουν επεξεργαστή και προγραμματίζονται για την εκτέλεση μίας μόνο εφαρμογής. Μία συνήθης χρήση τους είναι η ταυτοποίηση φυσικών οντοτήτων με κατάλληλα κρυπτογραφικά σχήματα, όπως το σχήμα Fiat-Shamir. Αποδεικνύουμε ότι υπάρχουν διαμορφώσεις συστημάτων όπου το μοντέλο ενεργών επιθέσεων Bellcore δεν είναι έγκυρο. Επεκτείνουμε το μοντέλο ώστε να είναι έγκυρο σε όλες, εκτός από μία, των διαμορφώσεων. Προτείνουμε τέλος ένα νέο μοντέλο ενεργών επιθέσεων, το S-TAM. Το S-TAM απαιτεί ισχυρότερες υποθέσεις από το μοντέλο Bellcore. Είναι όμως αποδείξιμα έγκυρο σε όλες τις διαμορφώσεις συστημάτων.

Οι έξυπνες κάρτες τρίτης γενιάς μπορούν να προγραμματιστούν για την εκτέλεση πολλών εφαρμογών. Υποστηρίζουν επίσης την εκτέλεση καταφορτώμενου κώδικα (downloadable code). Οι τεχνικές κρυπτανάλυσης που παρουσιάστηκαν εφαρμόζονται και σε αυτές. Εντοπίζουμε νέες και απλούστερες απειλές ασφάλειας στη χρήση έξυπνων καρτών τρίτης γενιάς σε ανοικτά και διαλειτουργικά περιβάλλοντα. Προτείνουμε μία αρχιτεκτονική, η οποία αντιμετωπίζει επιτυχώς αυτές τις απειλές.

Επόπτης

Απόστολος Τραγανίτης
Αναπληρωτής Καθηγητής
Τμήμα Επιστήμης Υπολογιστών

Πανεπιστήμιο Κρήτης

Επιβλέπων

Δημήτριος Ν. Σερπάνος
Αναπληρωτής Καθηγητής
Τμήμα Ηλεκτρολόγων Μηχανικών
και Τεχνολογίας Υπολογιστών
Πανεπιστήμιο Πατρών

# Security Analysis of Smart Card Systems

Artemios G. Voyiatzis

Master of Science Thesis

Department of Computer Science
University of Crete
Greece

**Abstract**

Smart cards are an ideal medium for representing natural entities in electronic transaction systems. Small size and easy of use facilitate their carry and usage everywhere by everyone. They also provide a high level of security, as they can store in a safe manner secret keys and also execute cryptographic algorithms.

*Side-channel cryptanalysis* is an innovative method for deriving secret keys stored in smart cards. This method is rather popular, since it can derive the secrets keys with minimum efforts.

We study the applicability of two variations of side channel cryptanalysis, namely passive and active hardware attacks, on the three smart card generations.

The key feature of first generation smart cards is hardware simpleness and processor absence. For this, they are referred as "memory cards". For security reasons, enough hardware logic has been incorporated to them, as to be capable for executing simple cryptographic algorithms. We developed a communication library for memory cards, using Gemplus GCR-410 smart card reader. Thus, we provide a common platform for realizing passive attacks on memory and smart cards. Furtherly, we propose a power consumption model adapted to memory cards. This model allows reconstruction of the secret cryptographic algorithms used in memory cards.

Smart cards are equipped with a processor and they are programmed for executing only one application. A common use of smart cards is the authentication of natural entities, utilizing proper cryptographic schemes, like the Fiat-Shamir identification scheme. We prove that there are systems configurations, where Bellcore's theoretical active attack model is not sound. We extend this model to make it sound in all-but-one cases. We propose a new active attack model, namely S-TAM. S-TAM claims stronger assumptions than Bellcore's attack. In return, it is provably sound in all system configurations.

Third generation smart cards can be programmed for executing multiple applications. They also support execution of download-able code. Presented techniques of cryptanalysis apply to them as well. We identify new and simpler security threats for third generation smart cards, when used in open and inter-operable systems. We propose an architecture, which defeats these threats.

Advisor

Apostolos Traganitis
Associate Professor
Department of Computer Science
University of Crete
Greece

Supervisor

Dimitrios N. Serpanos
Associate Professor
Department of Electrical and Computer Engineering
University of Patras
Greece

# Ευχαριστίες

*«Έλλησι τε και βαρβάροις, σοφοί τε και ανοήτοις οφειλέτης ειμί»*

*—Απόστολος Παύλος, Προς Ρωμαίους Α΄ 14*

Η εργασία αυτή αποτελεί το επιστέγασμα των σπουδών μου στο Πανεπιστήμιο Κρήτης. Για την την επίτευξη αυτού του στόχου είμαι οφειλέτης σε όλους όσους γνώρισα μέχρι σήμερα. Χωρίς να αμελώ τη συνεισφορά του κάθε ατόμου ξεχωριστά, θα ήθελα από το σημείο αυτό να εκφράσω τις ιδιαίτερες ευχαριστίες μου στους ανθρώπους που είχαν την κύρια συμβολή στην ολοκλήρωση της εργασίας αυτής.

Τον επόπτη, κύριο Απόστολο Τραγανίτη και τον επιβλέποντα και ακαδημαϊκό σύμβουλο, κύριο Δημήτρη Σερπάνο, για την καθοδήγηση και την υποστήριξή τους καθόλη τη διάρκεια των μεταπτυχιακών μου σπουδών. Τους καθηγητές του Τμήματος κυρίους Γιώργο Γεωργακόπουλο και Γιώργο Τζιρίτα για τη συμμετοχή τους στην εξεταστική επιτροπή της εργασίας.

Τον Βαγγέλη Αντωνιάδη· οι ομηρικές μας διαφωνίες ήταν από την πρώτη στιγμή εποικοδομητικές αφορμές για να εμβαθύνουμε στα θέματα ασφάλειας συστημάτων και κρυπτογραφίας που τόσο λατρεύουμε. Τη Σοφία Καρτσάκη· η ασυνέπειά της ήταν η αφορμή για να ολοκληρωθεί το πλέον σημαντικό κομμάτι της εργασίας. Τη Μαρία Γαμβρίλη, για τη φιλοξενία της καθώς και για τα σχόλια και τις υποδείξεις της στη σύνταξη της περίληψης της εργασίας. Το Μανόλη Σπανάκη για τη συνδρομή του στην εφαρμογή επίδειξης του Κεφαλαίου 3.

Το προσωπικό του Τμήματος Επιστήμης Υπολογιστών: τις κυρίες Ρένα Καλαϊτζάκη, Μαρία Σταυρακάκη, Χριστίνα Βαλίνι και Μαρία Μαμαλάκη καθώς και τον κύριο Γιάννη Σουρλατζή, για την πρόθυμη και άμεση βοήθεια που προσέφεραν για να ξεπερνώ απρόσκοπτα τα διαχειριστικά εμπόδια που καθυστερούν μία μεταπτυχιακή εργασία. Μακάρι να μπορούσα να πω το ίδιο και για τον ΕΛΚΕ.

Το Χρήστο Δήμου· ήταν πάντα εκεί, όποτε και αν χρειάστηκε.

Πάνω από όλα τους γονείς μου, Γιώργο και Ελένη· σε αυτούς οφείλω την ύπαρξη και την υπόστασή μου. Ελπίζω να σταθώ αντάξιος των θυσιών τους και να ανταποδίδω την αγάπη τους.

# Περιεχόμενα

# Κατάλογος Σχημάτων

# Κατάλογος Πινάκων

# Κεφάλαιο 1

# Εισαγωγή

*«Να ζούμε και να πεθαίνουμε για κάτι υψηλό, κάτι ικανό να κάνει τους ανθρώπους πιο ανθρώπους.»*

—*Αριστομένης Προβελέγγιος*

Η εποχή μας χαρακτηρίζεται από την ευρεία χρήση των νέων τεχνολογιών και υπηρεσιών που προσφέρουν οι πάσης φύσης υπολογιστικές συσκευές. Ο άνθρωπος επιθυμεί να παράγει και να ανταλλάσσει πληροφορία οπουδήποτε και οποτεδήποτε. Για το λόγο αυτό μεταφέρει καθημερινά μία ποικιλία συσκευών, όπως για παράδειγμα κινητό τηλέφωνο, πιστωτική κάρτα, κάρτα αναλήψεως χρημάτων, κάρτες πρόσβασης, ηλεκτρονικές ατζέντες (organizers) και υπολογιστές παλάμης (PDA, Personal Digital Assistant).

Κοινό γνώρισμα όλων αυτών των συσκευών είναι η δυνατότητά τους να αποθηκεύουν και να επεξεργάζονται πληροφορίες. Η προστασία όμως της ιδιωτικότητας του ατόμου (privacy) είναι στις μέρες μας αμφισβητήσιμη και θέατρο αντιπαράθεσης των υπερμάχων και των πολέμιων της εισαγωγής των νέων τεχνολογιών στην καθημερινότητά μας.

Οι «έξυπνες κάρτες» είναι μία τεχνολογία που γίνεται ολοένα και πιο δημοφιλής, ως ένα μέσο ασφαλούς αποθήκευσης και επεξεργασίας προσωπικών δεδομένων. Τα τελευταία χρόνια μία νέα κλάση επιθέσεων, οι επιθέσεις πλάγιου καναλιού, έχουν προταθεί στη βιβλιογραφία. Τα ερευνητικά αυτά αποτελέσματα θέτουν υπό αμφισβήτηση και επανεξέταση την ασφάλεια που προσφέρουν οι έξυπνες κάρτες, καθώς τα ιδιαίτερα γνωρίσματά τους παρέχουν γόνιμο έδαφος για την εφαρμογή τέτοιων επιθέσεων.

## 1.1 Περίληψη και συνεισφορά της εργασίας

Στην παρούσα εργασία εξετάζουμε την ασφάλεια συστημάτων έξυπνων καρτών σε περιβάλλοντα όπου μπορούν να υλοποιηθούν επιθέσεις πλάγιου καναλιού. Στη μελέτη μας κάνουμε ένα σαφή διαχωρισμό ανάμεσα σε τρεις γενιές έξυπνων καρτών:

- Κάρτες μνήμης νέας γενιάς
- «Κλασικές» έξυπνες κάρτες
- Έξυπνες κάρτες πολλαπλών εφαρμογών, με δυνατότητα ανανέωσης του κώδικά τους

Όπως θα δούμε και στη συνέχεια, οι κάρτες μνήμης δε δικαιολογούν απόλυτα τον ορισμό της έξυπνης κάρτας. Τα τελευταία χρόνια ωστόσο έχει ενσωματωθεί σε αυτές αρκετή λογική σε υλικό (hardware logic), ώστε να έχουν μία μορφή εξυπνάδας. Η εξυπνάδα αυτή είναι η μόνη δικλείδα ασφαλείας σε αρκετά συστήματα· θα δείξουμε όμως ότι δεν είναι αρκετή και κατά συνέπεια θα πρέπει σύντομα να εγκαταλειφθεί η χρήση τους.

Οι «κλασικές» έξυπνες κάρτες είναι κάρτες με μικροεπεξεργαστή, οι οποίες εκτελούν μία μόνο εφαρμογή. Σε πολλές από αυτές υλοποιούνται δικλείδες ασφαλείας με χρήση μηχανισμών κρυπτογραφίας. Οι πλέον ασφαλείς κρυπτογραφικοί μηχανισμοί έχουν αποδειχθεί, τόσο σε θεωρητικό, όσο και πρακτικό επίπεδο, ιδιαίτερα ευάλωτοι σε επιθέσεις πλάγιου καναλιού. Λόγω της φύσης τους, οι έξυπνες κάρτες είναι ο συνήθης στόχος αυτών των επιθέσεων. Εστιάζουμε στα σχήματα ταυτοποίησης, το πλέον διαδεδομένο κρυπτογραφικό μηχανισμό που χρησιμοποιείται σε συστήματα έξυπνων καρτών. Αποδεικνύουμε ότι η προσεκτική υλοποίηση του σχήματος Fiat-Shamir οδηγεί σε ικανοποιητική άμυνα απέναντι σε αυτές τις επιθέσεις. Προτείνουμε ωστόσο μία νέα μέθοδο επιθέσεων, η οποία είναι εγγυημένα έγκυρη. Συνεπώς ιδιαίτερη προσοχή θα πρέπει να δίνεται στην υλοποίηση του σχήματος.

Τα τελευταία χρόνια έχει προταθεί η χρήση έξυπνων καρτών, οι οποίες έχουν τη δυνατότητα να εκτελούν πολλές εφαρμογές και να ανανεώνεται ο κώδικας τους μετά την έκδοσή τους. Οι κάρτες αυτές μπορούν να αντιμετωπίσουν με μικρό κόστος επιθέσεις πλάγιου καναλιού. Η πλατφόρμα Java Card είναι η κυρία εκπρόσωπος αυτής της τεχνολογίας. Εντοπίσαμε προβλήματα ασφάλειας, τα οποία αφορούν συγκεκριμένους μηχανισμούς της πλατφόρμας. Τα προβλήματα αυτά μπορούν να χρησιμοποιηθούν για την υλοποίηση επιθέσεων σε ανοικτά και διαλειτουργικά περιβάλλοντα. Προτείναμε τέλος μία αρχιτεκτονική, η οποία αντιμετωπίζει επιτυχώς αυτά τα προβλήματα.

Η συνεισφορά της εργασίας μας είναι διττή: κατά πρώτον επεκτείναμε το χώρο εφαρμογής των επιθέσεων πλάγιου καναλιού σε δύο συγγενείς των «κλασικών» έξυπνων καρτών χώρους. Κατά δεύτερον αποδείξαμε ότι μία συγκεκριμένη επίθεση πλάγιου καναλιού, όπως έχει προταθεί στη βιβλιογραφία, δεν είναι έγκυρη σε όλα τα περιβάλλοντα και αντιπροτείναμε βελτιώσεις αυτής αλλά και μία νέα, η οποία είναι έγκυρη σε όλα τα περιβάλλοντα.

## 1.2 Η δομή της εργασίας

Η παρούσα αναφορά δομείται ως εξής: Στο κεφάλαιο 2 παρουσιάζουμε το θεωρητικό υπόβαθρο της εργασίας και δίνουμε τους απαραίτητους ορισμούς.

Στο κεφάλαιο 3 παρουσιάζουμε ένα θεωρητικό μοντέλο παθητικών επιθέσεων σε κάρτες μνήμης. Παρουσιάζουμε επίσης τη βιβλιοθήκη ανάγνωσης καρτών μνήμης για τον αναγνώστη Gemplus GCR-410, την οποίο αναπτύξαμε καθώς και την εφαρμογή επίδειξης λειτουργίας.

Στο κεφάλαιο 4 εστιάζουμε στις ενεργές επιθέσεις σε έξυπνες κάρτες. Πιο συγκεκριμένα, εξετάζουμε το σχήμα ταυτοποίησης Fiat-Shamir σε περιβάλλοντα ενεργών επιθέσεων και προτείνουμε μία άμυνα για την επίθεση Bellcore. Προτείνουμε επίσης μία βελτίωση της συγκεκριμένης επίθεσης, η οποία μπορεί να οδηγήσει σε επιτυχή αποτελέσματα. Τέλος, προτείνουμε ένα νέο μοντέλο ενεργών επιθέσεων, το S-TAM. Η επίθεση μπορεί να χρησιμοποιηθεί σε περιβάλλοντα όπου η επίθεση Bellcore ή η επαυξημένη επίθεση που προτείνουμε δεν είναι εφαρμόσιμες.

Στο κεφάλαιο 5 εστιάζουμε στην πλατφόρμα Java Card και γενικότερα στην τεχνολογία έξυπνων καρτών με δυνατότητα εκτέλεσης πολλαπλών εφαρμογών. Παρουσιάζουμε τα προβλήματα ασφάλειας που εντοπίσαμε σε ανοικτά και διαλειτουργικά περιβάλλοντα και τα νέα προβλήματα που δημιουργούνται σε μία αρχιτεκτονική που έχει προταθεί στη βιβλιογραφία. Τέλος, παρουσιάζουμε μία νέα αρχιτεκτονική, η οποία αντιμετωπίζει όλα τα προβλήματα που τέθηκαν.

Συνοψίζουμε στο κεφάλαιο 6, παραθέτοντας τα συμπεράσματα της παρούσας εργασίας καθώς και συγκεκριμένα μελλοντικά θέματα που σχετίζονται με αυτή.

# Κεφάλαιο 2

# Υπόβαθρο

*«Αν θέλεις να μη σε ξεχνούν γράφε πράγματα τα οποία αξίζει να διαβάζει κανείς ή κάνε πράγματα τα οποία αξίζουν τόσο ώστε να γράψει γι΄ αυτά κάποιος άλλος.»*

—*Βενιαμίν Φραγκλίνος*

Στο κεφάλαιο αυτό παραθέτουμε το υπόβαθρο της εργασίας μας. Αρχικά, κάνουμε μία εκτενή αναφορά στην τεχνολογία των έξυπνων καρτών, εστιάζοντας στα χαρακτηριστικά ασφάλειας που προσφέρουν και τα συγκριτικά πλεονεκτήματά τους. Κατόπιν δίνουμε τους αναγκαίους ορισμούς από την επιστήμη της κρυπτολογίας και παρουσιάζουμε τους αλγόριθμους και πρωτόκολλα κρυπτογραφίας που χρησιμοποιούνται σε σχήματα με έξυπνες κάρτες. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση των επιθέσεων πλάγιου καναλιού, μία τεχνική κρυπτανάλυσης που έχει προταθεί τελευταία στη βιβλιογραφία. Οι επιθέσεις πλάγιου καναλιού είναι ιδιαιτέρα δημοφιλείς σε έξυπνες κάρτες, καθώς οι υποθέσεις επιτυχούς υλοποίησης είναι αρκετά ασθενέστερες σε τέτοια περιβάλλοντα.

## 2.1 Έξυπνες κάρτες

Στην ενότητα αυτή παρουσιάζουμε τους διάφορους τύπους έξυπνων καρτών, εξηγούμε τα χαρακτηριστικά και πλεονεκτήματά τους και σχολιάζουμε τη χρήση συγκεκριμένων τύπων για συγκεκριμένες εφαρμογές. Κάνουμε επίσης μία μικρή εισαγωγή για το υλικό (hardware) των έξυπνων καρτών. Παρουσιάζουμε τέλος το λογισμικό και τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται.

### 2.1.1 Τί είναι μία έξυπνη κάρτα

Αρκετοί από εμάς χρησιμοποιούμε ήδη μία ή περισσότερες έξυπνες κάρτες στην καθημερινή μας ζωή. Έξυπνη κάρτα είναι μία τηλεκάρτα, μία κάρτα SIM


Σχήμα 2.1: Έξυπνη κάρτα πολλαπλών λειτουργιών της IBM

για το σύστημα κινητής τηλεφωνίας GSM, μία κάρτα που περιέχει τις πληροφορίες του συστήματος υγείας ή ένα ηλεκτρονικό πορτοφόλι. Το ηλεκτρονικό πορτοφόλι μας επιτρέπει να αποθηκεύσουμε ηλεκτρονικό «χρήμα», το οποίο να χρησιμοποιήσουμε αργότερα για μικροπληρωμές, όπως για παράδειγμα για την αγορά ενός εισιτηρίου λεωφορείου ή ενός αναψυκτικού από έναν αυτόματο πωλητή.

Η έξυπνη κάρτα είναι μία συσκευή, η οποία έχει τη δυνατότητα να αποθηκεύσει δεδομένα και να εκτελέσει εντολές. Είναι ένας μικροϋπολογιστής σε ένα ολοκληρωμένο κύκλωμα, με μέγεθος το πολύ 25 τετραγωνικά χιλιοστά. Ο μικροϋπολογιστής αυτός είναι πλαισιωμένος σε μία πλαστική κάρτα μεγέθους όσο μία συνήθης πιστωτική κάρτα. Στο σχήμα 2.1 βλέπουμε μία τέτοια έξυπνη κάρτα, της εταιρείας IBM. Οι πλαστικές κάρτες έχουν μία μακρά παράδοση.

### 2.1.2 Κάρτες μνήμης και κάρτες με μικροεπεξεργαστή

Οι πλαστικές κάρτες άρχισαν την πορεία τους στις αρχές της δεκαετίας του 1950, όταν ο οργανισμός Diners Club παρουσίασε την πρώτη πιστωτική κάρτα. Αυτή η κάρτα είχε το όνομα του/της[1] κατόχου της τυπωμένο στην εμπρόσθια όψη. Η χρήση της ήταν να επιβεβαιώσει ότι ο οργανισμός που την εξέδωσε είχε αρκετή εμπιστοσύνη στον κάτοχό της, ώστε να του επιτρέπει να αγοράζει αγαθά ή υπηρεσίες με πίστωση σε επιλεγμένα ξενοδοχεία και εστιατόρια. Ήταν πλέον ευκολότερο να πηγαίνει κανείς επαγγελματικά ταξίδια, χωρίς να ανησυχεί για το ποσό των χρημάτων που μπορεί να χρειάζονταν για την πληρωμή των λογαριασμών. Η επίδειξη της πιστωτικής κάρτας στο

[1] Για οικονομία λόγου, στο εξής οι όποιες αναφορές σε τρίτο πρόσωπο θα γίνονται χρησιμοποιώντας το αρσενικό πρόσωπο. Σε κάθε περίπτωση όμως θα εννοείται και το θηλυκό.

ξενοδοχείο ή το εστιατόριο ήταν αρκετή· το ποσό θα χρεωνόταν αργότερα στο λογαριασμό του ταξιδιώτη.

Αργότερα, παρουσιάστηκαν κάρτες με ανάγλυφη εκτύπωση. Το ανάγλυφο επέτρεπε την απευθείας αποτύπωση των στοιχείων του κατόχου, αντί να γίνεται χειροκίνητη καταγραφή. Μερικά χρόνια αργότερα, παρουσιάστηκε η μαγνητική λουρίδα (magnetic stripe), στην οποία αποθηκεύονταν οι πληροφορίες του λογαριασμού και το όνομα του κατόχου. Με την τεχνική αυτή η κάρτα έγινε μηχανικά αναγνώσιμη και κατά συνέπεια η πληροφορία μπορούσε να επεξεργαστεί ηλεκτρονικά. Απέμενε όμως ένα ακόμη πρόβλημα. Οποιοσδήποτε με τον κατάλληλο εξοπλισμό μπορούσε να διαβάσει και να γράφει τα δεδομένα στη μαγνητική λουρίδα. Αυτό οδηγούσε σε προβλήματα απάτης.

Το 1968 μία πατέντα για μία ταυτότητα με ενσωματωμένο κύκλωμα κατατέθηκε και η έξυπνη κάρτα γεννήθηκε. Η έξυπνη κάρτα είναι μία ασφαλής και ανθεκτική σε κακόβουλες επεμβάσεις (tamper-resistant) συσκευή. Τα δεδομένα που είναι αποθηκευμένα στην κάρτα μπορούν να προστατευθούν με μία μυστική πληροφορία (συνήθως ένα κωδικό), η οποία κοινοποιείται μόνο στον κάτοχο και στην έξυπνη κάρτα. Μόνο όποιος γνωρίζει το μυστικό μπορεί να χρησιμοποιήσει την κάρτα και τις πληροφορίες που είναι αποθηκευμένες μέσα σε αυτήν. Επιπλέον, με τη δυνατότητα να εκτελεί προγράμματα και εντολές, η έξυπνη κάρτα απέκτησε την ικανότητα να κρυπτογραφεί και αποκρυπτογραφεί πληροφορίες.

Η επιτυχία της έξυπνης κάρτας στην Ευρώπη ξεκίνησε στις αρχές της δεκαετίας του 1980. Την περίοδο 1982–1984, η Cartai Bancaire (η Ένωση Τραπεζικών Καρτών Γαλλίας), έτρεξε το πρώτο πιλοτικό πρόγραμμα. Σε συνεργασία με τις Bull, Philips και Schlumberger, η Cartai Bancaire έκανε τις πρώτες δοκιμές στις πόλεις Blois, Caen και Lyon της Γαλλίας. Οι δοκιμές αυτές είχαν τεράστια επιτυχία και μόνο ελάσσονα προβλήματα. Μία βελτίωση που προέκυψε από αυτές τις δοκιμές ήταν η ενσωμάτωση της μαγνητικής λουρίδας στην έξυπνη κάρτα, ώστε να διατηρηθεί η συμβατότητα με τα τότε συστήματα.

Μετά από αυτές τις δοκιμές, οι Γαλλικές τράπεζες εισήγαγαν τη χρήση των έξυπνων καρτών για τραπεζικές λειτουργίες. Αυτό ήταν η πρώτη μαζική δημόσια παρουσίαση των έξυπνων καρτών στη βιομηχανία των τραπεζικών λειτουργιών. Από τη δεκαετία του 1980 η βιομηχανία των έξυπνων καρτών έχει αναπτυχθεί. Οι Γερμανικές τράπεζες έχουν εκδώσει πάνω από 50 εκατομμύρια έξυπνες κάρτες GeldKarte στους πελάτες τους. Η GeldKarte είναι ένα ηλεκτρονικό πορτοφόλι, το οποίο θα δούμε στην ενότητα 2.4.1.

Στις μέρες μας κάθε κινητό τηλέφωνο, το οποίο είναι συμβατό με την προδιαγραφή GSM, περιέχει μία έξυπνη κάρτα, η οποία ταυτοποιεί τον κάτοχό της. Το σύστημα GSM χρησιμοποιείται πλέον τόσο στην Ευρώπη, όσο και στις άλλες ηπείρους.

### 2.1.3 Πλεονεκτήματα των έξυπνων καρτών

Η έξυπνη κάρτα είναι ουσιαστικά ένας φορητός υπολογιστής. Συνήθως έχει το σχήμα και το μέγεθος μίας πιστωτικής κάρτας. Μία έξυπνη κάρτα μπορεί να γίνει στο φυσικό επίπεδο ανθεκτική σε κακόβουλες επεμβάσεις (tamper-resistant). Επομένως, μπορεί να χρησιμοποιηθεί ως μονάδα αποθήκευσης υψηλής ασφάλειας για όλα τα είδη εμπιστευτικών πληροφοριών, όπως για παράδειγμα μυστικά κλειδιά. Η ασφάλεια και η μεταφερσιμότητα είναι δύο σημαντικοί λόγοι για τη χρήση των έξυπνων καρτών.

#### Παραδείγματα χρήσης

Στο παράδειγμα του ηλεκτρονικού πορτοφολιού, η κάρτα μπορεί να αποθηκεύσει το τρέχων υπόλοιπο και να ελέγχει αυστηρά την αύξηση ή μείωσή του. Τα ηλεκτρονικά πορτοφόλια που βασίζονται σε έξυπνες κάρτες μπορούν να μειώσουν σημαντικά το κόστος διαχείρισης των χρημάτων.

Ένα σημαντικό χαρακτηριστικό της έξυπνης κάρτας είναι ότι η πληροφορία που βρίσκεται μέσα της δε μπορεί να αντιγραφεί. Η μαγνητική λουρίδα μίας πιστωτικής κάρτας μπορεί εύκολα να αντιγραφεί και να χρησιμοποιηθεί εσφαλμένα. Αυτό δε θα μπορούσε ποτέ να συμβεί με μία πιστωτική κάρτα βασισμένη σε έξυπνη κάρτα. Για το λόγο αυτό, οι έξυπνες κάρτες έχουν αναγνωριστεί από μεγάλους τραπεζικούς οργανισμούς, όπως για παράδειγμα η Visa και η American Express, ως η νέα γενιά στις κάρτες οικονομικών συναλλαγών.

Σε ένα σύστημα ασφαλούς πρόσβασης σε ένα κτίριο, η κάρτα μπορεί να χρησιμοποιηθεί για να αποθηκεύσει τα δεδομένα που απαιτούνται για να ανοίξει μία πόρτα. Τα ίδια δεδομένα μπορούν αργότερα να ταυτοποιήσουν τον υπάλληλο στον υπολογιστή του. Μπορούν επίσης να χρησιμοποιηθούν για πληρωμές στην καφετέρια της εταιρείας.

Σε εφαρμογές home banking η κάρτα μπορεί να χρησιμοποιηθεί για την ταυτοποίηση του χρήστη πάνω από ένα δημόσιο δίκτυο δεδομένων, ανάμεσα στον υπολογιστή του χρήστη και το σύστημα της τράπεζας. Αυτό είναι ασφαλέστερο από τη συνήθη τακτική των ημερών μας, τα συνθηματικά (password).

Σε ένα σύστημα μαζικής μεταφοράς, η κάρτα μπορεί να αντικαταστήσει τα χάρτινα εισιτήρια. Το αντίτιμο μπορεί να υπολογιστεί με βάση την απόσταση. Αυτό μπορεί να γίνει την ώρα που ο πελάτης αποχωρεί από το μέσο μαζικής μεταφοράς και το αντίτιμο να αφαιρεθεί από την κάρτα στο σημείο εξόδου. Χρησιμοποιώντας μία ασύρματη κάρτα, ο ταξιδιώτης μπορεί να μη χρειαστεί να βγάλει καν την κάρτα από την τσέπη του.

Με όλα αυτά τα πλεονεκτήματα της έξυπνης κάρτας ως μίας ασφαλούς και φορητής συσκευής, θα βλέπουμε μία συνεχώς αυξανόμενη χρήση των έξυπνων καρτών. Σε λίγα χρόνια η έξυπνη κάρτα θα είναι μέρος της καθημερινότητάς μας και θα διευκολύνει σημαντικά τις καθημερινές μας εργασίες.

### 2.1.4 Το υλικό των έξυπνων καρτών

Η ενότητα αυτή άπτεται των γενικών αρχών που διέπουν το υλικό των έξυπνων καρτών και δίνουμε μία σφαιρική επισκόπηση της τεχνολογίας. Δεν καλύπτουμε την τεχνολογία υλικού εξαντλητικά. Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στο [45], το πλέον εξαιρετικό βιβλίο για αυτό το θέμα.

Τις πρώτες μέρες των έξυπνων καρτών, η κάρτα δεν ήταν στην πραγματικότητα έξυπνη, γιατί δεν είχε μικροεπεξεργαστή. Αυτές οι κάρτες αποκαλούνται «κάρτες μνήμης». Οι κάρτες μνήμης έχουν την ικανότητα να αποθηκεύουν ένα περιορισμένο ποσό πληροφορίας, όπως για παράδειγμα το όνομα του κατόχου τους. Το λογισμικό του υπολογιστή που επικοινωνεί με την κάρτα μπορεί να ανακτήσει αυτήν την πληροφορία.

Ανάλογα με τις απαιτήσεις ασφάλειας για τα αποθηκευμένα δεδομένα, η πρόσβαση στη μνήμη μπορεί να είναι προστατευμένη. Για παράδειγμα, η μνήμη μίας προπληρωμένης κάρτας τηλεφωνίας (γνωστή και ως «τηλεκάρτα» ) είναι προστατευμένη ώστε να αποτρέπει αυθαίρετη επαναφορά στη αρχική της κατάσταση. Κάρτες μνήμης χωρίς προστασία χρησιμοποιούνται σε μερικές εφαρμογές, οι οποίες δεν έχουν ιδιαίτερες απαιτήσεις ασφάλειας.

Οι «έξυπνες» κάρτες είναι κάρτες ολοκληρωμένου κυκλώματος με μικροεπεξεργαστή. Οι κάρτες αυτές μπορούν να χωριστούν σε έξυπνες κάρτες για μία συγκεκριμένη εφαρμογή και έξυπνες κάρτες πολλαπλών εφαρμογών. Στο σχήμα 2.2 φαίνεται ο διαχωρισμός αυτός.

### 2.1.5 Ασύρματες κάρτες

Για να χρησιμοποιηθεί μία έξυπνη κάρτα, ένας εξωτερικός υπολογιστής θα πρέπει να επικοινωνήσει με τον επεξεργαστή της κάρτας και να ανταλλάξει πληροφορίες και εντολές. Η επικοινωνία μπορεί να γίνει είτε με μετάδοση σήματος από τις επαφές της κάρτας, αν διαθέτει τέτοιες, όπως για παράδειγμα η κάρτα του σχήματος 2.1, είτε με ασύρματη μετάδοση. Υπάρχουν επίσης υβριδικές κάρτες, οι οποίες συνδυάζουν τις δύο τεχνολογίες και μπορούν να επικοινωνήσουν με οποιοδήποτε τρόπο.

Οι ασύρματες έξυπνες κάρτες χρησιμοποιούνται συχνά σε συστήματα όπου απαιτούνται ταχείς συναλλαγές ή όπου μόνο ένα περιορισμένο σύνολο δεδομένων πρέπει να μεταδοθεί. Παραδείγματα τέτοιων εφαρμογών είναι τα μέσα μαζικής μεταφοράς ή ο έλεγχος πρόσβασης σε κτίρια.

Οι κάρτες με επαφές είναι πιο δύσκολες στη χρήση τους από ότι οι ασύρματες. Στην πρώτη περίπτωση ο κάτοχος της κάρτας θα πρέπει να φροντίσει ότι εισάγει την κάρτα στον αναγνώστη με το σωστό τρόπο και από τη σωστή πλευρά. Επίσης, συχνά οι επαφές είναι η αιτία για αστοχίες, αν έχουν καταστραφεί, φθαρεί ή ακόμα αν δεν είναι αρκετά καθαρές.

Από την άλλη πλευρά, οι ασύρματες κάρτες έχουν επίσης κάποια μειονεκτήματα. Η ισχύς για τον επεξεργαστή πρέπει να μεταδοθεί ως ένα ηλε-


Σχήμα 2.2: Κατηγοριοποίηση καρτών με ολοκληρωμένο κύκλωμα

κτρομαγνητικό κύμα, το οποίο περιορίζει τη μέγιστη κατανάλωση ισχύος του επεξεργαστή και κατά συνέπεια τις δυνατότητες του επεξεργαστή. Επίσης, ο συνδυασμός αναλογικών και ψηφιακών τεχνολογιών κάνει τις ασύρματες κάρτες ακριβότερες από αντίστοιχες έξυπνες κάρτες με επαφές.

Παρά το ότι στις μέρες μας υπάρχουν συστήματα μαζικής μεταφοράς και ελέγχου πρόσβασης, τα οποία χρησιμοποιούν έξυπνες κάρτες, η τεχνολογία αυτή δε χρησιμοποιείται τόσο ευρέως, όσο των καρτών με επαφές. Η τάση αυτή ίσως αλλάξει στο κοντινό μέλλον. Οι ασύρματες κάρτες μπορεί να χρησιμοποιηθούν σε συστήματα πληρωμών.

### 2.1.6 Ο υπολογιστής της έξυπνης κάρτας

Όπως έχουμε εξηγήσει στα προηγούμενα, το πιο σημαντικό χαρακτηριστικό μίας έξυπνης κάρτας είναι το ότι περιέχει έναν υπολογιστή με Κεντρική Μονάδα Επεξεργασίας (CPU) και μνήμη. Σήμερα οι έξυπνες κάρτες έχουν περίπου την ίδια υπολογιστική ισχύ όπως ο πρώτος προσωπικός υπολογιστής της IBM.

Το ολοκληρωμένο μίας έξυπνης κάρτας αποτελείται από τον μικροεπεξεργαστή, ROM, EEPROM και RAM. Μία μνήμη EEPROM απαιτεί μεγαλύτερη επιφάνεια από μία PROM ίδιας χωρητικότητας. Για αυτό η EEPROM είναι πιο ακριβή και κάνει τη χωρητικότητα της EEPROM ένα σημαντικό παράγοντα για την τιμή μίας έξυπνης κάρτας.

Οι κάρτες μνήμης έχουν μόνο ROM και EEPROM. Δεν περιέχουν προγραμματιζόμενη λογική. Η EEPROM μπορεί να ασφαλισθεί με ένα κλειδί hard-wired, το οποίο ελέγχεται σε κάθε πρόσβαση.

### CPU

Σήμερα, οι περισσότερες έξυπνες κάρτες έχουν φθηνούς μικροεπεξεργαστές των 8 bit, αλλά στα κορυφαία μοντέλα καρτών μπορούμε να βρούμε επεξεργαστών των 16 ή και 32 bit.

### Κρυπτογραφικός συνεπεξεργαστής

Ένας προαιρετικός κρυπτογραφικός συνεπεξεργαστής αυξάνει την απόδοση των κρυπτογραφικών λειτουργιών. Εκτελώντας λειτουργίες υπογραφών μέσα στην ίδια την κάρτα, το ιδιωτικό κλειδί του χρήστη δε χρειάζεται ποτέ να βγει έξω από την κάρτα.

### ROM

Οι πληροφορίες που αποθηκεύονται στη ROM γράφονται κατά την παραγωγή της κάρτας. Είναι ίδιες για όλα τα ολοκληρωμένα μίας σειράς. Περιέχει το λειτουργικό σύστημα της κάρτας και ίσως μερικές εφαρμογές.

### EEPROM

Η EEPROM χρησιμοποιείται για τη μόνιμη αποθήκευση δεδομένων. Ακόμη και αν η κάρτα αποσυνδεθεί από την τροφοδοσία, η EEPROM διατηρεί τα δεδομένα. Μερικές κάρτες επιτρέπουν την αποθήκευση επιπλέον προγραμμάτων ή συγκεκριμένα τμήματα προγραμμάτων στην EEPROM.

### RAM

Η RAM είναι η παροδική μνήμη της κάρτας και κρατά τα δεδομένα της μόνο όσο η κάρτα τροφοδοτείται.

## 2.1.7 Μηχανικές επαφές

Μία έξυπνη κάρτα με επαφές έχει το πολύ οκτώ μηχανικές επαφές, οι οποίες χρησιμοποιούνται για να παρέχεται ισχύς και το σήμα ρολογιού στην κάρτα, καθώς και για τη μετάδοση δεδομένων. Το πρότυπο ISO 7816-2 καθορίζει λεπτομερώς τη θέση, το ελάχιστο μέγεθος και τη χρήση των επαφών αυτών.

Δύο από τις οκτώ επαφές είναι δεσμευμένες για μελλοντική χρήση και οι έξι χρησιμοποιούνται πραγματικά. Οι έξι ενεργές επαφές είναι:

- Η `Vcc` χρησιμοποιείται για την παροχή τάσης στο ολοκληρωμένο. Γενικά, η τάση είναι μεταξύ $4.5V$ και $5.5V$. Υπάρχουν ολοκληρωμένα που λειτουργούν με τάση περίπου $3V$.

- Η `RST` χρησιμοποιείται για το σήμα επαναφοράς (reset signal ώστε να επανέλθει η αρχική τιμή του μετρητή διευθύνσεων και του μικροεπεξεργαστή. Μία επαναφορά χωρίς τη διακοπή τροφοδοσίας καλείται warm reset. Αν η τροφοδοσία αφαιρεθεί και εφαρμοσθεί ξανά, η επαναφορά καλείται cold reset.

- Η `CLK` χρησιμοποιείται για την εφαρμογή του εξωτερικού παλμού ρολογιού, από το οποίο παράγεται ο χρονισμός του μικροεπεξεργαστή.

- Η `GND` χρησιμοποιείται για τη σύνδεση με τη γη.

- Η `Vpp` χρησιμοποιείται για το σήμα υψηλής τάσης, το οποίο απαιτείται για να προγραμματιστεί η EEPROM. Οι νεότερες κάρτες έχουν τη δυνατότητα να παράγουν αυτήν την υψηλή τάση εσωτερικά, ώστε να μη χρειάζεται να χρησιμοποιείται η επαφή αυτή.

- Η `I/O` χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ της έξυπνης κάρτας και της συσκευής ανάγνωσης με ένα τρόπο μονής κατεύθυνσης (half-duplex). Τα πρωτόκολλα για αυτήν την επικοινωνία περιγράφονται επίσης στη σειρά των προτύπων ISO 7816.

### 2.1.8 Φυσικές διαστάσεις

Αρκετές μορφές και μεγέθη έχουν καθοριστεί λεπτομερώς για τις πλαστικές κάρτες. Οι δύο πιο σημαντικές φόρμες για τις έξυπνες κάρτες είναι η ID-1 και η ID-000. Οι ακριβείς διαστάσεις και παρόμοιες λεπτομέρειες περιγράφονται επίσης στη σειρά των προτύπων ISO 7816. Η φόρμα ID-1 έχει το μέγεθος μίας πιστωτικής κάρτας και χρησιμοποιείται στα περισσότερα συστήματα έξυπνων καρτών. Η φόρμα ID-000 χρησιμοποιείται κυρίως στα κινητά τηλέφωνα και σε Αρθρώματα Ασφαλούς Πρόσβασης (SAM, Security Access Module) σε τερματικά. Οι κάρτες με φόρμα ID-1 δε χωράνε σε ένα μοντέρνο κινητό τηλέφωνο. Στην πραγματικότητα, πολλές κάρτες φθάνουν στον τελικό χρήστη σε φόρμα ID-1, με εγκοπές στα κατάλληλα σημεία. Ο χρήστης μπορεί να αποσπάσει στη συνέχεια την κάρτα, η οποία έχει πλέον φόρμα ID-000.

Ο κύριος λόγος για το μέγεθος και το σχήμα της έξυπνης κάρτας είναι το να είναι γεωμετρικά συμβατή με την κάρτα με μαγνητική λουρίδα. Αυτό επιτρέπει τη χρήση της ίδιας κάρτας είτε ως έξυπνης κάρτας, σε νεότερα συστήματα, είτε ως μαγνητική κάρτας σε παγιωμένες (legacy) εφαρμογές.

Σε εφαρμογές όπου η συμβατότητα με τις προτυποποιημένες μαγνητικές κάρτες δεν απαιτείται, υπάρχουν διαθέσιμες μικρές κρυπτογραφικές μονάδες, οι οποίες είναι ασφαλείς στο φυσικό επίπεδο, σε διάφορα σχήματα. Ένα τέτοιο παράδειγμα είναι το iButton της Dallas Semiconductors, το οποίο έχει σχήμα κουμπιού και δακτυλιδιού.

### 2.1.9 Ασφάλεια στο επίπεδο υλικού

Ο στόχος της σχεδίασης ενός ολοκληρωμένου έξυπνης κάρτας είναι να παρέχει υψηλή φυσική ασφάλεια για τα δεδομένα που αποθηκεύονται μέσα στην κάρτα. Ο επεξεργαστής και η μνήμη συνδυάζονται στο ίδιο ολοκληρωμένο. Αυτό κάνει δύσκολη την υποκλοπή των σημάτων που ανταλλάσσονται ανάμεσα στον επεξεργαστή και τη μνήμη.

Η σχεδίαση ενός ολοκληρωμένου έξυπνης κάρτας λαμβάνει υπόψη του πολλές ακόμα πιθανές απειλές. Για ένα πλήρη κατάλογο, παραπέμπουμε και πάλι στο [45]. Αναφέρουμε ενδεικτικά τη διαδοχική αφαίρεση των επιπέδων του ολοκληρωμένου, ώστε να διαβάζονται οπτικά τα δεδομένα, τον επιδέξιο χειρισμό της τάσης λειτουργίας ή του ρολογιού, ώστε να αποτύχει ο επεξεργαστής. Επίσης υπάρχουν επιθέσεις που χρησιμοποιούν υψηλή θερμοκρασία, ακτίνες X και υπεριώδη ακτινοβολία.

Πολύπλοκα αντίμετρα εφαρμόζονται ώστε να προστατευθεί το ολοκληρωμένο από τις διάφορες επιθέσεις. Για παράδειγμα, ενσωματώνονται στρώματα προστασίας των ημιαγωγών με επικάλυψη (passivation layer) για να αποτρέψουν την περαιτέρω ανάλυση του ολοκληρωμένου. Οι γραμμές διεύθυνσης και τα κελιά της μνήμης διατάσσονται σε ασυνήθιστα μοτίβα, ώστε να είναι δυσκολότερη η φυσική τους εξέταση. Επιπλέον μερικά ολοκληρωμένα έχουν τη δυνατότητα να αντιληφθούν αν ένα στρώμα πάνω από αυτό έχει αφαιρεθεί, όπως θα συνέβαινε στην περίπτωση που κάποιος προσπαθεί να εξετάσει το ολοκληρωμένο. Τα ολοκληρωμένα μπορούν να ανιχνεύσουν ασυνήθιστες διακυμάνσεις στους παλμούς του ρολογιού ή στην τροφοδοσία τάσης και να αντιδράσουν με διακοπή της λειτουργίας τους.

### 2.1.10 Η διαδικασία κατασκευής

Συνήθως, τα εξής βήματα ακολουθούνται για να παραχθεί και να εκδοθεί μία έξυπνη κάρτα:

- Κατασκευή του ολοκληρωμένου

- Ο δίσκος πυριτίου με τα ολοκληρωμένα ελέγχεται και σημειώνονται τα σωστά ολοκληρωμένα. Ο δίσκος τεμαχίζεται σε ξεχωριστά ολοκληρωμένα. Το κάθε ολοκληρωμένο συνδέεται με το μεταλλικό έλασμα, που περιέχει τις οκτώ επαφές, ώστε να δημιουργηθεί ένα άρθρωμα (module).

- Η χημική ένωση PVC χρησιμοποιείται τις περισσότερες φορές ως υλικό κατασκευής της πλαστικής κάρτας. Στο βήμα αυτό εκτυπώνονται στις δύο όψεις της κάρτας όσα κείμενα και γραφικά για τη συγκεκριμένη σειρά καρτών.

- Δημιουργείται ένα βαθούλωμα ή τρύπα στο πλαστικό και εφαρμόζεται το άρθρωμα.

- Αρχικοποίηση. Τα αρχεία δεδομένων και τα προγράμματα, που είναι κοινά σε μία σειρά καρτών, φορτώνονται στην EEPROM της έξυπνης κάρτας.

- Προσωποποίηση. Σε αυτό το βήμα οι προσωπικές πληροφορίες του κατόχου φορτώνονται στην EEPROM της έξυπνης κάρτας και εκτυπώνονται στις δύο της όψεις. Τα δεδομένα αυτά είναι συνήθως το όνομα του κατόχου και ο αριθμός λογαριασμού του ή τα μυστικά του κλειδιά. Η εκτύπωση των στοιχείων συνήθως συμπεριλαμβάνει το όνομα του κατόχου και μερικές φορές τη φωτογραφία του.

## 2.2 Λογισμικό έξυπνων καρτών

Στην ενότητα αυτή παρουσιάζουμε τις βασικές αρχές της ανάπτυξης λογισμικού για έξυπνες κάρτες. Θα εξηγήσουμε τη διάκριση μεταξύ των τμημάτων on-card και off-card μίας εφαρμογής και θα εξηγήσουμε τον τρόπο που αυτά τα δύο τμήματα αλληλεπιδρούν.

### 2.2.1 Ανάπτυξη εφαρμογών για έξυπνες κάρτες

Συνήθως μία εφαρμογή για έξυπνες κάρτες αποτελείται από δύο τμήματα, το on-card και το off-card. Το τμήμα off-card της εφαρμογής είναι το τμήμα που βρίσκεται στον υπολογιστή ή το τερματικό, το οποίο συνδέεται με την έξυπνη κάρτα μέσω μίας συσκευής ανάγνωσης έξυπνων καρτών. Το τμήμα αυτό μπορεί να αναπτυχθεί σε οποιαδήποτε γλώσσα προγραμματισμού υποστηρίζεται στον υπολογιστή, αρκεί να υπάρχει το κατάλληλο λογισμικό διασύνδεσης με την κάρτα μέσω του αναγνώστη.

Το on-card τμήμα της εφαρμογής είναι αποθηκευμένο στο ολοκληρωμένο της έξυπνης κάρτας. Το τμήμα αυτό αποτελείται από δεδομένα και ίσως από εκτελέσιμο κώδικα. Αν περιέχει και εκτελέσιμο κώδικα, ο κώδικας αυτός εκτελείται από το λειτουργικό σύστημα της κάρτας και μπορεί να χρησιμοποιήσει τις υπηρεσίες του λειτουργικού συστήματος.

Οι περισσότεροι κατασκευαστές έξυπνων καρτών προσφέρουν toolkit ή σύνολα εργαλείων ανάπτυξης λογισμικού, τα οποία υποστηρίζουν την ανάπτυξη του τμήματος on-card της εφαρμογής.

Τα υπάρχοντα εργαλεία προγραμματισμού είναι συνήθως χαμηλού επιπέδου και αναφέρονται αποκλειστικά σε συγκεκριμένα μοντέλα καρτών συγκεκριμένου κατασκευαστή. Αυτό έχει ως συνέπεια ότι τελικά η ανάπτυξη των συγκεκριμένων τμημάτων των εφαρμογών γίνεται από τους προγραμματιστές του λειτουργικού συστήματος της κάρτας, γιατί είναι οι μόνοι που μπορούν να γνωρίζουν τις τεχνικές λεπτομέρειες της κάθε κάρτας. Ο κώδικας αυτός πρέπει να ενσωματωθεί στη συνέχεια στη μάσκα της ROM της έξυπνης κάρτας στο βήμα της κατασκευής του ολοκληρωμένου.

Τα λειτουργικά συστήματα έξυπνων καρτών τελευταίας γενιάς επιτρέπουν στους προγραμματιστές των εφαρμογών να δημιουργούν και να φορτώνουν μόνοι τους το on-card τμήμα της εφαρμογής. Τα σημαντικότερα από αυτά τα λειτουργικά συστήματα είναι τα Java Card, Multos και Smart Card for Windows.

### 2.2.2 Λειτουργικά συστήματα

Στην ενότητα αυτή παρουσιάζουμε τις βασικές αρχές των έξυπνων καρτών με σύστημα αρχείων (file system smart card). Εστιάζουμε στη συνέχεια στα λειτουργικά συστήματα νέας γενιάς, τα οποία υποστηρίζουν ανεξάρτητη ανάπτυξη και φόρτωση κώδικα εφαρμογών. Τα νέα λειτουργικά συστήματα, Java Card, Multos και Smart Card for Windows είναι οι κύριοι υποψήφιοι για σενάρια πολλαπλών εφαρμογών.

#### Έξυπνες κάρτες με σύστημα αρχείων

Το πρότυπο ISO 7816-4 ορίζει το σύστημα αρχείων μίας έξυπνης κάρτας. Ένα τέτοιο σύστημα αρχείων χτίζεται βασιζόμενο στα εξής τρία συστατικά (βλ. σχήμα 2.3):

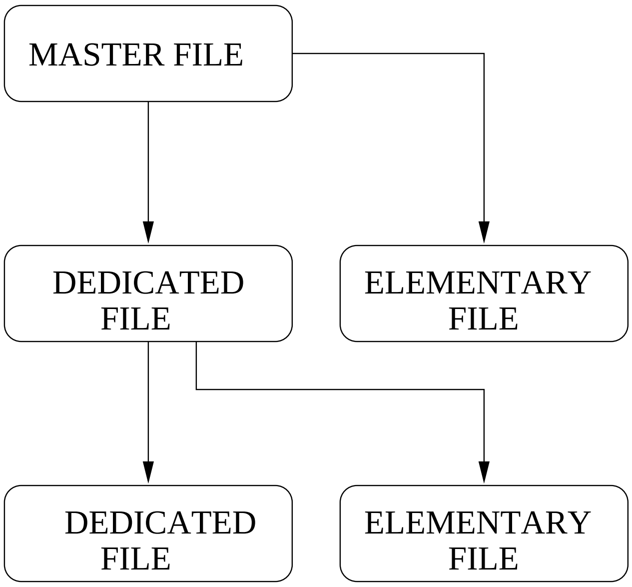
- Elementary file (EF)
- Dedicated file (DF)
- Master file (MF)

Κάθε αρχείο ορίζεται είτε από ένα αναγνωριστικό αρχείου 2 byte είτε από ένα συμβολικό όνομα.

Ένα elementary file μπορεί να περιέχει μόνο δεδομένα. Το μέγιστο μέγεθος ενός EF πρέπει να καθορίζεται όταν αυτό δημιουργείται. Εξαιτίας αυτού καθώς και άλλων περιορισμών είναι πολύ σημαντικό να αποφασίζεται η δομή του αρχείου πολύ προσεκτικά. Υπάρχουν τέσσερις διαφορετικοί τύποι EF:

- Transparent file
- Linear, variable length record file
- Linear, fixed length record file
- Cyclic, fixed length record file

Ένα dedicated file είναι ανάλογο με την έννοια του καταλόγου (directory) ενός συστήματος αρχείων, όπως το γνωρίζουμε στο UNIX. Ένα DF μπορεί να περιέχει EF και άλλα DF. Τα DF χρησιμοποιούνται για να διαχωρίζουν τα δεδομένα διαφορετικών εφαρμογών. Κάθε εφαρμογή συνήθως έχει το δικό της DF. Βασιζόμενοι σε αυτήν την αρχή είναι δυνατό να κτίσουμε μία δομή καταλόγων όπως στο UNIX.




Σχήμα 2.3: Σύστημα αρχείων έξυπνης κάρτας

Το master file είναι η ρίζα του συστήματος αρχείων. Όπως ένα DF, περιέχει DF και EF. Υπάρχει μόνο ένα MF σε κάθε έξυπνη κάρτα. Τα EF ενός MF συνήθως χρησιμοποιούνται για την αποθήκευση δεδομένων και κλειδιών, τα οποία είναι προσβάσιμα από διάφορες εφαρμογές της κάρτας. Κατά το πρότυπο ISO 7816-4, το αναγνωριστικό του MF είναι `0x3F00` σε κάθε σύστημα αρχείων έξυπνης κάρτας συμβατής με το πρότυπο ISO.

Το πρότυπο ISO 7816-4 καθορίζει ένα σύνολο από εντολές για ένα σύστημα αρχείων, όπως:

- Select File
- Read Binary
- Write Binary
- Update Binary
- Append Record

Η πρόσβαση στα δεδομένα των αρχείων (EF) ελέγχεται από συνθήκες πρόσβασης. Προτού μία συγκεκριμένη λειτουργία να μπορεί να εκτελεστεί σε ένα αρχείο, οι συνθήκες πρόσβασης για τη συγκεκριμένη λειτουργία και το συγκεκριμένο αρχείο πρέπει να ικανοποιηθούν. Αυτές οι συνθήκες πρόσβασης δεν ορίζονται στο πρότυπο ISO 7816-4 και διαφέρουν σημαντικά στις τρέχουσες υλοποιήσεις λειτουργικών συστημάτων.

Παραδείγματα μερικών συνθηκών πρόσβασης, οι οποίες ορίζονται στις κάρτες IBM Multi-Function (η κάρτα του σχήματος 2.1), είναι:

- Always: Η συγκεκριμένη λειτουργία μπορεί να εκτελεστεί στο συγκεκριμένο αρχείο χωρίς να απαιτείται η ικανοποίηση οποιασδήποτε συνθήκης πρόσβασης.
- Protected: Ένας Κώδικας Αυθεντικότητας Μηνύματος (message authentication code, MAC), ο οποίος ταιριάζει με την εντολή και δημιουργήθηκε με το μυστικό κλειδί, ασφαλίζει τη συγκεκριμένη λειτουργία στο συγκεκριμένο αρχείο. Το κλειδί που απαιτείται για το MAC, θα μπορούσε να είναι διαφορετικό για κάθε λειτουργία για το συγκεκριμένο αρχείο.
- External Authentication: Προτού η συγκεκριμένη λειτουργία μπορεί να εκτελεστεί στο συγκεκριμένο αρχείο, η κάρτα ελέγχει αν η εξωτερική (off-card) εφαρμογή μοιράζεται ένα κοινό μυστικό με την κάρτα.

Η χρήση MAC και η εξωτερική ταυτοποίηση είναι κλασικές μέθοδοι που χρησιμοποιούνται ευρέως στο λογισμικό των έξυπνων καρτών.

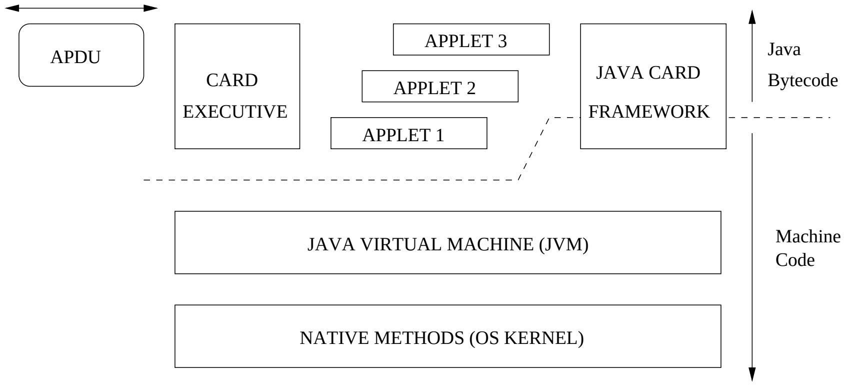
**Java Card**

Η πλατφόρμα Java Card [55] επιτρέπει την ανάπτυξη του τμήματος on-card μίας εφαρμογής να γραφτεί σε Java. Με τον τρόπο αυτό τα πλεονεκτήματα της γλώσσας προγραμματισμού Java παρέχονται στην ανάπτυξη λογισμικού για έξυπνες κάρτες. Επιπλέον, παρέχει μία καλή βάση για κάρτες πολλαπλών εφαρμογών, όπου υποστηρίζεται πάνω από μία εφαρμογή στη ίδια κάρτα. Η προδιαγραφή Java Card ανήκει στην εταιρεία SUN· επεκτάσεις και βελτιώσεις της προδιαγραφής συνεισφέρονται από τον οργανισμό Java Card Forum[24].

Ο εκτελέσιμος κώδικας της κάρτας αποτελείται από byte codes, τα οποία διερμηνεύονται από το Java Card Runtime Environment, JCRE [52]. Το JCRE ελέγχει επίσης την εκτέλεση των διαφόρων εφαρμογών και εγγυάται ότι αυτές οι εφαρμογές δε συγκρούονται μεταξύ τους. Ο στόχος της προδιαγραφής Java Card είναι ότι οι εφαρμογές μπορούν να εκτελεστούν σε οποιαδήποτε έξυπνη κάρτα την υποστηρίζει. Ο στόχος αυτός δεν έχει επιτευχθεί πλήρως ακόμη, γιατί οι τρέχουσες υλοποιήσεις διαφέρουν σε λεπτομέρειες από την τρέχουσα προδιαγραφή και είναι μερικώς ασύμβατες μεταξύ τους.

Κατά αυστηρό ορισμό, η Java Card είναι μία πλατφόρμα αλλά όχι ένα λειτουργικό σύστημα. Η πλατφόρμα αυτή υπάρχει σε ένα λειτουργικό σύστημα, το οποίο δεν είναι άμεσα προσβάσιμο από τις εφαρμογές. Οι εφαρμογές γράφονται με βάση τις διεπαφές που παρέχει η πλατφόρμα, όπως αντίστοιχα θα χρησιμοποιούσαν υπηρεσίες του λειτουργικού συστήματος.

Μία εφαρμογή on-card γραμμένη σε Java Card καλείται «card applet». Έχει ένα αναγνωριστικό εφαρμογής (Application Identifier, AID), όπως περιγράφεται στο πρότυπο ISO 7816-5. Το AID είναι ένα μοναδικό αναγνωριστικό




Σχήμα 2.4: Στοίβα λογισμικού μίας κάρτας Java Card

για κάθε εφαρμογή. Αποτελείται από δύο μέρη, όπου το πρώτο μέρος είναι παγκοσμίως καταχωρημένο στην εταιρεία στην οποία ανήκει η εφαρμογή.

Δε θα έπρεπε να προκαλεί έκπληξη ότι το υποσύνολο της γλώσσας Java, το οποίο μπορεί να χρησιμοποιηθεί σε μία έξυπνη κάρτα, έχει περιορισμούς οι οποίοι δεν είναι γνωστοί στη γλώσσα Java για προσωπικούς ή δικτυακούς υπολογιστές. Μερικές από τις διαφορές μεταξύ της προδιαγραφής Java Card και της προδιαγραφής της πλήρους γλώσσας Java είναι οι παρακάτω.

Η δυναμική φόρτωση κλάσεων δεν υποστηρίζεται. Η δυνατότητα για garbage collection δεν είναι υποχρεωτική. Δεν υπάρχει κλάση Security Manager–οι πολιτικές ασφάλειας υλοποιούνται κατευθείαν στην Εικονική Μηχανή Java (Java Virtual Machine, JVM) [53]. Η JVM δεν υποστηρίζει πολλαπλές ίνες. Τα αντικείμενα (objects) δε μπορούν να κλωνοποιηθούν. Το πακέτο `Java.lang` διαφέρει σημαντικά· για παράδειγμα η κλάση `String` δεν υφίσταται. Οι βασικοί τύποι `char, double, float` και `long` δεν είναι διαθέσιμοι, ο τύπος `int` και ακέραιοι των 32 bit υποστηρίζονται μόνο προαιρετικά.

Παρ᾽ όλα αυτά μπορεί να χρησιμοποιηθεί ένας κοινός μεταγλωττιστής Java για μεταγλωττιστούν πηγαία αρχεία Java Card σε byte code. Αντί για της πρότυπες κλάσεις του JDK, μπορεί να χρησιμοποιηθεί το Java Card framework κατά τη μεταγλώττιση. Ο κώδικας που προκύπτει, δε μπορεί να εκτελεστεί απευθείας στην κάρτα. Για λόγους απόδοσης, θα πρέπει πρώτα να μετατραπεί και να συνδεθεί στατικά (static linking).

Η στοίβα λογισμικού μίας Java Card περιγράφεται στο σχήμα 2.4. Το JCRE έχει τις εξής διεπαφές: το Card Executive διαχειρίζεται την κάρτα και είναι ο σύνδεσμος επικοινωνίας μεταξύ του card applet και του τμήματος off-card της εφαρμογής. Το JVM εκτελεί το bytecode της εφαρμογής και των συναρτήσεων βιβλιοθήκης που χρησιμοποιεί. Το Java Card framework παρέχει τις συναρτήσεις βιβλιοθήκης. Αυτές ορίζουν το πρότυπο Java Card API [54]. Ο πυρήνας του λειτουργικού συστήματος και το JVM είναι σε γλώσσα μηχανής, τα παραπάνω επίπεδα είναι γραμμένα σε Java bytecode.

### Multos

Το Multos είναι ένα από τα λειτουργικά συστήματα πολλαπλών εφαρμογών για έξυπνες κάρτες, το οποίο επιτρέπει τη φόρτωση ή αναβάθμιση μίας εφαρμογής στην κάρτα, αφού αυτή έχει εκδοθεί. Το πρότυπο Multos ανήκει στο βιομηχανικό consortium Maosco [36].

Μία κάρτα Multos παρέχει μία διεπαφή για ένα σύστημα αρχείων συμβατό με το πρότυπο ISO και επιπρόσθετα ένα περιβάλλον εκτέλεσης για εφαρμογές τρίτων. Οι προγραμματιστές των εφαρμογών μπορούν να δημιουργήσουν αυτές τις εφαρμογές χρησιμοποιώντας μία νέα γλώσσα, την Multos Executable Language, (MEL).

Το Multos καθορίζει ένα κοινό λειτουργικό σύστημα με ένα API, το οποίο καλείται Application Abstract Machine (AAM). Αυτό εγγυάται ότι μετά τη δημιουργία μία νέας εφαρμογής Multos, αυτή η εφαρμογή μπορεί να φορτωθεί σε οποιαδήποτε έξυπνη κάρτα Multos χωρίς περαιτέρω επεξεργασία. Το AAM βρίσκεται πάνω από το λειτουργικό σύστημα Multos και είναι ανεξάρτητο του υλικού. Διερμηνεύει την εφαρμογή που είναι γραμμένη MEL και τη μεταφράζει σε συγκεκριμένες εντολές του λειτουργικού συστήματος, τις οποίες στη συνέχεια εκτελεί.

Το Multos χρησιμοποιείται σαν ένα λειτουργικό σύστημα για το σύστημα ηλεκτρονικού πορτοφολιού Mondex, το οποίο θα δούμε στην ενότητα 2.4.1.

### Smart Card for Windows

Τον Οκτώβριο του 1998, η εταιρεία Microsoft ανακοίνωσε την είσοδό της στην αγορά των λειτουργικών συστημάτων για έξυπνες κάρτες, με ένα νέο προϊόν με όνομα Smart Card for Windows.

Η κάρτα είναι ένας συνδυασμός ενός παραδοσιακού λειτουργικού συστήματος συμβατού με το πρότυπο ISO 7816-4 και μίας προγραμματιζόμενης πλατφόρμας. Ο πυρήνας παρέχει ένα λειτουργικό σύστημα, έλεγχο πρόσβασης, μηχανισμούς εισόδου-εξόδου, υπηρεσίες κρυπτογραφίας, ένα API και μία επιλογή από εντολές ISO. Ένα προαιρετικό περιβάλλον εκτέλεσης (runtime environment) επιτρέπει την προσθήκη εφαρμογών που έχουν αναπτυχθεί από τρίτους.

Η Microsoft παρέχει στους προγραμματιστές όλα τα απαραίτητα συστατικά για να αναπτύξουν μία εφαρμογή. Με το προϊόν Smart Card for Windows, η Microsoft επιτρέπει σε άλλες εταιρείες να δημιουργήσουν ένα λειτουργικό σύστημα για έξυπνες κάρτες χρησιμοποιώντας ένα toolkit για έξυπνες κάρτες βασισμένο στο περιβάλλον Windows σε Visual Basic ή Visual C++. Χρησιμοποιώντας το toolkit αυτό, ο προγραμματιστής μπορεί να επιλέξει τα επιθυμητά συστατικά του λειτουργικού συστήματος και μετά να δημιουργήσει μία μάσκα, η οποία μπορεί να φορτωθεί στη συνέχεια στην έξυπνη κάρτα.

Το σύνολο εντολών είναι προσαρμοζόμενο. Η πρώτη έκδοση του Smart Card for Windows παρέχει εντολές συμβατές με τα πρότυπα ISO και EM-

V [17]· η υποστήριξη για εντολές GSM προβλέπεται σε μία επόμενη έκδοση.

Το σύστημα αρχείων του Smart Card for Windows βασίζεται σε μία συγχωνευμένη έκδοση του συστήματος αρχείων DOS FAT. O αριθμός των κατατμήσεων και και το μέγεθος κάθε κατάτμησης μπορεί να οριστεί κατά τη δημιουργία τους. Ένα επεκτάσιμο σύνολο υπηρεσιών κρυπτογραφίας είναι επίσης διαθέσιμο. Το εσωτερικό API είναι ανεξάρτητο γλώσσας προγραμματισμού. Η πρώτη έκδοση του περιβάλλοντος εκτέλεσης είναι ένας διερμηνευτής της Visual Basic· η υποστήριξη για τη C++ είναι μέσα στα πλάνα. Αυτό το περιβάλλον εκτέλεσης παρέχει την ελαστικότητα και τα πλεονεκτήματα που συναντάμε στα Java Card και Multos.

### 2.2.3 Αναγνώστες και τερματικά

Για να διαβαστούν και να γραφούν δεδομένα σε μία έξυπνη κάρτα ή για να εκτελεστεί μία εντολή σε αυτή, είναι απαραίτητο να έχουμε μία φυσική σύνδεση με την κάρτα. Για να γίνει η σύνδεση με μία κάρτα επαφών, θα πρέπει να εισαχθεί σε μία συσκευή αποδοχής έξυπνων καρτών (smart card acceptance device). Στην ενότητα αυτή περιγράφουμε τους διάφορους τύπους τέτοιων συσκευών. Στην αγορά υπάρχει μία ευρεία ποικιλία συσκευών αποδοχής. Μπορούμε να τις διαχωρίσουμε σε δύο σύνολα:

- Αναγνώστες (readers) και
- Τερματικά (terminals).

Ένας αναγνώστης έξυπνων καρτών είναι κατά βάση ένας συνδετήρας ανάμεσα στην έξυπνη κάρτα και στη συσκευή που επικοινωνεί με την κάρτα. Σε αντίθεση με αυτό, ένα τερματικό είναι ένας υπολογιστής από μόνο του και μπορεί να λειτουργήσει ανεξάρτητα, χωρίς να έχει συνδεθεί με μία άλλη συσκευή.

Ένας αναγνώστης μπορεί συνήθως και να γράψει δεδομένα στην κάρτα. Για οικονομία λόγου θα αναφέρουμε στο εξής ως «αναγνώστη» μία τέτοια συσκευή. Στην αγγλική βιβλιογραφία, η συσκευή αυτή αναφέρεται επίσης με τους όρους: read/write unit, contact unit, card acceptance device, interface device (IFD). Σε κάθε περίπτωση εννοείται το ίδιο πράγμα.

#### Αναγνώστης έξυπνων καρτών

Οι αναγνώστες έξυπνων καρτών είναι συνήθως μία ανεξάρτητη συσκευή, η οποία συνδέεται στη σειριακή, παράλληλη ή USB θύρα ενός υπολογιστή. Άλλοι τύποι αναγνωστών είναι ενσωματωμένοι στο πληκτρολόγιο ή ταιριάζουν σε μία σχισμή PCMCIA. Ένας άλλος τύπος αναγνώστη έχει το μέγεθος μίας δισκέτας 3.5 ιντσών και τοποθετείται στην αντίστοιχη υποδοχή του υπολογιστή.

Επιπρόσθετα με τη σχισμή για την κάρτα και τη διεπαφή σύνδεσης με τον υπολογιστή, ένας αναγνώστης μπορεί επίσης να έχει μία οθόνη και ένα πληκτρολόγιο για την εισαγωγή του PIN (PIN-pad). Η χρήση αυτού του πληκτρολογίου είναι για την εισαγωγή του Προσωπικού Αριθμού Αναγνώρισης

(Personal Identification Number, PIN), ο οποίος στέλνεται στην έξυπνη κάρτα για να αναγνωριστεί ο ιδιοκτήτης της κάρτας.

Για να λειτουργήσουν οι κάρτες που έχουν κρυπτογραφικό συνεπεξεργαστή, ο αναγνώστης χρειάζεται περισσότερο ηλεκτρικό ρεύμα από όσο μπορεί να αντλήσει από τη σειριακή θύρα του υπολογιστή. Για το λόγο αυτό, αρκετοί αναγνώστες έχουν μία εξωτερική τροφοδοσία ισχύος. Άλλοι αναγνώστες συνδέονται με την τροφοδοσία του υπολογιστή, χρησιμοποιώντας έναν ειδικό συνδετήρα, ο οποίος συνδέεται ανάμεσα στο πληκτρολόγιο και τον υπολογιστή. Οι αναγνώστες με διεπαφή PCMCIA μπορούν να αντλήσουν αρκετή ισχύ από τη διεπαφή PCMCIA.

Οι αναγνώστες έχουν διαφορετική υποστήριξη σε επίπεδο λογισμικού. Σχεδόν κάθε αναγνώστης έχει κάποιας μορφής υποστήριξη για το λειτουργικό σύστημα Microsoft Windows. Πολλοί από αυτούς συμμορφώνονται με την προδιαγραφή PC/SC [43], ενώ άλλοι έχουν μόνο ιδιόκτητη (proprietary υποστήριξη λογισμικού και δεν είναι συμβατοί με κάποια προδιαγραφή. Μία ιδιόκτητη διεπαφή λογισμικού μπορεί να μην είναι πρόβλημα αν η εφαρμογή θα πρέπει να δουλέψει με έναν μόνο τύπο αναγνώστη και ο προγραμματιστής της εφαρμογής δεν είναι εξοικειωμένος με κάποιο πρότυπο API. Σε όλες τις άλλες περιπτώσεις, μία ιδιόκτητη διεπαφή απαιτεί επιπλέον εργασία και εκμάθηση για τον προγραμματιστή.

Δεν έχει μόνο ο προγραμματιστής πλεονεκτήματα από μία εφαρμογή γραμμένη σύμφωνα με κάποιο πρότυπο API. Οι πελάτες έχουν το δικαίωμα της επιλογής ανάμεσα σε αναγνώστες, οι οποίοι υποστηρίζουν το συγκεκριμένο API. Μπορούν να αντικαταστήσουν τον αναγνώστη με κάποια άλλη συμβατή συσκευή, χωρίς να αλλάξουν ούτε μία γραμμή από τον κώδικα της εφαρμογής.

Για την ανάπτυξη εφαρμογών έξυπνων καρτών χρησιμοποιώντας τη γλώσσα και την πλατφόρμα εκτέλεσης Java, η φυσική επιλογή είναι το πρότυπο API OpenCard Framework. Το πρότυπο ανήκει στο OpenCard Consortium [42] και στην αγορά υπάρχουν ήδη πολλοί αναγνώστες που το υποστηρίζουν.

### Τερματικά έξυπνων καρτών

Το πλεονέκτημα ενός τερματικού έξυπνων καρτών έναντι ενός αναγνώστη έξυπνων καρτών συνδεδεμένου με έναν υπολογιστή είναι ο ισχυρότερος έλεγχος στην πρόσβαση της έξυπνης κάρτας. Ένα τερματικό μπορεί να σφραγισθεί για να αποτρέψει κακόβουλες επεμβάσεις στο υλικό του. Η εγκατάσταση του λογισμικού μπορεί να ελεγχθεί καλύτερα από ειδικά σχήματα, τα οποία δε θα μπορούσαν να εφαρμοσθούν σε έναν υπολογιστή γενικού σκοπού.

Ένα τέτοιο ασφαλές και προστατευμένο σύστημα είναι επιθυμητό σε συναλλαγές πληρωμών για παράδειγμα. Για μία συναλλαγή πληρωμής, ο πελάτης θέλει να είναι σίγουρος ότι καμία παραποίηση του υλικού ή του λογισμικού δε θα μπορούσε να αφαιρέσει περισσότερα χρήματα από την κάρτα από όσα έδωσε τη συγκατάθεσή του. Προτού χρησιμοποιηθεί ένα τερματικό για ένα συγκεκριμένο σχήμα ηλεκτρονικού πορτοφολιού, ο εκδότης συνήθως απαιτεί μία εξέταση του υλικού και του λογισμικού του τερματικού. Η πιστοποίηση αυτή εστιάζει κύρια στην ασφάλεια έναντι απάτης και κακομεταχείρισης.

Μερικά σχήματα πορτοφολιού απαιτούν μία ειδική έξυπνη κάρτα για τον έμπορο, η οποία εισάγεται σε μία από τις σχισμές του τερματικού. Σε αυτά τα σχήματα, τα χρήματα μεταφέρονται από την κάρτα του πελάτη στην κάρτα του εμπόρου ή η κάρτα του εμπόρου πιστοποιεί την αυθεντικότητα του τερματικού στην κάρτα του πελάτη.

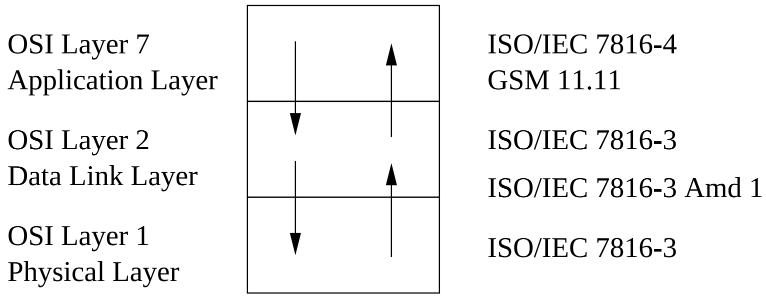
Συχνά, ένας έμπορος δέχεται πληρωμές από περισσότερα του ενός σχήματα πορτοφολιών. Η χρήση ενός ξεχωριστού τερματικού για κάθε σχήμα που δέχεται θα αύξανε το κόστος και θα καταλάμβανε περισσότερο χώρο. Για αυτό το λόγο, συχνά το τερματικό έχει μία σχισμή για την κάρτα του πελάτη και τέσσερις ή περισσότερες σχισμές για τις κάρτες του εμπόρου. Με ένα τέτοιο τερματικό, ο έμπορος μπορεί να δέχεται αρκετά διαφορετικά ηλεκτρονικά πορτοφόλια με μία μόνο συσκευή.

Ένα τερματικό είναι από μόνο του ένας υπολογιστής, με επεξεργαστή και μνήμη. Σήμερα, ένα τερματικό πληρωμών υψηλής ποιότητας έχει τυπικά έναν επεξεργαστή 32 bit και 1 Megabyte μνήμη. Μερικά τερματικά μπορούν επίσης να προγραμματιστούν σε Java. Για συντήρηση ή για την προσθήκη νέων χαρακτηριστικών στο τερματικό, είναι συχνά απαραίτητο να φορτώσουμε λογισμικό ή ανανεωμένες εκδόσεις λογισμικού σε αυτές τις συσκευές. Ένα τέτοιο φόρτωμα θα αλλάξει το σύστημα, μετά την πιστοποίηση. Για αυτό, θα πρέπει να υπάρχουν επιπλέον μηχανισμοί ασφάλειας σε συνδυασμό με την πιστοποίηση του νέου κώδικα που φορτώθηκε.

## 2.3 Πρωτόκολλα επικοινωνίας με έξυπνες κάρτες

Η δυνατότητα για επικοινωνία προς τις δύο πλευρές είναι προαπαιτούμενο για όλες τις αλληλεπιδράσεις μεταξύ μίας έξυπνης κάρτας και ενός τερματικού. Ωστόσο, όπως είδαμε στην ενότητα 2.1.7, είναι διαθέσιμη μόνο μία γραμμή επικοινωνίας. Τα ψηφιακά δεδομένα ανταλλάσσονται μεταξύ της κάρτας και του τερματικού μέσω αυτής της ηλεκτρικής σύνδεσης. Αφού μόνο μία γραμμή υπάρχει, η κάρτα και το τερματικό θα πρέπει να παίρνουν σειρά στην αποστολή των δεδομένων· την ίδια ώρα η άλλη πλευρά θα πρέπει να λειτουργεί ως παραλήπτης. Η εναλλασσόμενη αποστολή και λήψη δεδομένων λέγεται μετάδοση μονής κατεύθυνσης (**half-duplex**). Μία μετάδοση διπλής κατεύθυνσης δεν υλοποιείται προς το παρόν στις έξυπνες κάρτες. Ωστόσο, καθώς οι περισσότεροι επεξεργαστές για έξυπνες κάρτες έχουν δύο θύρες εισόδου/εξόδου και δύο από τις οκτώ επαφές είναι δεσμευμένες για μελλοντική χρήση, η μετάδοση διπλής κατεύθυνσης θα είναι σίγουρα τεχνικώς δυνατή.

Η επικοινωνία με την κάρτα ξεκινά πάντα από το τερματικό. Η κάρτα απαντά πάντα σε εντολές του τερματικού, το οποίο σημαίνει ότι η κάρτα ποτέ δε στέλνει δεδομένα χωρίς εξωτερική αίτηση. Αυτό οδηγεί σε μία σχέση αφέντη-σκλάβου, όπου το τερματικό είναι ο αφέντης και η κάρτα ο σκλάβος.




Σχήμα 2.5: Επικοινωνία έξυπνης κάρτας και τερματικού κατά το μοντέλο αναφοράς OSI

Αφού η κάρτα εισαχθεί σε ένα τερματικό, οι επαφές της καταρχήν συνδέονται μηχανικά με αυτές του τερματικού. Οι πέντε ενεργές επαφές στη συνέχεια ενεργοποιούνται ηλεκτρικά με τον τρόπο που περιγράφεται στο πρότυπο ISO 7816-3. Στη συνέχεια η κάρτα αυτόματα εκτελεί τη διαδικασία εκκίνησης και στέλνει ένα ATR, Answer To Reset στο τερματικό. Το τερματικό εξετάζει το ATR, στο οποίο αναφέρονται διάφορες λεπτομέρειες για την κάρτα και τη μετάδοση δεδομένων, και στη συνέχεια στέλνει την πρώτη εντολή. Η κάρτα επεξεργάζεται την εντολή και παράγει μία απάντηση, την οποία στέλνει πίσω στο τερματικό. Η διαδικασία αυτή συνεχίζεται έως ότου η κάρτα απενεργοποιηθεί.

Ανάμεσα στο ATR και στην πρώτη εντολή, το τερματικό μπορεί να στείλει μία εντολή Επιλογής Τύπου Πρωτοκόλλου ( PTS, Protocol Type Selection). Το τερματικό μπορεί να χρησιμοποιήσει αυτήν την εντολή, η οποία όπως και η ATR, είναι ανεξάρτητη του πρωτοκόλλου μετάδοσης, για να θέσει διάφορες παραμέτρους επικοινωνίας, όπως για παράδειγμα την ταχύτητα επικοινωνίας.

Η συνολική διαδικασία μετάδοσης δεδομένων προς και από την κάρτα μπορεί να αναπαρασταθεί χρησιμοποιώντας το μοντέλο επιπέδων OSI. Αυτό διαχωρίζει σε επίπεδα τα ηλεκτρικά σήματα στην επαφή `I/O`, λογικές διαδικασίες στο πραγματικό πρωτόκολλο μετάδοσης και στη συμπεριφορά των εφαρμογών που χρησιμοποιούν αυτές τις διαδικασίες. Η συμπεριφορά και αλληλεπίδραση μέσα σε αυτά αλλά και ανάμεσα στα διάφορα επίπεδα καθορίζονται σε διάφορα διεθνή πρότυπα, όπως για παράδειγμα τα ISO 7816, που έχουμε αναφέρει και σε προηγούμενες ενότητες. Οι σχέσεις αυτές σημειώνονται στο σχήμα 2.5.

Υπάρχουν διάφοροι τρόποι με τους οποίους μπορεί να υλοποιηθεί η επικοινωνία με την κάρτα. Υπάρχουν επίσης αρκετές διαφορετικές διαδικασίες για τον επανασυγχρονισμό της επικοινωνίας, αν υπάρξει κάποια διαταραχή. Η ακριβής υλοποίηση των εντολών, οι αντίστοιχες απαντήσεις και οι διαδικασίες που χρησιμοποιούνται σε περίπτωση που συμβεί κάποιο λάθος καθορίζονται υπό τη μορφή των «πρωτοκόλλων μετάδοσης». Υπάρχουν συνολικά 15 προτυποποιημένα πρωτόκολλα μετάδοσης. Τα πρωτόκολλα αυτά σημειώνονται με

T= (από το αγγλικό: Transmission protocol) και ένα σειριακό αριθμό, όπως συνοψίζεται στον πίνακα 2.1. Δύο από αυτά τα πρωτόκολλα κυριαρχούν σε

| Πρωτόκολλο μετάδοσης | Χαρακτηριστικά |
|---|---|
| T=0 | Ασύγχρονο, μονής κατεύθυνσης, προσαρμοσμένο σε byte, ISO 7816-3 |
| T=1 | Ασύγχρονο, μονής κατεύθυνσης, προσαρμοσμένο σε πακέτο, ISO 7816-3 Ammendum 1 |
| T=2 | Ασύγχρονο, διπλής κατεύθυνσης, προσαρμοσμένο σε πακέτο, ISO 10536-4 |
| T=3 | Διπλής κατεύθυνσης, δεν έχει οριστεί ακόμη |
| T=4 | Ασύγχρονο, μονής κατεύθυνσης, προσαρμοσμένο σε byte, επέκταση του T=0, δεν έχει οριστεί ακόμη |
| T=5–T=13 | Δεσμευμένα για μελλοντικούς ορισμούς |
| T=14 | Για εθνικές λειτουργίες, δεν περιλαμβάνεται στο πρότυπο ISO |
| T=15 | Δεσμευμένα για μελλοντικούς ορισμούς |

Πίνακας 2.1: Προτυποποιημένα πρωτόκολλα μετάδοσης κατά ISO

χρήση παγκοσμίως, το T=0 και το T=1. Το T=2 βρίσκεται υπό μελέτη και θα είναι διαθέσιμο σε λίγα χρόνια ως διεθνές πρότυπο. Στη Γερμανία, ένα τρίτο πρωτόκολλο, το T=14 χρησιμοποιείται για τα ευρέως διαθέσιμα καρτοτηλέφωνα. Επιπρόσθετα των τεχνικά πολύπλοκων πρωτοκόλλων μετάδοσης για έξυπνες κάρτες, υπάρχει ένα επιπλέον σύνολο πολύ απλών σύγχρονων πρωτοκόλλων για κάρτες μνήμης. Συνήθως χρησιμοποιούνται στις τηλεκάρτες, στις κάρτες ασφάλειας υγείας και σε παρόμοιες εφαρμογές. Ωστόσο δεν έχουν μηχανισμούς διόρθωσης λαθών και βασίζονται σε λογική ενσωματωμένη στο ολοκληρωμένο κύκλωμα (hard-wired logic).

### 2.3.1 Σύγχρονη μετάδοση δεδομένων

Η σύγχρονη μετάδοση δεδομένων δε χρησιμοποιείται σε έξυπνες κάρτες με μικροεπεξεργαστή. Ωστόσο, είναι η συνήθης διαδικασία για τις κάρτες μνήμης. Οι κάρτες μνήμης χρησιμοποιούνται σε μεγάλες ποσότητες ως προπληρωμένο ηλεκτρονικό πορτοφόλι, για παράδειγμα στα καρτοτηλέφωνα.

Στις κάρτες μνήμης η σύγχρονη μετάδοση δεδομένων είναι στενά συνδεδεμένη με το υλικό του ολοκληρωμένου κυκλώματος και είναι σχεδιασμένη ώστε να είναι όσο το δυνατό απλούστερη. Δεν υπάρχει διαχωρισμός επιπέδων στο πρωτόκολλο μετάδοσης, ούτε λογική διευθυνσιοδότηση. Έτσι η εφαρμογή στο τερματικό πρέπει να προσπελάσει απευθείας τις διευθύνσεις μνήμης του ολοκληρωμένου. Το πρωτόκολλο επιτρέπει να προσπελαστούν τα δεδομένα της μνήμης με φυσικές διευθύνσεις καθώς και να διαβαστούν ή να γραφούν. Αυτό σημαίνει ότι η πραγματική διαδικασία μεταφοράς δεδομένων συμπεριλαμβάνει επίσης τις λειτουργίες διευθυνσιοδότησης και διαχείρισης της μνήμης.

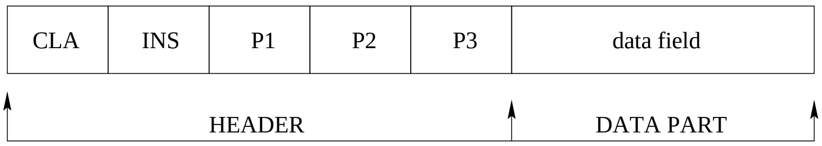
Δεν υπάρχει επίσης διαδικασία για την ανίχνευση και διόρθωση λαθών κατά τη διάρκεια της μετάδοσης· θα πρέπει ωστόσο να σημειώσουμε ότι τέτοια λάθη συμβαίνουν αρκετά σπάνια. Αν πάραυτα η εφαρμογή του τερματικού ανιχνεύσει κάποιο λάθος μετάδοσης, θα πρέπει να ξαναδιαβάσει τη σχετική περιοχή της μνήμης της κάρτας. Όλοι αυτοί οι περιορισμοί υφίστανται ώστε να επιτρέπεται η μετάδοση δεδομένων μεταξύ της κάρτας και του τερματικού με υψηλούς ρυθμούς και χρησιμοποιώντας μόνο ένα μικρό μέρος λογικής σε υλικό (logic hardware).

Αφού η σύγχρονη μετάδοση δεδομένων χρησιμοποιείται μόνο για να είναι όσο το δυνατό πιο απλή η μετάδοση δεδομένων, η, σχεδόν αναπόφευκτη, συνέπεια είναι η ισχυρή εξάρτηση από το υλικό. Αυτό σημαίνει ότι τα σύγχρονα πρωτόκολλα μετάδοσης δεν είναι ομοιόμορφα και μερικές φορές διαφέρουν σημαντικά από ολοκληρωμένο σε ολοκληρωμένο. Μόνο το ATR είναι προτυποποιημένο. Ένα τερματικό το οποίο θα πρέπει να επικοινωνεί με διάφορους τύπους καρτών μνήμης θα πρέπει να συμπεριλαμβάνει διάφορες υλοποιήσεις σύγχρονων πρωτοκόλλων μετάδοσης.

Ο ακριβής ορισμός της μετάδοσης δεδομένων που χρησιμοποιείται στις κάρτες μνήμης είναι «σειριακή μετάδοση δεδομένων σύγχρονα με το ρολόι». Αυτό ξεκάθαρα φανερώνει τις βασικές συνθήκες που εφαρμόζονται σε αυτού του τύπου την επικοινωνία. Όπως και στις ασύγχρονες επικοινωνίες, τα δεδομένα στέλνονται ανάμεσα στην κάρτα και το τερματικό σειριακά, δηλαδή bit προς bit. Ωστόσο, τα bit στέλνονται σύγχρονα με ένα βοηθητικό σήμα παλμού ρολογιού. Με τον τρόπο αυτό η μεταφορά πληροφορίας αρχής και τέλους μετάδοσης είναι περιττή.

Στην περίπτωση μίας απλής κάρτας μνήμης, δεν υπάρχει επίσης πληροφορία ανίχνευσης λάθους, το οποίο σημαίνει ότι δε στέλνεται bit ισοτιμίας ή κάποιο άθροισμα ισοτιμίας (checksum). Η μικρή πιθανότητα σφαλμάτων στη μετάδοση οφείλεται στο πολύ μικρό ρυθμό ρολογιού, ο οποίος είναι μεταξύ 10 kHz και 100 kHz. Αφού ένα bit στέλνεται σε κάθε κύκλο ρολογιού, ένας ρυθμός ρολογιού 20 kHz ορίζει ένα ρυθμό μετάδοσης 20 kbit/s. Ο ουσιαστικός ρυθμός είναι ωστόσο μικρότερος για τις κάρτες μνήμης, καθώς θα πρέπει να στέλνεται επιπλέον πληροφορία διευθυνσιοδότησης.

Η σχέση αφέντη-σκλάβου είναι ισχυρότερη στις κάρτες μνήμης από ότι στις έξυπνες κάρτες. Για παράδειγμα, το τερματικό ελέγχει πλήρως τη φυσική διευθυνσιοδότηση της μνήμης. Η κάρτα μπορεί μόνο να εμποδίσει συγκεκριμένες, ολόκληρες περιοχές από το να διαγραφούν. Αυτό ελέγχεται από λογική hard-wired, η οποία βρίσκεται «μπροστά» από τη μνήμη. Η λογική αυτή διαχειρίζεται επίσης την πολύ απλή διαδικασία μετάδοσης δεδομένων.




Σχήμα 2.6: Δομή ενός μηνύματος του πρωτοκόλλου T=0

### 2.3.2 Ασύγχρονη μετάδοση δεδομένων - Το πρωτόκολλο T=0

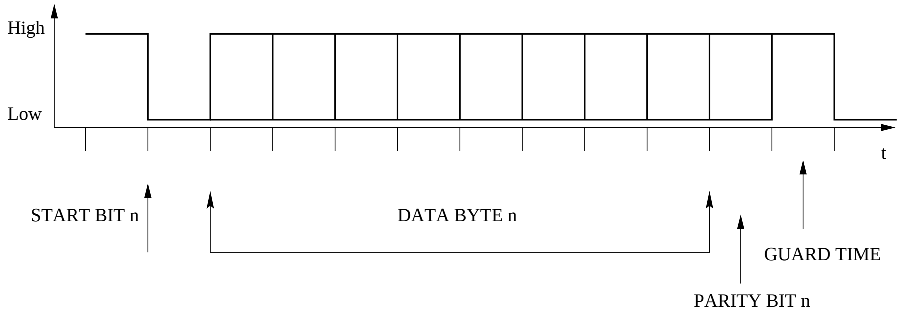
Το πρωτόκολλο αυτό χρησιμοποιήθηκε για πρώτη φορά στη Γαλλία, κατά τη διάρκεια της ανάπτυξης των έξυπνων καρτών (πβ. ενότητα 2.1.2) και ήταν επίσης το πρώτο πρωτόκολλο που προτυποποιήθηκε διεθνώς. Δημιουργήθηκε στα πρώτα χρόνια της τεχνολογίας των έξυπνων καρτών και έτσι είναι σχεδιασμένο για ελάχιστη χρήση μνήμης και μέγιστη απλότητα. Το πρωτόκολλο αυτό χρησιμοποιείται παγκοσμίως στο σύστημα GSM και κατά συνέπεια είναι το πλέον δημοφιλές ανάμεσα ανάμεσα σε όλα τα πρωτόκολλα έξυπνων καρτών. Το πρωτόκολλο T=0 προτυποποιείται στο ISO 7816-3. Αντίστοιχες συμβατές προδιαγραφές περιέχονται στα πρότυπα GSM 11.11 και EMV.

Το πρωτόκολλο T=0 είναι προσαρμοσμένο σε byte, το οποίο σημαίνει ότι η μικρότερη μονάδα που επεξεργάζεται από το πρωτόκολλο είναι ένα byte. Μία ομάδα δεδομένων μετάδοσης περιέχει μία επικεφαλίδα, η οποία συνίσταται από ένα class byte, ένα byte εντολής και τρία byte παραμέτρων. Η επικεφαλίδα μπορεί να ακολουθείται από ένα προαιρετικό τμήμα δεδομένων. Το μήκος των δεδομένων αναφέρεται στην παράμετρο $P3$. Αυτή σημειώνει είτε το μήκος των δεδομένων της εντολής είτε το αναμενόμενο μήκος της απαντήσεως (σχήμα 2.6).

Λόγω της προσαρμογής του πρωτοκόλλου σε byte, αν ανιχνευτεί ένα σφάλμα μετάδοσης, θα πρέπει άμεσα να ζητηθεί η επαναμετάδοση του εσφαλμένου byte. Σε αντίθεση, τα πρωτόκολλα που είναι προσαρμοσμένα σε πακέτα, απαιτούν την επαναμετάδοση όλου του πακέτου αν συμβεί ένα λάθος. Η ανίχνευση σφάλματος στο πρωτόκολλο T=0 γίνεται αποκλειστικά με την προσθήκη ενός bit ισοτιμίας σε κάθε byte. Στο σχήμα 2.7 φαίνεται η μετάδοση ενός byte δεδομένων χρησιμοποιώντας το πρωτόκολλο T=0. Ο χρόνος guard χρησιμοποιείται ώστε να ξεχωρίζουν μεταξύ τους τα διάφορα byte κατά τη μετάδοση. Αυτό παρέχει τόσο στον αποστολέα, όσο και στον παραλήπτη, επιπλέον χρόνο ώστε να εκτελέσουν τις λειτουργίες του πρωτοκόλλου.

### 2.3.3 Ασύγχρονη μετάδοση δεδομένων - Το πρωτόκολλο T=1

Το πρωτόκολλο μετάδοσης T=1 είναι ένα ασύγχρονο πρωτόκολλο μετάδοσης μονής κατεύθυνσης για έξυπνες κάρτες. Βασίζεται στο διεθνές πρότυπο ISO 7816-3. Το πρότυπο EMV είναι επίσης σχετικό με το πρωτόκολλο. Το πρωτόκολλο T=1 είναι προσαρμοσμένο σε πακέτα. Αυτό σημαίνει ότι ένα




Σχήμα 2.7: Μεταφορά ενός byte με το πρωτόκολλο T=0

πακέτο είναι η μικρότερη μονάδα δεδομένων η οποία μπορεί να μεταδοθεί ανάμεσα στην κάρτα και το τερματικό.

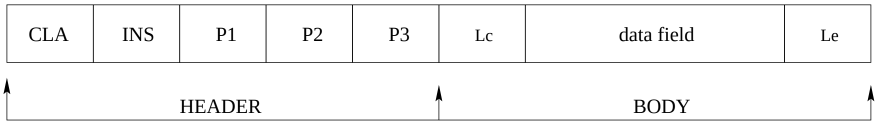
Το πρωτόκολλο αυτό χαρακτηρίζεται από την αυστηρή διάκριση επιπέδων και μπορεί να ταξινομηθεί στο μοντέλο αναφοράς OSI ως πρωτόκολλο επιπέδου σύνδεσης δεδομένων (data link layer). Σε αυτό το πλαίσιο, η διάκριση των επιπέδων σημαίνει ότι τα δεδομένα που απευθύνονται σε υψηλότερα επίπεδα, όπως για παράδειγμα το επίπεδο εφαρμογής, μπορούν να επεξεργαστούν διάφανα από το επίπεδο σύνδεσης δεδομένων. Δεν είναι απαραίτητο για τα επίπεδα εκτός από αυτό για το οποίο αναφέρονται τα δεδομένα να διερμηνεύσουν ή να αλλάξουν τα περιεχόμενα των μεταφερόμενων δεδομένων.

Η ασφαλής ανταλλαγή μηνυμάτων ειδικότερα, απαιτεί αυστηρή προσκόλληση στη διάκριση των επιπέδων. Μόνο τότε μπορούν τα κρυπτογραφημένα δεδομένα του χρήστη να περάσουν ανάμεσα στις διεπαφές των επιπέδων χωρίς να απαιτούνται περίπλοκες μέθοδοι και τεχνάσματα. Το πρωτόκολλο T=1 είναι το μόνο διεθνές πρωτόκολλο για έξυπνες κάρτες αυτή τη στιγμή, το οποίο επιτρέπει την ασφαλή μετάδοση δεδομένων σε όλες τις παραλλαγές, χωρίς κάποιο πρόβλημα ή συμβιβασμούς. Παρενθετικά αναφέρουμε ότι το πρωτόκολλο T=1 δεν περιορίζεται μόνο για επικοινωνίες έξυπνων καρτών με τερματικά. Χρησιμοποιείται από πολλά τερματικά για να ανταλλάσσουν χρήσιμα δεδομένα και δεδομένα ελέγχου με τους υπολογιστές, με τους οποίους είναι συνδεδεμένα.

### 2.3.4 Δομή μηνυμάτων: APDU

Η ανταλλαγή δεδομένων μεταξύ του on-card τμήματος μίας εφαρμογής έξυπνης κάρτας και του off-card τμήματος, γίνεται με τη χρήση των APDU. Ο όρος APDU είναι το ακρωνύμιο για το «Application Protocol Data Unit». Αναπαριστά μία διεθνώς προτυποποιημένη μονάδα δεδομένων στο επίπεδο εφαρμογής, το οποίο είναι το επίπεδο 7 στο μοντέλο αναφοράς OSI. Το επίπεδο αυτό είναι ακριβώς πάνω από το επίπεδο μετάδοσης στις έξυπνες κάρτες (πβ. σχήμα 2.5).

Γίνεται ένας διαχωρισμός ανάμεσα σε εντολές APDU, οι οποίες αναπα-




Σχήμα 2.8: Η δομή μίας εντολής APDU

ριστούν εντολές προς την κάρτα, και απαντήσεις APDU, οι οποίες είναι οι απαντήσεις της κάρτας σε αυτές τις εντολές. Απλούστερα, ένα APDU είναι κατά κάποιο τρόπο ένα κουτί που περιέχει είτε μία πλήρη εντολή προς την κάρτα ή μία πλήρη απάντηση από την κάρτα. Τα APDU μεταφέρονται από το πρωτόκολλο μετάδοσης διάφανα, δηλαδή χωρίς να αλλαχθούν ή διερμηνευθούν.

Τα APDU που είναι συμβατά με το πρότυπο ISO 7816-4 είναι σχεδιασμένα να είναι ανεξάρτητα από το πρωτόκολλο μετάδοσης. Το περιέχομενο και η μορφή ενός APDU λοιπόν πρέπει να μην αλλάζει όταν χρησιμοποιείται ένα διαφορετικό πρωτόκολλο μετάδοσης. Αυτό ισχύει κατά κύριο λόγο για τα δύο προτυποποιημένα πρωτόκολλα που εξετάσαμε παραπάνω, τα T=0 και T=1. Η απαίτηση για ανεξαρτησία από το πρωτόκολλο επηρέασε το σχεδιασμό των APDU, αφού θα πρέπει να είναι δυνατό να μεταδοθούν διάφανα χρησιμοποιώντας τόσο το προσαρμοσμένο σε byte πρωτόκολλο T=0, όσο και το προσαρμοσμένο σε πακέτο πρωτόκολλο T=1.

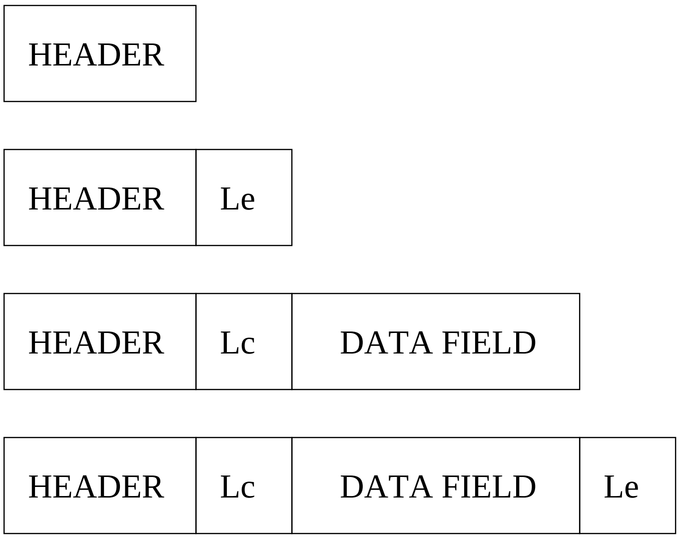
### Η δομή μίας εντολής APDU

Μία εντολή APDU αποτελείται από μία επικεφαλίδα και ένα σώμα, όπως φαίνεται στο σχήμα 2.8 Το σώμα μπορεί να είναι μεταβλητού μήκους ή και να λείπει τελείως, αν το αντίστοιχο πεδίο δεδομένων είναι κενό.

Η επικεφαλίδα αποτελείται από τέσσερα στοιχεία, τα οποία είναι το class byte, CLA, το instruction byte, INS και δύο byte παραμέτρων, P1 και P2. Το CLA χρησιμοποιείται ακόμη και σήμερα για να αναγνωρίζεται οι εφαρμογές και το σύνολο εντολών τους. Για παράδειγμα, το GSM χρησιμοποιεί το CLA=A0, ενώ το CLA=8X χρησιμοποιείται κύρια από εταιρείες για ιδιωτική χρήση. Το CLA=0X είναι δεσμευμένο για εντολές συμβατές με το πρότυπο ISO.

Το INS χρησιμοποιείται για να κωδικοποιηθεί η πραγματική εντολή προς την κάρτα. Σχεδόν όλοι οι συνδυασμοί είναι επιτρεπτοί για την τιμή αυτού του byte, με τον περιορισμό ότι χρησιμοποιούνται μόνο άρτιοι. Ο περιορισμός αυτός προκύπτει από μία λεπτομέρεια του πρωτοκόλλου μετάδοσης T=0.

Τα δύο byte παραμέτρων χρησιμοποιούνται κυρίως για να παρέχουν περισσότερες πληροφορίες για τη συγκεκριμένη εντολή που επιλέχθηκε με το INS. Για παράδειγμα, χρησιμοποιούνται για να επιλέξουμε τις διάφορες επιλογές για την εντολή `SELECT FILE` ή για να ορισθεί το offset για την εντολή `READ BINARY`.




Σχήμα 2.9: Οι τέσσερις δυνατές περιπτώσεις εντολής APDU

Το σώμα, το οποίο μπορεί να παραληφθεί με μία εξαίρεση, είναι το δεύτερο τμήμα μίας εντολής APDU. Το σώμα εξυπηρετεί διπλό ρόλο. Πρώτα, ορίζει το μέγεθος των δεδομένων που στέλνονται στην κάρτα (το πεδίο Lc, Length command) και το μέγεθος των δεδομένων που πρέπει να στείλει η κάρτα ως απάντηση (το πεδίο Le, Length executed). Δεύτερον, περιέχει τα δεδομένα που συνδέονται με τη συγκεκριμένη εντολή που στέλνεται στην κάρτα. Αν η τιμή του πεδίου Le είναι μηδέν, το τερματικό αναμένει ότι η κάρτα θα στείλει το μέγιστο αριθμό δεδομένων που είναι διαθέσιμα για τη συγκεκριμένη εντολή. Αυτή είναι η μόνη εξαίρεση στην αριθμητική ερμηνεία του πεδίου μήκους.

Τα διάφορα μέρη μίας εντολής APDU μπορούν να συνδυαστούν για να παράγουν τέσσερις δυνατούς τύπους εντολής APDU που προβλέπει το πρότυπο ISO 7816-4, όπως φαίνεται στο σχήμα 2.9.

### Η δομή μίας απάντησης APDU

Η απάντηση APDU, η οποία στέλνεται από την κάρτα ως απάντηση σε μία εντολή APDU, αποτελείται από ένα προαιρετικό σώμα και ένα υποχρεωτικό trailer, όπως φαίνεται στο σχήμα 2.10. Το σώμα αποτελείται από το πεδίο δεδομένων, το μέγεθος του οποίου καθορίζεται από το πεδίο Le της αντίστοιχης εντολής APDU. Το μέγεθος του πεδίου δεδομένων μπορεί να είναι μηδέν, ανεξάρτητα από την τιμή που έχει οριστεί στην εντολή APDU, αν η έξυπνη κάρτα τελείωσε την επεξεργασία της εντολής λόγω κάποιου σφάλματος ή λάθος παραμέτρων. Αυτό σημειώνεται με τις δύο λέξεις κατάστασης, SW1 και SW2 του trailer.

Η κάρτα πρέπει να στείλει πάντα ένα trailer ως απάντηση σε μία εντολή. Τα

| DATA FIELD | SW1 | SW2 |
|---|---|---|

Σχήμα 2.10: Η δομή της απάντησης APDU

δύο byte SW1, SW2, τα οποία επίσης καλούνται return code, κωδικοποιούν την απάντηση της εντολής. Για παράδειγμα, ο κωδικός `0x9000` σημαίνει ότι η εντολή εκτελέστηκε εξ ολοκλήρου και επιτυχώς. Θα πρέπει να σημειωθεί ότι παρά το γεγονός ότι υπάρχει το πρότυπο για τους κωδικούς αυτούς, πολλές εφαρμογές χρησιμοποιούν ασύμβατους κωδικούς. Η μόνη εξαίρεση είναι ο κωδικός `0x9000`, ο οποίος σχεδόν καθολικά σημαίνει επιτυχή εκτέλεση. Σε κάθε άλλη περίπτωση είναι απαραίτητο να συμβουλευθεί κανείς τις προδιαγραφές της κάρτας για την ακριβή ερμηνεία του κωδικού.

## 2.4 Συστήματα με έξυπνες κάρτες

Κλείνοντας την αναφορά μας στην τεχνολογία των έξυπνων καρτών, θα παρουσιάσουμε ορισμένες χαρακτηριστικά συστήματα και εμπορικά προϊόντα όπου χρησιμοποιούνται οι έξυπνες κάρτες. Ο κατάλογος σε καμία περίπτωση δεν είναι πλήρης· σκοπός μας είναι να καταδείξουμε το εύρος και την ποικιλία των συστημάτων όπου μπορούν να χρησιμοποιηθούν οι έξυπνες κάρτες.

### 2.4.1 Ηλεκτρονικά πορτοφόλια

Τα πιο υποσχόμενα συστήματα για ηλεκτρονικό χρήμα βασίζονται σε έξυπνες κάρτες. Τα περισσότερα συστήματα πληρωμών είτε χρησιμοποιούν ήδη έξυπνες κάρτες είτε έχουν ανακοινώσει σχέδια για αυτές στο άμεσο μέλλον. Για λόγους ασφαλείας, οι πληρωμές με πιστωτικές κάρτες απαιτούν σήμερα συναλλαγές on-line. Η διατήρηση μίας μόνιμης σύνδεσης on-line ή η δημιουργία μίας για κάθε συναλλαγή εμπεριέχει κόστη, τα οποία μπορεί να είναι πολύ ακριβά για την πληρωμή για μικρά ποσά (συχνά αναφερόμενα ως «μικροπληρωμές» ). Για την πληρωμή μικρών ποσών, τα συστήματα πορτοφολιού βασισμένα σε έξυπνες κάρτες παρέχουν μία ασφαλή, αξιόπιστη και φθηνή λύση. Οι κάρτες ηλεκτρονικού πορτοφολιού είναι επίσης μία ελκυστική εναλλακτική ιδέα για μετρητά σε περιοχές όπου η διαχείριση μετρητών (κύρια κερμάτων) είναι μία ακριβή εργασία, όπως για παράδειγμα στους αυτόματους πωλητές.

Ως ένα ηλεκτρονικό πορτοφόλι, η έξυπνη κάρτα αποθηκεύει το πραγματικό υπόλοιπο. Για να αποτρέψει χειρισμό του υπολοίπου χωρίς εξουσιοδότηση, η έξυπνη κάρτα απαιτεί ταυτοποίηση των εντολών. Η αύξηση και μείωση του υπολοίπου ασφαλίζεται με τη χρήση δύο διαφορετικών κρυπτογραφικών κλειδιών. Για να φορτωθούν χρήματα στην κάρτα απαιτείται ένα τρίτο κλειδί, το οποίο συνήθως ανήκει στην τράπεζα.

Για να αφαιρεθούν χρήματα από την κάρτα και να μεταφερθούν στο λογαριασμό του εμπόρου, και πάλι ένα κλειδί απαιτείται. Το τερματικό του εμπόρου έχει αυτό το κλειδί, το οποίο βρίσκεται αποθηκευμένο σε ένα ασφαλές μέρος, το οποίο λέγεται Άρθρωμα Ασφαλούς Πρόσβασης (SAM, Security Access Module). Αυτό το άρθρωμα είναι συνήθως μία δεύτερη έξυπνη κάρτα, η οποία βρίσκεται μονίμως στο τερματικό. Αυτοί οι μηχανισμοί ασφαλείας αποτρέπουν την αύξηση ή τη μείωση του υπολοίπου μίας κάρτας-ηλεκτρονικό πορτοφόλι από μη εξουσιοδοτημένα άτομα.

Αυτή τη στιγμή υπάρχουν πάνω από 40 διαφορετικά ηλεκτρονικά πορτοφόλια που χρησιμοποιούνται σε δοκιμαστικές εφαρμογές. Μερικά από τα πλέον γνωστά παραθέτονται στον πίνακα 2.2. Οι κάρτες είναι έγκυρες για ένα σύστημα μόνο και δε μπορούν να χρησιμοποιηθούν για πληρωμές σε ένα άλλο σύστημα. Αυτό δε βοηθά την ελκυστικότητα των ηλεκτρονικών πορτοφολιών, αν οι κάρτες ενός συστήματος είναι αποδεκτές μόνο σε μία χώρα. Ειδικά με την εισαγωγή του ευρώ σε πολλές ευρωπαϊκές χώρες, οι καταναλωτές πολύ δύσκολα θα προτιμούν μία συλλογή από ασύμβατα ηλεκτρονικά πορτοφόλια για να πληρώνουν με ένα ενιαίο νόμισμα.

Μία νέα πρωτοβουλία, το Common Electronic Purse Specification (CEPS), έχει ξεκινήσει για να παρουσιάσει μία κοινή αρχιτεκτονική για ένα καθολικό σύστημα από συμβατά αλλά ανεξάρτητα συστήματα ηλεκτρονικών πορτοφολιών. Αυτή τη στιγμή συμμετέχουν οργανισμοί από 22 χώρες, οι οποίοι αντιπροσωπεύουν πάνω από το 90% των συστημάτων ηλεκτρονικών πορτοφολιών παγκοσμίως.

| Όνομα συστήματος | Χώρα | Κατασκευαστής συστήματος | Κάρτες (εκατομμύρια) |
|---|---|---|---:|
| Cash | Ελβετία | Proton | 2.9 |
| Chipper | Ολλανδία | KPN | 10.0 |
| Chipknip | Ολλανδία | Ολλανδικές τράπεζες | 9.0 |
| Danmont | Δανία | PBS Danmont | 0.9 |
| GeldKarte | Γερμανία | ZKA | 50.0 |
| Mondex | 49 χώρες | Mondex | 1.0 |
| Proton | 15 χώρες | Banksys/Proton World | 30.0 |
| Quick | Αυστρία | Austria Card | 5.0 |
| Visa Cash | 19 χώρες | Visa | 8.0 |

Πίνακας 2.2: Συστήματα ηλεκτρονικών πορτοφολιών

### 2.4.2 Ταυτοποίηση και ασφαλής πρόσβαση

Ένα επαναλαμβανόμενο θέμα στην εποχή των νέων τεχνολογιών είναι η πρόκληση για την ορθή αναγνώριση ενός προσώπου σε έναν υπολογιστή, σε μία σύνοδο επικοινωνίας ή στο ρόλο του αποστολέα ενός μηνύματος. Η έξυπνη κάρτα έχει ένα σημαντικό ρόλο ως μία ασφαλή συσκευή, η οποία παρέχει δυνατότητες κρυπτογραφίας.

Η πιο κοινή εναλλακτική λύση σε σχήματα που βασίζονται σε έξυπνες κάρτες είναι η χρήση συνθηματικών λέξεων ή φράσεων. Τα συνθηματικά όμως μπορούν να υποκλαπούν ή να μοιραστούν ελεύθερα. Για το λόγο αυτό, πολλές εταιρείες εξετάζουν σχήματα που βασίζονται σε έξυπνες κάρτες ή βιομετρικές μεθόδους. Στη συνέχεια κάνουμε μία μικρή περίληψη της αναγνώρισης με χρήση έξυπνων καρτών.

Συνήθως η έξυπνη κάρτα απαιτεί η ίδια την ταυτοποίηση του κατόχου της με ένα συνθηματικό. Αυτό αναφέρεται μερικές φορές ως ταυτοποίηση «δύο παραγόντων», γιατί συνδυάζει «κάτι που έχεις» –την έξυπνη κάρτα– με «κάτι που ξέρεις» –το συνθηματικό που χρειάζεται για να χρησιμοποιηθεί η κάρτα. Η ταυτοποίηση δύο παραγόντων μπορεί να προσφέρει ένα υψηλότερο επίπεδο ασφάλειας από ότι η ταυτοποίηση με συνθηματικό ή με μία συσκευή από μόνη της.

### Πρόσβαση σε σταθμούς εργασίας

Ο πιο συνηθισμένος τρόπος για την ασφάλιση της πρόσβασης σε ένα σταθμό εργασίας είναι η ταυτοποίηση του χρήστη, προτού εκκινήσει μία συνεδρία χρήστη (user session). Μερικά συστήματα δεν εκκινούν ούτε το λειτουργικό σύστημα, αν δεν προηγηθεί ταυτοποίηση. Και οι δύο παραλλαγές μπορούν να συνδυαστούν με μία επιπλέον λειτουργία, η οποία κλειδώνει τη συνεδρία μετά από ένα ορισμένο χρόνο χωρίς δραστηριότητα και στη συνέχεια απαιτεί ταυτοποίηση για να συνεχιστεί η συνεδρία.

Για τα δημοφιλή λειτουργικά συστήματα, όπως διάφορες παραλλαγές του UNIX και το Microsoft Windows NT, υπάρχουν πολλά προϊόντα στην αγορά που παρέχουν ασφαλή πρόσβαση στο σταθμό εργασίας βασισμένη σε έξυπνες κάρτες. Στα περισσότερα από αυτά τα προϊόντα, η ταυτοποίηση με έξυπνη κάρτα χρησιμοποιεί κλασσικούς κρυπτογραφικούς μηχανισμούς. Οι κρυπτογραφικοί υπολογισμοί γίνονται από την έξυπνη κάρτα, η οποία διατηρεί εσωτερικά τις μυστικές πληροφορίες που απαιτούνται για την ταυτοποίηση. Οι κρυπτογραφικοί μηχανισμοί είναι το αντικείμενο της επόμενης ενότητας.

### Login σε δίκτυα και εξυπηρέτες

Η ταυτοποίηση απαιτείται για να εργαστεί κανείς σε ένα απομακρυσμένο εξυπηρέτη, για να προσπελάσει τα δεδομένα ενός εξυπηρέτη ή για να χρησιμοποιήσει ένα ιδιωτικό δίκτυο. Είναι προφανές ότι όσο πιο ασφαλείς είναι οι μηχανισμοί ταυτοποίησης, τόσο πιο προστατευμένα θα είναι τα δεδομένα, οι εξυπηρέτες και τα ιδιωτικά δίκτυα. Σε όλες αυτές τις περιπτώσεις, η ταυτοποίηση συνήθως βασίζεται σε ένα πρωτόκολλο πρόκλησης-απάντησης (challenge-response protocol) με ασύμμετρα κλειδιά.

Η ταυτοποίηση μπορεί να γίνει προς δύο πλευρές. Είτε ο εξυπηρέτης πρέπει να αποδείξει την ταυτότητα του στον πελάτη, είτε ο πελάτης πρέπει να αποδείξει την ταυτότητα του στον εξυπηρέτη ή και τα δύο. Για αυτό, είτε ο εξυπηρέτης, είτε ο πελάτης ή και οι δύο θα πρέπει να κρατούν με ασφάλεια ένα ιδιωτικό κλειδί. Για τα κλειδιά του πελάτη, η φορητή έξυπνη κάρτα είναι ιδανικό μέσο. Μπορεί με ασφάλεια να αποθηκεύσει το ιδιωτικό κλειδί και να εκτελέσει κρυπτογραφικούς αλγορίθμους με αυτό.

Ο εξυπηρέτης μπορεί επίσης να χρησιμοποιήσει μία έξυπνη κάρτα για το ιδιωτικό κλειδί. Ωστόσο, ένα από τα πλεονεκτήματα της έξυπνης κάρτας είναι η φορητότητά της, το οποίο δεν είναι θέμα για τους εξυπηρέτες. Για το λόγο αυτό, είναι σύνηθες να χρησιμοποιούνται άλλες κρυπτογραφικές συσκευές στους εξυπηρέτες.

### 2.4.3 Ψηφιακές υπογραφές

Όταν αγοράσουμε κάποια αγαθά ή υπηρεσίες, πρέπει συχνά να υπογράψουμε ένα συμβόλαιο στο χαρτί για να επιβεβαιώσουμε ότι όντως κάναμε τη συγκεκριμένη παραγγελία και ότι δεσμευόμαστε να την εξοφλήσουμε. Αν κάνουμε την ίδια συμφωνία μέσω δικτύου, χρειαζόμαστε το ηλεκτρονικό ισοδύναμο της υπογραφής σε ένα χαρτί, δηλαδή μία ψηφιακή υπογραφή. Μία τέτοια υπογραφή θα πρέπει να εγγυάται ότι το πρόσωπο που την παράγει δε μπορεί να αρνηθεί στη συνέχεια την παραγγελία ή τη δήλωσή του.

Οι διάφορες μέθοδοι για ψηφιακές υπογραφές βασίζονται σε ένα ζεύγος ασύμμετρων κλειδιών. Η οντότητα που δημιουργεί την υπογραφή έχει ένα ιδιωτικό κλειδί, το οποίο δε μπορεί να προσπελαστεί ή χρησιμοποιηθεί από οποιοδήποτε άλλο. Ένα δεύτερο κλειδί, το οποίο αντιστοιχίζεται με το ιδιωτικό κλειδί είναι δημόσια γνωστό. Αυτό το κλειδί καλείται δημόσιο κλειδί. Ο μοναδικός κάτοχος του ιδιωτικού κλειδιού είναι ο μόνος που μπορεί να υπογράψει μία παραγγελία ή μία δήλωση, ενώ οποιοσδήποτε μπορεί να ελέγξει την υπογραφή, χρησιμοποιώντας το αντίστοιχο δημόσιο κλειδί.

Στις μέρες μας οι ηλεκτρονικές υπογραφές είναι πολύ δημοφιλείς, καθώς μπορούν να επιταχύνουν σημαντικά τις διαδικασίες στο ηλεκτρονικό εμπόριο. Είναι λοιπόν θέμα πρώτης προτεραιότητας να παραμένει το ιδιωτικό κλειδί πραγματικά ιδιωτικό. Αν κάποιος μπορούσε να αντιγράψει το ιδιωτικό κλειδί κάποιου άλλου, η ψηφιακή υπογραφή δε θα αντιστοιχούσε μοναδικά στον ιδιοκτήτη του κλειδιού. Για αυτό, το ιδιωτικό κλειδί πρέπει να αποθηκεύεται σε ένα πολύ ασφαλές μέρος, όπου κανείς δε θα μπορούσε να το αντιγράψει και μόνο ο ιδιοκτήτης του θα μπορούσε να το χρησιμοποιεί.

Τα πλέον ασφαλή μέρη για να αποθηκεύουμε τέτοιες πληροφορίες είναι μία κρυπτογραφική μονάδα υλικού, η οποία συχνά καλείται «cryptographic token». Η έξυπνη κάρτα μπορεί να θεωρηθεί η πιο βολική και φορητή τέτοια μονάδα. Οι μοντέρνες έξυπνες κάρτες μπορούν να παράγουν την ψηφιακή υπογραφή μέσα στην κάρτα. Παράλληλα, δεν παρέχουν κάποια συνάρτηση για να εξαχθεί το μυστικό κλειδί των υπογραφών στον έξω κόσμο. Η νομοθεσία σε μερικές ευρωπαϊκές χώρες απαιτεί ότι το κλειδί της υπογραφής θα παράγεται μέσα στην κάρτα. Το γεγονός αυτό, σε συνδυασμό με το ότι το κλειδί δε μπορεί να εξαχθεί από την κάρτα, κάνει αρκετά απίθανη την ύπαρξη ενός δεύτερου αντιγράφου του κλειδιού υπογραφής.

## 2.5 Κρυπτολογία

Στην ενότητα αυτή δίνουμε τους απαραίτητους ορισμούς από την επιστήμη της κρυπτολογίας. Παρουσιάζουμε επίσης τις βασικές αρχές των αλγορίθμων κρυπτογραφίας και τέλος εστιάζουμε στην κρυπτανάλυση, ώστε στην επόμενη ενότητα να παρουσιάσουμε τις επιθέσεις πλάγιου καναλιού και την κρυπτανάλυση που βασίζεται σε αυτή.

### 2.5.1 Ορισμοί

Η επιστήμη της κρυπτολογίας ασχολείται με τις μυστικές, συγκαλυμμένες ή κωδικοποιημένες επικοινωνίες. Η επιστήμη αυτή χωρίζεται σε δύο κλάδους, την κρυπτογραφία και την κρυπτανάλυση.

**Ορισμός 1.** *Η κρυπτογραφία είναι η τέχνη και η επιστήμη που ασχολείται με τις αρχές, έννοιες και μεθόδους που καθιστούν μία πληροφορία ακατανόητη και που ανακτούν μία πληροφορία από μία ακατανόητα κωδικοποιημένη μορφή.*

**Ορισμός 2.** *Η κρυπτανάλυση είναι η τέχνη και η επιστήμη που ασχολείται με τη μελέτη ακατανόητα κωδικοποιημένων μηνυμάτων και με τις λειτουργίες που πρέπει να εκτελεστούν ώστε να αποκτήσουν νόημα αυτά τα μηνύματα, χωρίς τη γνώση του αλγόριθμου ή/και του κλειδιού που χρησιμοποιήθηκε για την κωδικοποίηση σε αυτή τη μορφή.*

Από την εποχή του Καίσαρα, όπου ιστορικά αναφέρεται το πρώτο σύστημα κρυπτογραφίας, έως τις μέρες μας, η κρυπτολογία ήταν μία τέχνη, την οποία υπηρετούσαν άνθρωποι που επινοούσαν τεχνικές συγκεκριμένου σκοπού, για να ικανοποιήσουν και να παρακάμψουν τις συγκεκριμένες απαιτήσεις ασφάλειας που τίθονταν κάθε φορά. Τα τελευταία είκοσι χρόνια παρατηρείται όμως μία μετάβαση από το χώρο της τέχνης στο χώρο της επιστήμης, με έναυσμα το έργο του Claude Shannon στη θεμελίωση της Θεωρίας Πληροφορίας [50].

Ο στόχος ενός συστήματος κρυπτογραφίας είναι η ασφάλεια της πληροφορίας που διαχειρίζεται. Η ασφάλεια της πληροφορίας σε ένα σύστημα κρυπτογραφίας ορίζεται από ένα πλαίσιο τεσσάρων βασικών μορφών: εμπιστευτικότητα, ακεραιότητα δεδομένων, ταυτοποίηση και αδυναμία αποκήρυξης:

1. Η *εμπιστευτικότητα (confidentiality)* είναι μία υπηρεσία που χρησιμοποιείται για να διατηρήσουμε μία πληροφορία μακριά από όλους εκτός από όσους είναι εξουσιοδοτημένοι να την προσπελάσουν. Οι αγγλικοί όροι secrecy και privacy είναι όροι συνώνυμοι με την εμπιστευτικότητα

και χρησιμοποιούνται συχνά αντ' αυτής. Υπάρχουν πολυάριθμες προσεγγίσεις για την παροχή εμπιστευτικότητας, οι οποίες εκτείνονται από φυσική προστασία έως μαθηματικούς αλγόριθμους που μετατρέπουν τα δεδομένα σε ακατανόητη μορφή.

2. Η *ακεραιότητα δεδομένων (data integrity)* είναι μία υπηρεσία που εστιάζει στην αλλαγή των δεδομένων χωρίς εξουσιοδότηση. Για να εγγυηθεί κανείς την ακεραιότητα των δεδομένων, θα πρέπει να έχει τη δυνατότητα να ανιχνεύει παραποίηση των δεδομένων από μη εξουσιοδοτημένα μέρη. Η παραποίηση των δεδομένων περιλαμβάνει ενέργειες όπως εισαγωγή, διαγραφή και αντικατάσταση.

3. Η *ταυτοποίηση (authentication)* είναι μία υπηρεσία που σχετίζεται με την αναγνώριση. Η λειτουργία αυτή εφαρμόζεται τόσο σε οντότητες όσο και στην ίδια την πληροφορία. Σε μία διμερή επικοινωνία, θα πρέπει να αναγνωρίσει ο ένας τον άλλον. Για μία πληροφορία που παραδίδεται από ένα κανάλι επικοινωνίας θα πρέπει να ταυτοποιηθεί για παράδειγμα ο αποστολέας της, η ημερομηνία και ώρα αποστολής, το περιεχόμενο των δεδομένων ή ότι άλλο χρειάζεται κατά περίπτωση.

4. Η *αδυναμία αποκήρυξης (non-repudiation)* είναι μία υπηρεσία, η οποία εμποδίζει μία οντότητα να αρνηθεί προηγούμενες υποσχέσεις ή ενέργειες. Όταν δημιουργούνται διενέξεις εξαιτίας της άρνησης μίας οντότητας ότι προέβη σε συγκεκριμένες ενέργειες, θα πρέπει να υπάρχει ένας τρόπος να επιλυθούν. Για παράδειγμα, μία οντότητα μπορεί να εξουσιοδοτήσει μία άλλη οντότητα για την αγορά ενός αγαθού και στη συνέχεια να αρνηθεί ότι έδωσε αυτήν την εξουσιοδότηση. Μία διαδικασία, η οποία περιλαμβάνει μία έμπιστη τρίτη οντότητα, απαιτείται σε αυτήν την περίπτωση για να λυθεί η διένεξη.

Ο θεμέλιος στόχος της κρυπτογραφίας είναι να εστιάσει επαρκώς σε αυτές τις τέσσερις περιοχές, τόσο στη θεωρία, όσο και στην πράξη. Η κρυπτογραφία ασχολείται με την παρεμπόδιση και την ανίχνευση απατών και άλλων κακόβουλων πράξεων. Τα εργαλεία της κρυπτογραφίας χρησιμοποιούνται για να προστατεύουμε την πληροφορία. Εργαλεία κρυπτογραφίας είναι για παράδειγμα οι αλγόριθμοι κρυπτογράφησης, οι συναρτήσεις hash (hash functions) και τα σχήματα ψηφιακών υπογραφών.

### 2.5.2 Αλγόριθμοι κρυπτογραφίας

Στην ενότητα θα δώσουμε μία περίληψη των πλέον σχετικών αλγόριθμων κρυπτογραφίας που χρησιμοποιούνται σε συστήματα έξυπνων καρτών. Για μία εκτενέστερη καταγραφή αλγορίθμων κρυπτογραφίας αλλά και των θεωρητικών βάσεων της κρυπτογραφίας, παραπέμπουμε στο [38] και το [48].

Υπάρχουν δύο βασικές κατηγορίες αλγόριθμων κρυπτογραφίας, οι συμμετρικοί και οι ασύμμετροι. Οι συμμετρικοί αλγόριθμοι χρησιμοποιούν το ίδιο κλειδί για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων, για αυτό και αναφέρονται ως «αλγόριθμοι ιδιωτικού κλειδιού». Οι ασύμμετροι αλγόριθμοι χρησιμοποιούν δύο κλειδιά, το ένα για την κρυπτογράφηση και το άλλο για την αποκρυπτογράφηση. Για αυτό αναφέρονται και ως «αλγόριθμοι δημοσίου κλειδιού».

### 2.5.3 Συμμετρικοί αλγόριθμοι

Οι συμμετρικοί αλγόριθμοι κρυπτογραφίας είναι κατά κανόνα γρήγοροι και μπορούν να χρησιμοποιηθούν για την κρυπτογράφηση και αποκρυπτογράφηση μεγάλου όγκου δεδομένων. Ωστόσο, το γεγονός ότι χρησιμοποιείται το ίδιο κλειδί για την κρυπτογράφηση και αποκρυπτογράφηση, δημιουργεί ένα πρόβλημα όταν οι συμμετρικοί αλγόριθμοι χρησιμοποιούνται για να εγγυηθούμε την εμπιστευτικότητα μίας επικοινωνίας. Ο αποστολέας και ο παραλήπτης ενός μηνύματος θα πρέπει να χρησιμοποιούν το ίδιο κλειδί, δηλαδή ο παραλήπτης και μόνο αυτός πρέπει να γνωρίζει το κλειδί του αποστολέα για να εγγυηθεί η εμπιστευτικότητα της επικοινωνίας. Κάθε παραλήπτης θα πρέπει να γνωρίζει τα κλειδιά όλων των πιθανών αποστολέων για να είναι σε θέση να αποκρυπτογραφεί όλα τα εισερχόμενα μηνύματα. Προφανώς, ο γεωμετρικά αυξανόμενος αριθμός των απαιτούμενων κλειδιών είναι μη ρεαλιστικός, αν ο αριθμός των οντοτήτων που πρέπει να επικοινωνήσουν σε ένα δίκτυο είναι μεγάλος. Η λύση στο πρόβλημα αυτό είναι ένας συνδυασμός συμμετρικών και ασύμμετρων κλειδιών, τον οποίο θα δούμε στην ενότητα 2.5.5.

#### DES

Ο DES είναι ένας αλγόριθμος κρυπτογράφησης πακέτου (block cipher), δηλαδή μοιράζει τα δεδομένα εισόδου σε πακέτα και εργάζεται σε ένα πακέτο κάθε φορά. Ο DES χρησιμοποιεί μέγεθος πακέτου 64 bit και μέγεθος κλειδιού 56 bit. Στην πραγματικότητα, τα κλειδιά του DES αναπαριστώνται με 64 bit. Τα πλεονάζοντα bit χρησιμοποιούνται ως bit ισοτιμίας και αγνοούνται κατά την πραγματική διαδικασία κρυπτογράφησης.

Το βασικό συστατικό του DES είναι μία αντικατάσταση, η οποία ακολουθείται από μία μετάθεση. Μαζί καλούνται ένας «γύρος». Ο DES έχει 16 γύρους, δηλαδή σε κάθε πακέτο 64 bit εφαρμόζονται διαδοχικά 16 αντικαταστάσεις και μεταθέσεις. Λόγω της επαναληπτικής φύσης του, ο DES μπορεί να υλοποιηθεί εύκολα σε υλικό (hardware).

Το μέγεθος του κλειδιού που χρησιμοποιείται στο DES είναι αρκετά μικρό πλέον. Δεδομένων των δυνατοτήτων των σημερινών υπολογιστών, ο αλγόριθμος DES προσφέρει μόνο μία μέση ασφάλεια στις μέρες μας. Στο συνέδριο RSA '99 ανακοινώθηκε ένας διαγωνισμός, όπου το κλειδί μήκους 56 bit βρέθηκε σε λιγότερο από 24 ώρες, παρά το γεγονός ότι χρειάστηκε να εξεταστεί πάνω από το αναμενόμενο 50% του χώρου κλειδιών, από ένα δίκτυο συνεργαζόμενων υπολογιστών.

### Τριπλός DES

Ο τριπλός DES βασίζεται στο DES, αλλά χρησιμοποιεί κλειδιά μήκους 112 bit. Το κλειδί χωρίζεται σε δύο κλειδιά μήκους 56 bit, $K_1$ και $K_2$. Τα δεδομένα κρυπτογραφούνται πρώτα με το κλειδί $K_1$, μετά αποκρυπτογραφούνται με το $K_2$ και τέλος κρυπτογραφούνται ξανά με το $K_1$. Αυτή η μέθοδος είναι γνωστή με το όνομα EDE, Encrypt-Decrypt-Encrypt. Για την αποκρυπτογράφηση ακολουθείται η αντίστροφη διαδικασία. Πρώτα τα δεδομένα αποκρυπτογραφούνται με το $K_1$, μετά κρυπτογραφούνται με το $K_2$ και τέλος αποκρυπτογραφούνται με το $K_1$.

Ο τριπλός DES παρέχει μεγάλο επίπεδο ασφάλειας. Οι άμεσες επιθέσεις (brute force attacks), όπως στον απλό DES δεν είναι ρεαλιστικές με τα σημερινά δεδομένα, γιατί ο χώρος κλειδιών αυξάνει γεωμετρικά με το μέγεθος του κλειδιού. Έτσι για να βρούμε ένα κλειδί 112 bit με εξαντλητική αναζήτηση, θα χρειαστούμε κατά μέσο όρο $2^{56}$ φορές περισσότερο χρόνο από ότι για να βρούμε ένα κλειδί 56 bit.

### Άλλοι συμμετρικοί αλγόριθμοι

Υπάρχουν πάρα πολλοί συμμετρικοί αλγόριθμοι εκτός του DES, όπως για παράδειγμα οι RC4, RC5, IDEA και Skipjack. Όλοι αυτοί οι αλγόριθμοι όμως είναι δυνητικά άσχετοι με τις έξυπνες κάρτες, καθώς ελάχιστες υλοποιήσεις τους είναι διαθέσιμες. Ο γενικός κανόνας είναι ότι οι έξυπνες κάρτες υλοποιούν τον αλγόριθμο DES ή τον τριπλό DES.

## 2.5.4 Ασύμμετροι αλγόριθμοι

Η βασική ιδέα που οδήγησε στους ασύμμετρους αλγόριθμους ήταν ότι τα κλειδιά ενός αλγορίθμου θα μπορούσαν να ήταν ζεύγη ενός κλειδιού κρυπτογράφησης και ενός κλειδιού αποκρυπτογράφησης και ότι θα ήταν αδύνατο να υπολογίσει κανείς από το ένα κλειδί το άλλο. Η αρχή αυτή εφευρέθηκε από τους Whitfield Diffie και Martin Hellman το 1976 και ανεξάρτητα από αυτούς από το Ralph Merkle το 1978.

Από τότε πολλοί αλγόριθμοι ασύμμετρης κρυπτογράφησης έχουν προταθεί, οι περισσότεροι από τους οποίους ήταν ανασφαλείς ή μη πρακτικοί. Όλοι οι ασύμμετροι αλγόριθμοι είναι πολύ αργοί συγκρινόμενοι με τους συμμετρικούς. Ο πολύ γνωστός αλγόριθμος RSA για παράδειγμα, είναι χίλιες φορές πιο αργός από το DES, όταν υλοποιείται σε υλικό και εκατό φορές πιο αργός όταν υλοποιείται σε λογισμικό για την κρυπτογράφηση της ίδιας ποσότητας δεδομένων. Ωστόσο οι ασύμμετροι αλγόριθμοι έχουν ένα μεγάλο πλεονέκτημα όταν χρησιμοποιούνται για να εξασφαλίσουν την εμπιστευτικότητα μίας επικοινωνίας: χρησιμοποιούν διαφορετικά κλειδιά για την κρυπτογράφηση και την αποκρυπτογράφηση. Έτσι το ένα κλειδί, το οποίο διατηρείται μυστικό, μπορεί να χρησιμοποιείται για την αποκρυπτογράφηση, ενώ το άλλο κλειδί, το οποίο μπορεί να δημοσιοποιηθεί, χρησιμοποιείται για την κρυπτογράφηση.

Με τον τρόπο αυτό μόνο ο κάτοχος του μυστικού κλειδιού μπορεί να αποκρυπτογραφήσει ένα κρυπτογραφημένο μήνυμα. Για το λόγο αυτό οι ασύμμετροι αλγόριθμοι καλούνται συχνά και αλγόριθμοι δημόσιου κλειδιού, καθώς ένα από τα δύο κλειδιά μπορεί να είναι δημόσιο.

Με τους ασύμμετρους αλγόριθμους δεν έχουμε το πρόβλημα που παρουσιάζουν οι συμμετρικοί αλγόριθμοι σε ένα δίκτυο. Κάθε οντότητα χρειάζεται να αποθηκεύσει το δικό της μυστικό κλειδί και όλα τα δημόσια κλειδιά μπορούν να αποθηκευθούν σε ένα ανοικτό κατάλογο, τον οποίο να προσπελαύνει η καθεμία κάθε φορά που θέλει να επικοινωνήσει με μία συγκεκριμένη οντότητα. Το σχήμα αυτό είναι πρακτικό ακόμη και σε πολύ μεγάλα δίκτυα.

### RSA

Ο αλγόριθμος RSA επινοήθηκε από τους Ron Rivest, Adi Shamir και Leonard Adleman το 1978 [46]. Βασίζεται στη δυσκολία παραγοντοποίησης του γινομένου δύο πολύ μεγάλων πρώτων αριθμών. Ο RSA χρησιμοποιεί ζεύγη κλειδιών, όπου το δημόσιο κλειδί αποτελείται από ένα ισοϋπόλοιπο $n$ και ένα δημόσιο εκθέτη $e$ και το ιδιωτικό κλειδί αποτελείται από το ίδιο ισοϋπόλοιπο $n$ και ένα ιδιωτικό εκθέτη $d$.

Τα δύο κλειδιά παράγονται από δύο μεγάλους πρώτους αριθμούς,$p$ και $q$, οι οποίοι επιλέγονται τυχαία. Για μέγιστη ασφάλεια συνίσταται τα μεγέθη των δύο αυτών αριθμών να είναι ίσα. Το ισοϋπόλοιπο $n$ υπολογίζεται ως το γινόμενο των δύο αυτών αριθμών

$$n = pq$$

Στη συνέχεια επιλέγεται ένα κλειδί κρυπτογράφησης $e$ έτσι ώστε το $e$ και το $(p-1)(q-1)$ να είναι σχετικά πρώτοι. Το κλειδί αποκρυπτογράφησης $d$ υπολογίζεται έτσι ώστε

$$ed \equiv 1 \pmod{(p-1)(q-1)} \Leftrightarrow d \equiv e^{-1} \pmod{(p-1)(q-1)}$$

Αν τώρα $p$ είναι το αρχικό κείμενο, με μέγεθος ίσο με το μέγεθος του $n$ και $y$ είναι το κρυπτογραφημένο κείμενο αντίστοιχα, οι παρακάτω εξισώσεις περιγράφουν τη διαδικασία κρυπτογράφησης και αποκρυπτογράφησης:

$$y \equiv x^e \bmod n$$

$$x \equiv y^d \bmod n$$

Καθώς το πρόβλημα τις παραγοντοποίησης του γινομένου δύο μεγάλων πρώτων αριθμών παραμένει πολύ δύσκολο να λυθεί, είναι ανέφικτο να υπολογίσουμε το ιδιωτικό κλειδί από το δημόσιο, δεδομένου ότι χρησιμοποιούμε αρκετά μεγάλους πρώτους. Στο [9] περιέχεται μίας λεπτομερής ανάλυση της ασφάλειας και των επιθέσεων στον RSA.

Οι έξυπνες κάρτες σήμερα χρησιμοποιούν κλειδιά μεγέθους 512, 768 και 1024 bit. Τα 512 bit δε θεωρούνται πολύ ασφαλές μέγεθος κλειδιού πλέον.

Τα 1024 1024 θεωρείται ότι προσφέρουν ένα λογικό επίπεδο ασφάλειας για εφαρμογές όπως οι ψηφιακές υπογραφές και η κρυπτογράφηση κειμένων.

### DSA

Το ίδρυμα NIST πρότεινε το 1991 τον αλγόριθμο Digital Signature Algorithm, DSA για να χρησιμοποιηθεί στο πρότυπο Digital Signature Standard, DSS. Η ασφάλεια του αλγόριθμου στηρίζεται σε ένα άλλο δύσκολο πρόβλημα, τον υπολογισμό διακριτών λογαρίθμων κατά ένα ισοϋπόλοιπο $m$. Το αρχικό πρότυπο όριζε το μήκος του $m$ σε 512 bit, αλλά όπως αποδείχθηκε δεν ήταν αρκετό. Σε νεότερη έκδοση του προτύπου, το μήκος του $m$ είναι μεταβλητό, από 512 έως 1024 bit με βήμα 64 bit.

Ο αλγόριθμος DSA δουλεύει ως εξής: το $p$ είναι ένας πρώτος αριθμός, από 512 έως 1024 bit με βήμα 64 bit. Το $q$ είναι ένας πρώτος αριθμός 160 bit, παράγοντας του $p-1$. Το $g = h^{(p-1)/q} \bmod p$, όπου $h < p-1$ τέτοιο ώστε $h^{(p-1)/q} \bmod p > 1$. Το $x$ είναι τέτοιο ώστε $x < q$ και το $y = g^x \bmod p$.

Τα $p, q, g$ είναι δημόσια πληροφορία, το $x$ είναι το ιδιωτικό κλειδί και το $y$ είναι το δημόσιο κλειδί. Η $H$ είναι μία μονόδρομη συνάρτηση hash (one-way hash function). Το πρότυπο DSS ορίζει τον αλγόριθμο SHA-1, Secure Hash Algorithm ως $H$. Ο αποστολέας ενός μηνύματος $m$ εκτελεί τις εξής πράξεις: Αρχικά παράγει ένα τυχαίο αριθμό $k < q$. Στη συνέχεια υπολογίζει το $r = (g^k \bmod p) \bmod q$ και $s = (k^{-1}(H(m) + xr)) \bmod q$.

Οι ποσότητες $r$ και $s$ είναι η υπογραφή του μηνύματος και στέλνονται μαζί με αυτό. Ο παραλήπτης επιβεβαιώνει την υπογραφή, κάνοντας τους εξής υπολογισμούς:

$$w = s^{-1} \bmod q$$

$$u_1 = H(m) \cdot w \bmod q$$

$$u_2 = rw \bmod q$$

$$v = ((g^{u_1} y^{u_2}) \bmod p) \bmod q$$

Η υπογραφή επιβεβαιώνεται αν $v = r$.

## 2.5.5 Υβριδικοί αλγόριθμοι

Όπως έχουμε δει στα προηγούμενα, οι συμμετρικοί αλγόριθμοι είναι γρήγοροι αλλά δεν είναι πρακτικοί σε δίκτυα που συνδέουν πολλές οντότητες. Οι αλγόριθμοι δημόσιου κλειδιού από την άλλη πλευρά είναι πρακτικοί σε τέτοια δίκτυα, αλλά έχουν το μειονέκτημα ότι είναι αργοί.

Ωστόσο μπορούμε να συνδυάσουμε τα πλεονεκτήματα και των δύο, ενώ ταυτόχρονα να αποφύγουμε τα μειονεκτήματά τους. Οι υβριδικοί αλγόριθμοι χρησιμοποιούν κλειδιά συνεδρίας (session keys) για συμμετρικούς αλγόριθμους με τυχαίο τρόπο. Τα κλειδιά αυτά θα χρησιμοποιηθούν για την ταχεία κρυπτογράφηση μεγάλου όγκου δεδομένων. Μετά, το κλειδί συνεδρίας κρυπτογραφείται με ένα ασύμμετρο αλγόριθμο. Το γεγονός ότι ο ασύμμετρος αλγόριθμος είναι αργός, δεν ενοχλεί σε αυτήν την περίπτωση, γιατί κρυπτογραφούμε μόνο το κλειδί συνεδρίας. Από την άλλη πλευρά του καναλιού επικοινωνίας χρησιμοποιείται η αντίστροφη διαδικασία, ώστε ο παραλήπτης να αποκτήσει το συγκεκριμένο κλειδί, το οποίο χρησιμοποιεί στη συνέχεια, όπως περιγράψαμε.

Τα περισσότερα συστήματα σήμερα χρησιμοποιούν ένα τέτοιο συνδυασμό συμμετρικών και ασύμμετρων αλγόριθμων.

## 2.6 Επιθέσεις πλάγιου καναλιού

Ο στόχος της κρυπτανάλυσης είναι η ανάκτηση πληροφορίας από ένα ακατανόητο μήνυμα. Στις μέρες μας χρησιμοποιούνται αλγόριθμοι κρυπτογράφησης, οι οποίοι στηρίζονται σε δύσκολα μαθηματικά προβλήματα, όπως η παραγοντοποίηση ενός αριθμού σε πρώτους παράγοντες ή ο υπολογισμός του διακριτού λογαρίθμου. Μέχρι στιγμής δεν έχουμε κάποιον αλγόριθμο που να λύνει αυτά τα προβλήματα «γρήγορα». Είναι γενική πεποίθηση ότι τέτοιο αλγόριθμοι δεν υπάρχουν, ωστόσο κάτι τέτοιο δεν έχει αποδειχθεί μέχρι στιγμής.

Τα τελευταία χρόνια έχει παρουσιαστεί μία πολύ πρωτότυπη ιδέα για τα συστήματα κρυπτογραφίας. Το σύστημα αντιμετωπίζεται ως μία βιολογική οντότητα, η οποία μπορεί να αλληλεπιδράσει με το περιβάλλον της και όχι ως ένα αφηρημένο μαθηματικό αντικείμενο. Αφού το σύστημα αλληλεπιδρά με το περιβάλλον του, είναι ενδιαφέρον να εξετάσουμε το πώς συμπεριφέρεται σε διάφορα εξωτερικά ερεθίσματα. Η έρευνα στην περιοχή αυτή απέδειξε ότι τα περισσότερα συστήματα είναι πολύ ευάλωτα σε τέτοιου είδους επιθέσεις, καθώς τα κρυπτογραφικά κλειδιά μπορούν να αποκαλυφθούν σε πολύ μικρό χρονικό διάστημα. Η τεχνική αυτή ονομάζεται «επίθεση πλάγιου καναλιού» (side-channel attack), γιατί χρησιμοποιεί ένα πλάγιο κανάλι πληροφορίας, το οποίο δεν υπάρχει στον αφηρημένο κόσμο των μαθηματικών, για να ανακτήσει τις πληροφορίες που επιθυμεί.

Ανάλογα με το σύστημα, το πλάγιο κανάλι πληροφορίας μπορεί να είναι η ηλεκτρομαγνητική ακτινοβολία που εκπέμπει η συσκευή, η θερμότητα που εκλύει, η κατανάλωση ισχύος, η τροφοδοσία με ετεροκλινές ρολόι (clock skew), η εισαγωγή λαθών, η λειτουργία εκτός προδιαγραφών, η μερική ή ολική καταστροφή δομικών στοιχείων· ο κατάλογος είναι μάλλον ανεξάντλητος, καθώς περιορίζεται μόνο από τη φαντασία του επιτιθέμενου.

Η ιδέα των επιθέσεων αυτών δεν είναι νέα. Παρόμοιες τεχνικές έχουν χρησιμοποιηθεί κατά το παρελθόν σε ποικίλα συστήματα, όπως για παράδειγμα για την ανάκτηση της εικόνας μίας οθόνης υπολογιστή από μεγάλη απόσταση [57] ή από το καλώδιο RS 232 της σειριακής θύρας ενός υπολογιστή [51]. Η πρωτοτυπία της ιδέας έγκειται στο ότι για πρώτη φορά χρησιμοποιήθηκαν τέτοιες τεχνικές για την ανάκτηση κρυπτογραφικών κλειδιών. Ένα σημαντικό χαρακτηριστικό αυτών των επιθέσεων είναι ότι υλοποιούνται με μικρό έως μέσο κόστος και χωρίς να απαιτείται εξειδικευμένος εξοπλισμός [27] [32] [2] [3].

Οι επιθέσεις πλάγιου καναλιού χωρίζονται σε δύο μεγάλες κατηγορίες, τις παθητικές και τις ενεργές. Ο διαχωρισμός αφορά τον τρόπο που υλοποιείται το πλάγιο κανάλι πληροφορίας. Στις παθητικές επιθέσεις, συνήθως μετρούμε κάποιο χαρακτηριστικό της συσκευής κρυπτογράφησης κατά την εκτέλεση του αλγόριθμου κρυπτογράφησης. Στη συνέχεια, με κατάλληλα στατιστικά μοντέλα, συσχετίζουμε την πληροφορία αυτή με το κλειδί του αλγόριθμου κρυπτογράφησης. Στις ενεργές επιθέσεις, η συσκευή κρυπτογράφησης υποχρεώνεται να λειτουργήσει υπό τον έλεγχο του επιτιθέμενου, ο οποίος επηρεάζει εσωτερικά της στοιχεία κατά την εκτέλεση του αλγόριθμου κρυπτογράφησης.

### 2.6.1 Παθητικές επιθέσεις

Οι παθητικές επιθέσεις παρουσιάστηκαν πρώτες χρονικά. Το 1996, ο Paul Kocher, όντας 24 χρόνων, παρατήρησε ότι οι έξυπνες κάρτες χρειάζονταν διαφορετικό χρόνο για την εκτέλεση ενός κρυπτογραφικού αλγορίθμου. Αναλύοντας περισσότερο τα δεδομένα, διαπίστωσε ότι υπάρχει μία ισχυρή εξάρτηση μεταξύ του χρόνου που απαιτείται για την εκτέλεση του αλγορίθμου και του κλειδιού του [30]. Εξετάζοντας προσεκτικά τον κώδικα διάφορων αλγόριθμων κρυπτογράφησης, όπως για παράδειγμα των Diffie-Hellman, RSA και DSS, διαπίστωσε ότι ακολουθούνται διαφορετικά μονοπάτια εκτέλεσης ανάλογα με το κάθε bit του κλειδιού, τα οποία δημιουργούσαν αυτή τη διαφορά.

Σε έναν κώδικα για έξυπνες κάρτες, τα μονοπάτια αυτά δημιουργούνται από εντολές διακλάδωσης υπό συνθήκη και από τη φύση των εντολών που εκτελούνται σε κάθε βήμα. Επειδή η έξυπνη κάρτα είναι μία συσκευή περιορισμένων πόρων, συνήθως γίνεται βελτιστοποίηση του κώδικα με ειδικούς μεταφραστές. Ο στόχος της βελτιστοποίησης είναι ταχύτερη εκτέλεση και το μικρό μέγεθος του τελικού κώδικα. Όπως αποδείχθηκε όμως, αυτές οι βελτιστοποιήσεις ήταν η πηγή του προβλήματος. Τα αποτελέσματα του Kocher επιβεβαιώθηκαν και πρακτικά, τόσο από τον ίδιο όσο και ανεξάρτητα [16].

Η βιομηχανία των έξυπνων καρτών αντέδρασε γρήγορα σε αυτές τις επιθέσεις, παρουσιάζοντας κάρτες, οι οποίες εκτελούσαν τους αλγόριθμους κρυπτογραφίας σε σταθερό χρόνο. Σύντομα, ο Kocher χρησιμοποίησε την κατανάλωση ισχύος μίας κάρτας για να εξάγει τα κρυπτογραφικά κλειδιά. Η μέθοδος αυτή είναι ακόμη ισχυρότερη, αφού επιτρέπει την εξαγωγή των κλειδιών ακόμη πιο εύκολα [31] [23] [40] [12]. Καθώς η κάρτα τροφοδοτείται εξωτερικά με την απαραίτητη ισχύ, είναι πολύ δύσκολο να καταλάβει αν κάποιος παρακολουθεί την κατανάλωση ισχύος της. Εξίσου δύσκολο πρόβλημα είναι να διατηρήσει την κατανάλωση ισχύος σε σταθερό επίπεδο, ώστε να μην διαρρέει πληροφορία για τα κλειδιά του αλγόριθμου.

Στις μέρες μας, η περιοχή των παθητικών επιθέσεων είναι αρκετά ενεργή, καθώς αξιολογούνται διαρκώς νέες τεχνικές διατήρησης σταθερής κατανάλωσης για τους γνωστούς αλγόριθμους [15] [59] [49]. Ταυτόχρονα, παρουσιάζονται νέοι τρόποι για τη συσχέτιση της παραμικρής διακύμανσης με το κλειδί του αλγόριθμου [1] [39] [14] [37] [47]. Η σοβαρότητα του θέματος τονίζεται από το γεγονός ότι στις απαιτήσεις του αλγόριθμου AES (του αντικαταστάτη του DES), η σταθερή κατανάλωση ισχύος ήταν από τις πλέον σημαντικές και αρκετοί αλγόριθμοι απέτυχαν σε αυτό [13] [8]. Ο αλγόριθμος Rijndael που επιλέχθηκε τελικά, λύνει το πρόβλημα αυτό με ένα κομψό τρόπο. Ο Rijndael εκτελεί παράλληλα τις ίδιες λειτουργίες στο ζεύγος δεδομένα-κλειδί και στο συμπληρωματικό αυτού. Με τον τρόπο αυτό η κατανάλωση ισχύος είναι σταθερή σε όλη την εκτέλεση.

### 2.6.2 Ενεργές επιθέσεις

Οι ενεργές επιθέσεις παρουσιάστηκαν μερικούς μήνες μετά τις παθητικές. Οι πρωτοπόροι στο χώρο αυτό ήταν οι Dan Boneh, Richard DeMillo και Richard J. Lipton, οι οποίοι συνεργάζονταν με την τότε Bellcore και νυν Telcordia. Για το λόγο αυτό η επίθεση που παρουσίασαν αναφέρεται στη βιβλιογραφία ως «επίθεση Bellcore» (Bellcore attack).

Η επίθεση βασίζεται στην εισαγωγή λαθών κατά τη διάρκεια της εκτέλεσης ενός κρυπτογραφικού αλγόριθμου. Σημειώνουμε ότι όλα τα ρεαλιστικά συστήματα έχουν μία πιθανότητα αστοχίας, ενώ υπάρχουν αρκετές τεχνικές για την εκούσια εισαγωγή λαθών ακόμα και σε συσκευές που αντιστέκονται σε κακόβουλη μεταχείριση (tamper-resistant devices) [27] [32] [2] [3]. Έδειξαν λοιπόν, ότι σε περιβάλλοντα όπου μπορούν να συμβούν παροδικά λάθη σε συγκεκριμένα σημεία, μπορούν να ανακτηθούν τα κλειδιά του αλγόριθμου RSA, όταν υλοποιείται με το Θεώρημα Κινεζικού Υπολοίπου (Chinese Remainder Theorem) και με τη μέθοδο Montgommery και των σχημάτων ταυτοποίησης Fiat-Shamir και Schnorr [10].

Η επίθεση Bellcore γνώρισε μεγάλη απήχηση στο χώρο της κρυπτανάλυσης και μέχρι σήμερα πολλοί ερευνητές έχουν παρουσιάσει αντίστοιχες επιθέσεις [6] [7] [34] [26] [25] [60]. Θα πρέπει όμως να τονίσουμε ότι τα αποτελέσματα μέχρι στιγμής είναι θεωρητικά. Δεν έχει παρουσιαστεί κάποια πρακτική επίθεση βασισμένη σε αυτά, παρά μόνο μία επιβεβαίωση της ορθότητας των επιθέσεων της Bellcore με χρήση προσομοίωσης [4].

Αρκετοί θεωρούν ότι αυτές οι επιθέσεις είναι ανέφικτες σε ρεαλιστικά συστήματα και ειδικά σε έξυπνες κάρτες [35]. Πιστεύουμε ότι η άποψη αυτή δεν είναι σωστή και ότι αυτό θα επιβεβαιωθεί στο πέρασμα του χρόνου, καθώς οι άμυνες στις παθητικές επιθέσεις θα εξελιχθούν τόσο, ώστε να είναι προτιμότερο να εφαρμόσει κανείς μία ενεργή επίθεση. Δε θα πρέπει να παραβλέψουμε επίσης την επίθεση των Biham και Shamir [7], οι οποίοι έχουν περιγράψει τις γενικές αρχές για την εφαρμογή ενεργών επιθέσεων σε τελείως άγνωστα κρυπτοσυστήματα. Έχουν δείξει πώς θα μπορούσε να χρησιμοποιηθεί η εισαγωγή μόνιμων λαθών για τη μερική ή και ολική ανακατασκευή μυστικών αλγορίθμων· μία αρκετά ισχυρή υπόθεση, η οποία χρήζει ιδιαίτερης προσοχής. και όχι άκριτη απόρριψη των ενεργών επιθέσεων.

# Κεφάλαιο 3

# Παθητικές επιθέσεις σε κάρτες μνήμης

*«Στη ζωή υπάρχουν δύο είδη ανθρώπων: οι αισιόδοξοι και οι απαισιόδοξοι. Οι απαισιόδοξοι έχουν συνήθως δίκιο. Αν όμως η ανθρωπότητα έφτασε ως εδώ, αυτό το χρωστάει στους αισιόδοξους.»*

*—Καρλ Πόππερ, Αυστριακός Φιλόσοφος*

## 3.1 Εισαγωγικά

Τα τελευταία χρόνια οι κάρτες μνήμης, οι πρόγονοι των έξυπνων καρτών, έχουν εξοπλιστεί με μονάδα ελέγχου και μονάδα αυθεντικοποίησης. Με τον τρόπο αυτό η δημιουργία πλαστών καρτών και διάφορες άλλες μορφές απάτης με κάρτες μνήμης έχουν εξαλειφθεί. Για την αυθεντικότητα της κάρτας, ένα μυστικό κλειδί αποθηκεύεται σε αυτήν. Η ανάγνωση του κλειδιού αυτού προστατεύεται από τη μονάδα ελέγχου, όπως φαίνεται στο σχήμα 3.1. Όταν η κάρτα χρησιμοποιείται σε μία συναλλαγή, προηγείται η διαδικασία αυθεντικοποίησης, όπως φαίνεται στο σχήμα 3.2. Η εξωτερική εφαρμογή, στην οποία έχει αποθηκευθεί εκ των προτέρων το μυστικό κλειδί της κάρτας, στέλνει μία πρόκληση (challenge). Η μονάδα αυθεντικότητας της κάρτας υπολογίζει και στέλνει μία απάντηση (response), χρησιμοποιώντας το μυστικό της κλειδί και την πρόκληση. Η εξωτερική εφαρμογή συγκρίνει τα δύο αποτελέσματα και αποφασίζει αν η κάρτα είναι αυθεντική, όποτε προχωράει η συναλλαγή, ή πλαστή, οπότε αρνείται τη συναλλαγή.

Η ασφάλεια του σχήματος στηρίζεται τόσο στη μυστικότητα του κλειδιού, όσο και στη μυστικότητα του αλγορίθμου που υπολογίζει την απάντηση. Για το πρώτο σκέλος έχουν προταθεί τεχνικές για την ανάκτηση των κλειδιών [2] [3] [32]. Το δεύτερο σκέλος παραμένει όμως ένα ανοικτό πρόβλημα. Για το συγκεκριμένο πρόβλημα που εξετάζουμε η εξαγωγή των κλειδιών δεν είναι αρκετή, καθώς ο αλγόριθμος είναι μυστικός. Ακόμη και αν έχουμε τα κλειδιά, δε μπορούμε να κατασκευάσουμε μία πλαστή κάρτα, αφού δε θα μπορεί να υπολογίζει τη σωστή απάντηση.

## 3.2 Ορισμός του προβλήματος

Το πρόβλημα που εξετάζουμε είναι η ανακατασκευή του μυστικού αλγόριθμου, ο οποίος χρησιμοποιείται στις κάρτες μνήμης. Πιστεύουμε ότι οι τεχνικές των παθητικών επιθέσεων και ειδικότερα η απλή ή διαφορική ανάλυση κατανάλωσης ισχύος, μπορούν να λύσουν αυτό το πρόβλημα. Η υπόθεσή μας στηρίζεται στη βεβαιότητα ότι οι κάρτες μνήμης δεν υλοποιούν μηχανισμούς προστασίας από τέτοιου είδους επιθέσεις. Στην περίπτωση αυτή θα αποδειχθεί για άλλη μία φορά ότι η ασφάλεια ενός συστήματος δεν πρέπει να στηρίζεται στη μυστικότητα του αλγόριθμου που χρησιμοποιεί, αλλά στη δυσκολία της αντιστροφής της διαδικασίας κρυπτογράφησης χωρίς τη γνώση του μυστικού κλειδιού.

### 3.2.1 Διαδικασία επίλυσης

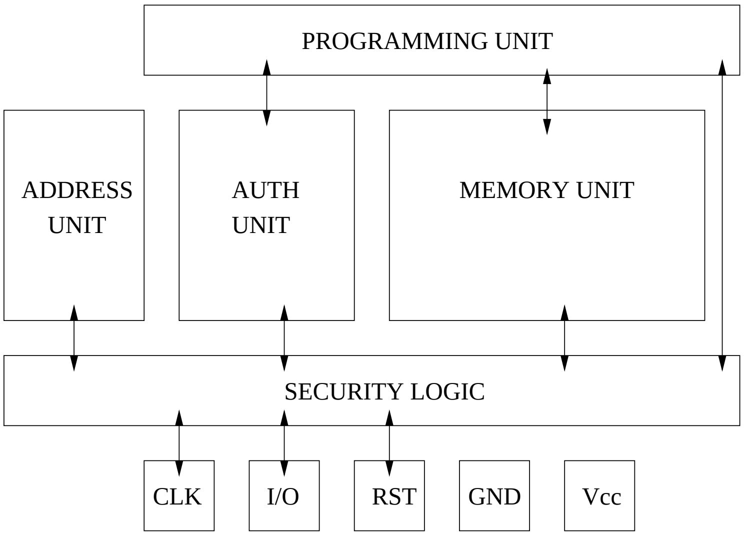
Οι Biham και Shamir έχουν περιγράψει μεθοδολογίες για την ανακατασκευή άγνωστων αλγορίθμων χρησιμοποιώντας επιθέσεις πλάγιου καναλιού σε έξυπνες κάρτες [7]. Πιστεύουμε ότι δεν είναι αναγκαίο να εφαρμόσουμε ενεργές επιθέσεις σε κάρτες μνήμης γιατί είναι αρκετά απλές στη λειτουργία, οπότε οι υποθέσεις για μία επιτυχή παθητική επίθεση θα είναι πολύ ασθενείς.

Θα θεωρήσουμε μία επίθεση τύπου «απλή ανάλυση κατανάλωσης ισχύος» ή «διαφορική ανάλυση κατανάλωσης ισχύος», αν χρειαστεί. Για να γίνει αυτό θα πρέπει καταρχήν να μοντελοποιήσουμε την κατανάλωση ισχύος της κάρτας μνήμης και στη συνέχεια να κάνουμε την κατάλληλη δειγματοληψία και ανάλυση των δεδομένων. Στο βήμα της ανάλυσης μπορούμε να επιβεβαιώσουμε την ορθότητα του μοντέλου μας ή να το τροποποιήσουμε κατάλληλα.

## 3.3 Μοντέλο κατανάλωσης ισχύος

Είναι γνωστό ότι η μονάδα αυθεντικοποίησης αποτελείται από ένα σύνολο πυλών XOR και NXOR. Είναι επίσης γνωστό ότι για τον υπολογισμό της απάντησης, η μονάδα αυθεντικοποίησης χρησιμοποιεί δεδομένα που είναι αποθηκευμένα στη μνήμη της κάρτας, δεδομένα που είναι αποθηκευμένα στην ίδια τη μονάδα (ξεχωριστή μνήμη) και την πρόκληση που στέλνεται από τον αναγνώστη.

Κατά τη διάρκεια ζωής μίας κάρτας μνήμης, η μόνη μεταβλητή πληροφορία που εισέρχεται στη μονάδα αυθεντικοποίησης είναι η πρόκληση από τον αναγνώστη. Όπως έχει παρατηρηθεί σε προηγούμενες εργασίες [8] [13] [31], το βάρος Hamming των δεδομένων ή ο αριθμός των αλλαγών $0 \leftrightarrow 1$ από το προηγούμενο δεδομένο επηρεάζει αναλογικά την κατανάλωση ισχύος. Στην




Σχήμα 3.1: Διάγραμμα μπλοκ μίας κάρτας μνήμης νεότερης γενιάς

περίπτωσή μας το μόνο δεδομένο που αλλάζει είναι η πρόκληση. Μάλιστα αυτό το δεδομένο δίνεται εξωτερικά, κατά συνέπεια μπορούμε να ελέγχουμε πλήρως το βάρος Hamming του. Έχει παρατηρηθεί επίσης ότι διαφορετικές πύλες συνεισφέρουν διαφορετική κατανάλωση ισχύος· μερικές πύλες μάλιστα καταναλώνουν επιπλέον ισχύ όταν αλλάζει η έξοδός τους.

Συνδυάζοντας τις παραπάνω πληροφορίες, θα θεωρήσουμε ότι η κατανάλωση ισχύος της μονάδας αυθεντικότητας σε μία δεδομένη χρονική στιγμή δίδεται από την εξής σχέση:

$$P[x] = \sum_x G_{XOR} + \sum_x G_{NXOR} + \sum G_{flip} + N$$

όπου:

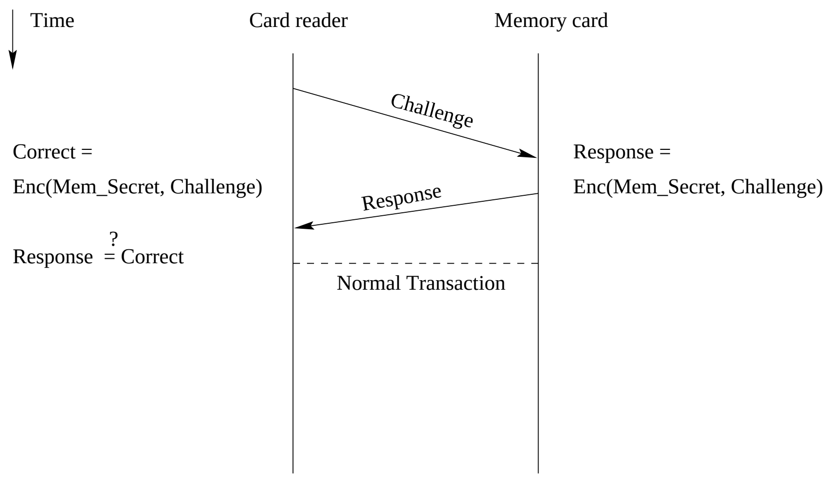
$\sum_x G_{XOR}$ Οι πύλες XOR που χρησιμοποιούνται λόγω της εισόδου $x$

$\sum_x G_{NXOR}$ Αντίστοιχα για τις πύλες NXOR

$\sum G_{flip}$ Η συνδρομή των πυλών λόγω της αλλαγής της εξόδου τους

$N$ Θόρυβος

Το παραπάνω μοντέλο μπορεί να περιγράψει τον συνολικό αριθμό πυλών που χρησιμοποιούνται σε κάθε βήμα του αλγορίθμου. Μπορεί επίσης να περιγράψει τις αλλαγές στις εισόδους των διαφόρων πυλών. Η πληροφορία αυτή δεν είναι αρκετή για να ανακατασκευάσουμε πλήρως τον αλγόριθμο, αλλά παρέχει μία αρκετά ποιοτική περιγραφή του. Πιστεύουμε ότι συνδυάζοντας αυτήν




Σχήμα 3.2: Πρωτόκολλο αυθεντικοποίησης κάρτας μνήμης

την περιγραφή με κατάλληλα επιλεγμένες εισόδους (προκλήσεις), μπορούμε καταρχήν να απαλείψουμε το παράγοντα του θορύβου και στη συνέχεια να εντοπίσουμε επακριβώς την αλληλεπίδραση μεταξύ των πυλών. Αυτή η πληροφορία θα είναι αρκετή για την πλήρη ανακατασκευή. Πράγματι γνωρίζοντας το συνολικό αριθμό και το είδος των πυλών που χρησιμοποιούνται κάθε χρονική στιγμή, μπορούμε να κατασκευάσουμε το διάγραμμα block του αλγορίθμου. Εντοπίζοντας στη συνέχεια τις αλληλεπιδράσεις των πυλών, μέσω του παράγοντα $\sum G_{flip}$, θα μπορέσουμε να ανακατασκευάσουμε τη συνδεσμολογία μεταξύ των πυλών. Η γνώση των πυλών και της συνδεσμολογίας τους περιγράφει ακριβώς τον αλγόριθμο κρυπτογράφησης που χρησιμοποιεί η κάρτα μνήμης.

## 3.4 Προβλήματα υλοποίησης

Για την υλοποίηση των παθητικών επιθέσεων σε κάρτες μνήμης απαιτείται ένας κατάλληλος αναγνώστης. Ο αναγνώστης αυτός θα πρέπει να είναι αρκετά ευέλικτος και προσαρμόσιμος, ώστε να μπορεί κάποιος να εφαρμόσει ελεγχόμενα πειράματα και μετρήσεις κατανάλωσης. Στην αγορά είναι διαθέσιμοι αρκετοί αναγνώστες καρτών μνήμης και έξυπνων καρτών. Ωστόσο κανείς δεν ικανοποιεί τις απαιτήσεις μας.

Η λειτουργία των έξυπνων καρτών έχει προτυποποιηθεί από το διεθνή οργανισμό ISO, στη σειρά προτύπων ISO 7816. Στα πρότυπα αυτά περιγράφεται ο ακριβής τρόπος επικοινωνίας μεταξύ ενός αναγνώστη έξυπνων καρτών και μίας έξυπνης κάρτας. Οι κάρτες αυτές χρησιμοποιούν ασύγχρονα πρωτόκολλα ανταλλαγής δεδομένων, τα οποία περιγράφονται επακριβώς στη σειρά των προτύπων.

Η λειτουργία των καρτών μνήμης δεν έχει προτυποποιηθεί ακόμη[1]. Για λόγους οικονομίας και απλότητας του υλικού, οι κάρτες μνήμης χρησιμοποιούν σύγχρονο (με το ρολόι) πρωτόκολλο ανταλλαγής δεδομένων. Άμεση απόρροια των παραπάνω είναι ότι υπάρχουν δύο διαφορετικές κατηγορίες αναγνωστών, με σύγχρονο και ασύγχρονο πρωτόκολλο μετάδοσης δεδομένων. Στην περίπτωση των σύγχρονων η κατάσταση είναι ακόμη πιο περίπλοκη, καθώς κάθε εταιρεία κατασκευής καρτών έχει υλοποιήσει διαφορετικό πρωτόκολλο επικοινωνίας και παρέχει διαφορετικό (ασύμβατο) αναγνώστη.

Είναι φανερό ότι για την επιτυχή κρυπτανάλυση καρτών μνήμης απαιτείται εξειδίκευση και εστίαση σε ένα συγκεκριμένο κατασκευαστή, το οποίο δεν είναι αποδεκτό για ένα εργαστηριακό περιβάλλον. Παράλληλα πολλές από τις εταιρείες αυτές δεν προσφέρουν ικανοποιητική πληροφόρηση για τη λειτουργία των αναγνωστών τους, γεγονός που δυσχεραίνει ακόμη περισσότερο την προσπάθεια για έναν «καθολικό» αναγνώστη καρτών.

### 3.4.1 Ο αναγνώστης Gemplus GCR 410

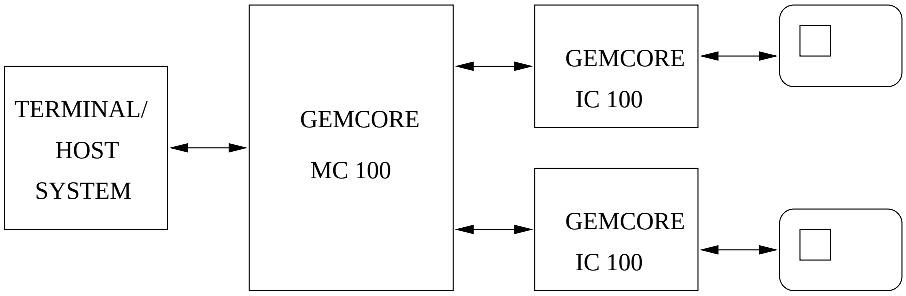
Ο αναγνώστης έξυπνων καρτών Gemplus GCR 410 συγκεντρώνει μία πληθώρα χαρακτηριστικών, η οποία τον καθιστά ιδανική επιλογή ως αναγνώστη καρτών. Ο αναγνώστης αυτός μπορεί να επικοινωνεί τόσο με ασύγχρονες όσο και σύγχρονες κάρτες. Το χαρακτηριστικό αυτό είναι ίσως μοναδικό ανάμεσα στις εμπορικά διαθέσιμες επιλογές. Επιπλέον παρέχει μία σειρά οδηγών (drivers) για όλα τα δημοφιλή λειτουργικά συστήματα, όπως Microsoft Windows, Linux, SUN Solaris. Για τους σκοπούς της εργασίας μας είναι τέλος μία ιδανική επιλογή, καθώς παρέχει δωρεάν τεκμηρίωση για όλες τις λειτουργίες του [19] [20] [21] [22]), ενώ με την υποστήριξη της εταιρείας κατασκευής έχει δημιουργηθεί μία κοινότητα προγραμματιστών, για τη διάχυση της γνώσης και της εμπειρίας από τον προγραμματισμό αυτών των συσκευών.

#### Περιγραφή λειτουργίας

Το σχηματικό διάγραμμα λειτουργίας του αναγνώστη έξυπνων καρτών Gemplus GCR 410 φαίνεται στο σχήμα 3.3

Με την αρχιτεκτονική αυτή διαχωρίζεται η λειτουργικότητα του αναγνώστη σε δύο διακριτά τμήματα. Ο μικροελεγκτής GemCore MC100 είναι υπεύθυνος για τον έλεγχο της επικοινωνίας με τις έξυπνες κάρτες και τη διασύνδεση του αναγνώστη με το εξωτερικό σύστημα ή τερματικό. Το ολοκληρωμένο GemCore IC100 είναι υπεύθυνο για την φυσική μετάδοση των δεδομένων από και προς την έξυπνη κάρτα. Έτσι είναι δυνατό να συνδεθούν περισσότερα του

[1] Ο οργανισμός ISO έχει ξεκινήσει την προτυποποίηση, στο πρότυπο ISO 7816-10. Δεν υπάρχει όμως ακόμη η τελική έκδοση του προτύπου. Ακόμη και τότε όμως, η πλειοψηφία των εφαρμογών αναμένεται να χρησιμοποιεί τις υπάρχουσες κάρτες, μίας και το κόστος της αναβάθμισης των συστημάτων είναι πολύ μεγάλο




Σχήμα 3.3: Σχηματικό διάγραμμα του Gemplus GCR 410

ενός ολοκληρωμένα πάνω στον ίδιο μικροελεγκτή και έτσι να ελέγχονται δύο έξυπνες κάρτες ταυτόχρονα.

#### Διεπαφή του GCR 410 με κάρτες μνήμης

Για την επικοινωνία με κάρτες μνήμης, ο αναγνώστης GCR 410 χρησιμοποιεί μία πολύ απλή ιδέα. Ο μικροελεγκτής GemCore MC100 υλοποιεί ένα διερμηνευτή της assembly του μικροελεγκτή Intel 8051. Κατάλληλα κωδικοποιημένες εντολές γλώσσας assembly ενθυλακώνονται σε κανονικές εντολές, όπως ορίζονται στο πρότυπο ISO 7816-4. Τα δεδομένα αυτά αναγνωρίζονται από τον μικροελεγκτή και διερμηνεύονται. Με τον τρόπο αυτό μπορεί να υλοποιηθεί πλήρως το σύγχρονο πρωτόκολλο μετάδοσης που απαιτείται για την επικοινωνία με μία κάρτα μνήμης. Το πλήρες σύνολο εντολών που υποστηρίζει ο διερμηνευτής περιγράφεται στο δεύτερο παράρτημα του [20].

## 3.5 Βιβλιοθήκη επικοινωνίας με κάρτες μνήμης

Έχοντας τις παραπάνω γνώσεις, το επόμενο βήμα είναι η δημιουργία μίας βιβλιοθήκης σε γλώσσα προγραμματισμού υψηλού επιπέδου για την πρόσβαση σε κάρτες μνήμης. Με τον τρόπο αυτό ο κρυπταναλυτής μπορεί να χρησιμοποιήσει την αντίστοιχη γλώσσα προγραμματισμού για να αναπτύξει τις τεχνικές του καθώς και να συγκεντρώσει τα διάφορα δείγματα για περαιτέρω ανάλυση σε μία ευέλικτη μορφή απεικόνισης. Τέλος με τη βιβλιοθήκη παρέχεται ένα επίπεδο αφαίρεσης, καθώς, όπως θα δούμε έχουν υλοποιηθεί ήδη οι βασικές λειτουργίες για την επικοινωνία με μία κάρτα μνήμης. Συνεπώς ο κρυπταναλυτής απαλλάσσεται από τον κόπο να ξαναγράψει τον κώδικα assembly για γνωστές λειτουργίες και αποφεύγει τη χρονοβόρα διαδικασία της αποσφαλμάτωσής του.

### 3.5.1 Περιβάλλον ανάπτυξης

Για την ανάπτυξη της βιβλιοθήκης εργαστήκαμε σε περιβάλλον Microsoft Windows 98 Second Edition, χρησιμοποιώντας του οδηγούς συσκευών της Gemplus. Η ανάπτυξη της βιβλιοθήκης ρουτινών έγινε στη γλώσσα προγραμματισμού C, χρησιμοποιώντας το περιβάλλον ανάπτυξης εφαρμογών Microsoft Visual Studio. Για τον έλεγχο και την αποσφαλμάτωση της βιβλιοθήκης χρησιμοποιήσαμε τον αναγνώστη έξυπνων καρτών Gemplus GCR410-X και κάρτες μνήμης SLE 4436 και 5536 της Siemens.

### 3.5.2 Η βιβλιοθήκη

Η βιβλιοθήκη αυτή τη στιγμή υλοποιεί τις εξής λειτουργίες:

- `MyGemplusLib_open_channel(Port, BaudRate, ChannelId)`
  Θέτει σε λειτουργία την κάρτα μνήμης, θεωρώντας ότι ο αναγνώστης είναι συνδεδεμένος στη σειριακή πόρτα `Port` και με ταχύτητα επικοινωνίας `BaudRate`

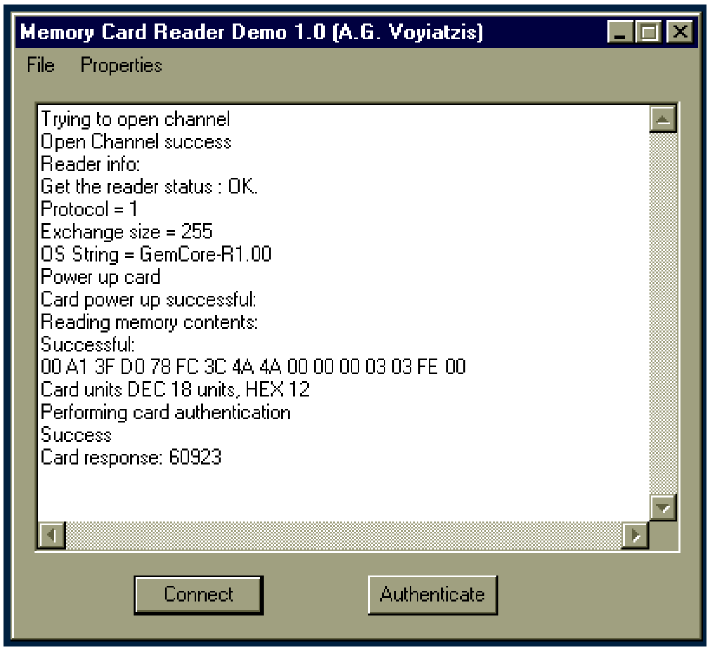
- `MyGemplusLib_get_ifd_status(INT16 ChannelId, char *debuginfo)`
  Επιστρέφει πληροφορίες ελέγχου για τον αναγνώστη.

- `MyGemplusLib_reset_card(INT16 ChannelId)`
  Ορίζει τις κατάλληλες παραμέτρους για επικοινωνία με κάρτες μνήμης και τροφοδοτεί την κάρτα με ρεύμα, ώστε να ξεκινήσει η λειτουργία της.

- `MyGemplusLib_simple_authenticate(INT16 ChannelId, WORD32 ReadLength, WORD8 ReadBuffer, WORD8 byte1, WORD8 byte2, WORD8 byte3, WORD8 byte4, WORD8 byte5, WORD8 byte6)`
  Εκτελεί το πρωτόκολλο αυθεντικοποίησης της κάρτας, δίνοντας 6 byte ως πρόκληση. Η απάντηση της κάρτας επιστρέφεται στο `ReadBuffer` και έχει μέγεθος `ReadLength`.

- `MyGemplusLib_read_memory(WORD8 StartAddress, WORD8 Length, INT16 ChannelId, WORD8 ReadBuffer[1024])`
  Επιστρέφει `Length` byte της μνήμης, ξεκινώντας από τη διεύθυνση `StartAddress`. Τα δεδομένα περιέχονται στον πίνακα `ReadBuffer`.

- `MyGemplusLib_helper(int response, char *message)`
  Επιστρέφει την περιγραφή του μηνύματος λάθους `response`. Βοηθητική ρουτίνα για την καλύτερη διάγνωση λαθών κατά την επικοινωνία με τις κάρτες μνήμης.




Σχήμα 3.4: Στιγμιότυπο εκτέλεσης της εφαρμογής επίδειξης

## 3.6 Επίδειξη λειτουργίας

Για να επιδείξουμε τη λειτουργία της βιβλιοθήκης αναπτύξαμε[2] μία εφαρμογή για το λειτουργικό περιβάλλον Microsoft Windows. Η εφαρμογή αυτή είναι σε θέση να εκτελέσει τις τρεις βασικές αλληλεπιδράσεις με μία κάρτα μνήμης:

- Έναρξη λειτουργίας
- Ανάγνωση περιεχομένων της μνήμης
- Εκτέλεση πρωτοκόλλου αυθεντικοποίησης

Στο σχήμα 3.4 απεικονίζεται ένα στιγμιότυπο εκτέλεσης της εφαρμογής, όπου έχουν εκτελεστεί οι τρεις παραπάνω λειτουργίες.

## 3.7 Ζητήματα υλοποίησης

Η βιβλιοθήκη επικοινωνίας είναι γραμμένη σε C. Ο κώδικας αποτελείται από 700 γραμμές περίπου. Το μεταγλωττισμένο αρχείο έχει μέγεθος 22KB περίπου. Για τη μεταγλώττιση της βιβλιοθήκης είναι απαραίτητη η εγκατάσταση

[2] Στην ανάπτυξη της εφαρμογής συνεργάστηκε ο Μανόλης Σπανάκης

των απαραίτητων οδηγών συσκευών (driver) της Gemplus και του αντίστοιχου API.

Η εφαρμογή επίδειξης είναι γραμμένη σε C++. Ο κώδικας αποτελείται από 550 γραμμές περίπου. Το εκτελέσιμο αρχείο έχει μέγεθος 82KB.

## 3.8 Σύνοψη

Στο τμήμα αυτό της εργασίας μας εντοπίσαμε το κατάλληλο υλικό, το οποίο συνδυάσαμε με μία βιβλιοθήκη λογισμικού, ώστε να δημιουργήσουμε ένα σύστημα, το οποίο μπορεί να επικοινωνεί με κάρτες μνήμης και κάρτες με μικροεπεξεργαστή. Το κύριο χαρακτηριστικό του συστήματος αυτού είναι η ευελιξία και η επεκτασιμότητά του. Από τις διαθέσιμες πηγές πληροφόρησης, δεν υπάρχει παρόμοια προσπάθεια μέχρι στιγμής.

Παράλληλα παρουσιάσαμε ένα θεωρητικά μοντέλο, το οποίο περιγράφει την κατανάλωση ισχύος μίας κάρτας μνήμης. Η έλλειψη του κατάλληλου εξοπλισμού στον περιορισμένο χρόνο της εργασίας αυτής, δε μας επέτρεψε να επιβεβαιώσουμε την ορθότητα του μοντέλου μας και να προχωρήσουμε στη συλλογή και ανάλυση των δειγμάτων κατανάλωσης ισχύος.

Είναι πεποίθησή μας ότι ο συνδυασμός των δύο προσπαθειών αυτών θα οδηγήσει σε επιβεβαίωση του θεωρητικού μοντέλου. Η επιβεβαίωση αυτή θα μας επιτρέψει στη συνέχεια να ανακατασκευάσουμε τους μυστικούς αλγόριθμους που χρησιμοποιούνται σε μονάδες αυθεντικοποίησης των καρτών μνήμης. Με τον τρόπο αυτό θα καταδείξουμε ότι οι κάρτες μνήμης δεν προσφέρουν ένα ικανοποιητικό επίπεδο ασφάλειας στις εφαρμογές που χρησιμοποιούνται.

# Κεφάλαιο 4

# Ενεργές επιθέσεις σε έξυπνες κάρτες

*Most discoveries do not begin with "Eureka!".*
*They begin with "That's odd. I wonder why that happened?"*

## 4.1 Εισαγωγή

Οι ενεργές επιθέσεις βασισμένες σε σφάλματα υλικού παρουσιάστηκαν για πρώτη φορά από τους Boneh, DeMillo και Lipton στο [10], όντας συνεργαζόμενοι με την τότε Bellcore (πλέον Telcordia). Για το λόγο αυτό συχνά η επίθεση αυτή αναφέρεται ως η «επίθεση της Bellcore» (Bellcore attack). Η κεντρική ιδέα των ενεργών επιθέσεων είναι η εξαγωγή μυστικών κλειδιών ενός κρυπτογραφικού αλγορίθμου, εισάγοντας λάθη κατά τη διάρκεια των υπολογισμών. Τα αποτελέσματα στην περιοχή των ενεργών επιθέσεων είναι προς το παρόν μόνο θεωρητικά. Δεν έχουν παρουσιαστεί επιτυχείς υλοποιήσεις τους σε ρεαλιστικά περιβάλλοντα.

Οι έξυπνες κάρτες παρουσιάζονται ως ο ιδανικός στόχος αυτών των επιθέσεων, καθώς δεν έχουν τη δυνατότητα να ανιχνεύουν εύκολα εξωτερικές παρεμβάσεις κατά τη διάρκεια της λειτουργίας τους. Παράλληλα, έχουν παρουσιαστεί τεχνικές για την εισαγωγή σφαλμάτων και την απενεργοποίηση των μηχανισμών προστασίας με αρκετά ικανοποιητικά αποτελέσματα [2] [3] [32].

Στο τμήμα αυτό της εργασίας εστιάσαμε στις ενεργές επιθέσεις στο σχήμα ταυτοποίησης Fiat-Shamir. Τα σχήματα ταυτοποίησης είναι ο θεμέλιος λίθος σε πρωτόκολλα συναλλαγών όπου χρησιμοποιούνται έξυπνες κάρτες. Πριν από κάθε συναλλαγή απαιτείται η ταυτοποίηση της κάρτας, ώστε να αποδεικνύεται ότι είναι αυθεντική, καθώς έχουν παρουσιαστεί αρκετές μορφές απάτης βασισμένες στην πλαστότητα της κάρτας. Το σχήμα Fiat-Shamir και το σχήμα Schnorr είναι τα δύο συνηθέστερα σχήματα ταυτοποίησης που χρησιμοποιούνται σε συστήματα με έξυπνες κάρτες.

Παρουσιάζουμε καταρχήν το τμήμα της επίθεσης της Bellcore, το οποίο αναφέρεται στο σχήμα Fiat-Shamir. Στη συνέχεια παρουσιάζουμε τη σχετική εργασία μας, η οποία αφορά τεχνικές προστασίας από αυτήν την επίθεση και το σχετικό επίπεδο ασφάλειας που προσφέρουν σε διάφορα περιβάλλοντα. Τέλος παρουσιάζουμε ένα νέο μοντέλο ενεργών επιθέσεων, με ενδιαφέροντες εφαρμογές.

## 4.2 Υπόβαθρο

### 4.2.1 Το σχήμα ταυτοποίησης Fiat-Shamir

Το πρωτόκολλο Fiat-Shamir [18] είναι ένα σχήμα ταυτοποίησης μηδενικής γνώσης δύο οντοτήτων. Η ταυτοποίηση της οντότητας γίνεται αποδεικνύοντας ότι κατέχει κάποια μυστική πληροφορία. Για ευκολία η οντότητα που αποδεικνύει την ταυτότητά της θα αναφέρεται ως Alice και η οντότητα που επαληθεύει την ταυτότητα ως Bob.

Με τον όρο «μηδενική γνώση» (zero-knowledge) υποδηλώνεται ότι η Alice δε φανερώνει τη μυστική πληροφορία που κατέχει στο Bob. Με τον τρόπο αυτό οποιαδήποτε οντότητα πιστοποιεί την Alice, δε μαθαίνει κάτι παραπάνω για αυτήν, από όσα ήδη ήξερε.

Το σχήμα δουλεύει ως εξής: η Alice έχει ορίσει ένα διαιρέτη (modulus) $N$ μεγέθους $n$ δυαδικών ψηφίων. Το $N$ είναι γινόμενο δύο πρώτων αριθμών. Έχει ορίσει επίσης ένα σύνολο από αντιστρέψιμα στοιχεία $s_1, s_2, \ldots, s_\ell \pmod N$. Το δημόσιο κλειδί της Alice είναι το σύνολο

$$PK_\ell = \{u_i \mid u_i = s_i^2 \pmod N) \ \textit{και}\ 1 \le i \le \ell\}$$

Η Alice αποδεικνύει την ταυτότητά της στο Bob, χρησιμοποιώντας το εξής πρωτόκολλο, αφού συμφωνήσουν κατ' αρχήν σε μία παράμετρο ασφαλείας $t \le l$:

1. **Commitment**. Η Alice διαλέγει ένα τυχαίο αριθμό $r \in \mathbb{Z}_N^*$ και στέλνει στο Bob το $z = r^2 \bmod N$
2. **Challenge**. Ο Bob διαλέγει ένα τυχαίο υποσύνολο $S \subseteq \{1, \ldots, t\}$ και το στέλνει στην Alice
3. **Response**. Η Alice υπολογίζει το $y = r \cdot \prod_{i \in S} s_i \bmod N$ και στέλνει το $y$ στο Bob
4. **Verify**. Ο Bob επαληθεύει την απάντηση της Alice, ελέγχοντας ότι $z \in \mathbb{Z}_N^*$ και ότι $y^2 = z \cdot \prod_{i \in S} u_i \bmod N$

Το πρωτόκολλο ολοκληρώνεται επιτυχώς αν η απάντηση επαληθευθεί και ανεπιτυχώς σε κάθε άλλη περίπτωση.

Η πιθανότητα να εξαπατηθεί ο Bob από κάποια πλαστή Alice, η οποία δεν κατέχει τη μυστική πληροφορία, είναι $2^{-t}$. Συνήθως το πρωτόκολλο επαναλαμβάνεται μερικές φορές (π.χ. τέσσερις) για να μειωθεί η πιθανότητα αυτή. Χρησιμοποιώντας $t = 10$ και επαναλαμβάνοντας το πρωτόκολλο τέσσερις φορές, η πιθανότητα εξαπάτησης γίνεται $2^{-40}$.

Η ασφάλεια του σχήματος στηρίζεται στο πρόβλημα εξαγωγής τετραγωνικών ριζών πάνω στο δακτύλιο $\mathbb{Z}_N$, το οποίο θεωρείται δύσκολο. Πιστεύεται ότι το πρόβλημα είναι ισοδύναμο με την παραγοντοποίηση του $N$ σε πρώτους παράγοντες. Δεν υπάρχει μέχρι σήμερα κάποια απόδειξη ότι τα δύο αυτά προβλήματα είναι πράγματι δύσκολα.

### 4.2.2 Η επίθεση της Bellcore στο σχήμα Fiat-Shamir

Η επίθεση της Bellcore υποθέτει ότι κατά τη διάρκεια των υπολογισμών της Alice είναι δυνατή η εισαγωγή ενός παροδικού σφάλματος. Για να επιτευχθεί αυτό σε μία ρεαλιστική υλοποίηση, απαιτείται η επίλυση ενός σύνθετου προβλήματος, της «απομόνωσης χώρου και χρόνου» (time-space isolation problem). Το σύστημα που εκτελεί το πρωτόκολλο Fiat-Shamir θα πρέπει να απομονωθεί τόσο χρονικά, ώστε να γίνει τη σωστή στιγμή το λάθος, όσο και χωρικά, ώστε το σφάλμα να συμβεί στο σωστό σημείο του συστήματος. Στην περίπτωση που το πρωτόκολλο υλοποιείται σε μία έξυπνη κάρτα, πιστεύεται ότι το πρόβλημα είναι ευκολότερο να λυθεί, λόγω των μικρών δυνατοτήτων της κάρτας σε μνήμη και επεξεργαστική ισχύ, συγκρινόμενη για παράδειγμα με ένα τυπικό προσωπικό υπολογιστή.

Η επίθεση της Bellcore θεωρεί ότι το παροδικό λάθος συμβαίνει στο χρόνο ανάμεσα στο πρώτο και το τρίτο βήμα του πρωτοκόλλου. Το λάθος είναι η αλλαγή ενός δυαδικού ψηφίου στη μνήμη όπου είναι αποθηκευμένος ο προσωρινός τυχαίος αριθμός $r$. Είναι σημαντικό ότι στο συγκεκριμένο πρωτόκολλο, ο επιτιθέμενος λύνει το πρόβλημα απομόνωσης χρόνου. Στο δεύτερο βήμα του πρωτοκόλλου είναι αυτός ο οποίος -σε λογικά πλαίσια- ορίζει το χρόνο αποστολής του υποσυνόλου $S$. Με τον τρόπο αυτό γνωρίζει ακριβώς πότε πρέπει να εισάγει το λάθος και καλείται να εντοπίσει μόνο σε ποιό σημείο της μνήμης της έξυπνης κάρτας είναι αποθηκευμένο το $r$. Το θεωρητικό μοντέλο επίθεσης της Bellcore συνοψίζεται στο εξής θεώρημα:[1]

**Θεώρημα 1.** *Έστω $N$ ένα ακέραιο υπόλοιπο μεγέθους $n$ δυαδικών ψηφίων και γινόμενο δύο πρώτων αριθμών. Έστω $\ell$ η παράμετρος ασφαλείας του πρωτοκόλλου Fiat-Shamir. Δεδομένων $\ell$ εκτελέσεων του πρωτοκόλλου με λάθη, τα μυστικά $s_1, \dots, s_\ell$ μπορούν να ανακτηθούν σε όσο χρόνο απαιτείται για να γίνουν $O(n\ell + \ell^2 \log \ell)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο (modular multiplications).*

*Περίληψη απόδειξης.* Έστω ότι συνέβη μία δυαδική αναστροφή, στη θέση $i$, όπου $i \in \{0, 1, \dots, n-1\}$ του $r$. Η αρχική τιμή του $r$ αλλάζει σε $r + E$, όπου

[1] Η απόδειξη βασίζεται στο [11], αναθεωρημένη έκδοση της αρχικής δημοσίευσης [10]

$E = \pm 2^i$. Το πρόσημο εξαρτάται από τη συγκεκριμένη αναστροφή, «0 σε 1» ή «1 σε 0». Όταν συμβεί η αναστροφή, η Alice υπολογίζει και στέλνει στο Bob μία λάθος τιμή του $y$, έστω $\hat{y}$, στο τρίτο βήμα του πρωτοκόλλου:

$$\hat{y} = (r + E) \cdot \prod_{i \in S} s_i$$

Από αυτήν, ο Bob μπορεί να υπολογίσει:

$$T(S) = \prod_{i \in S} s_i = \frac{2E \cdot \hat{y}}{\frac{\hat{y}^2}{\prod_{i \in S} u_i} - z + E^2} \bmod N$$

Στο βήμα αυτό θα πρέπει να υπολογίσουμε το $\prod_{i \in S} u_i$, το οποίο απαιτεί $O(\ell)$ πολλαπλασιασμούς κατά ισοϋπόλοιπο. Στη συνέχεια πρέπει να εντοπίσουμε τη θέση λάθους $i$, ώστε να κρατήσουμε το ορθό $T(S)$, δοκιμάζοντας όλες τις δυνατές θέσεις. Συνολικά λοιπόν, χρειαζόμαστε $O(n + \ell)$ πολλαπλασιασμούς κατά ισοϋπόλοιπο για ένα $T(S)$ και άρα για τα $\ell$ διαφορετικά απαιτούνται $O(n\ell + \ell^2)$.

Από τη στιγμή που ο Bob μπορεί να υπολογίζει $T(S)$ για διάφορα $S$ της επιλογής του, μπορεί να ανακτήσει τα $s_1, \ldots, s_\ell$. Στην απλούστερη περίπτωση ο Bob ορίζει τα $S$ ως μονοσύνολα. Τότε $S = \{s_k\}$, $T(S) = \prod_{i \in S} s_i = s_k$ και άρα βρίσκει άμεσα το $s_k$.

Στην περίπτωση που η Alice απορρίπτει μονοσύνολα $S$, ο Bob μπορεί και πάλι να ανακτήσει τα $s_i$ εργαζόμενος ως εξής: Αναπαριστούμε ένα υποσύνολο $S \subseteq \{1, 2, \ldots, \ell\}$ με το χαρακτηριστικό διάνυσμά του $U \in \{0,1\}^\ell$, δηλαδή $U_i = 1$ αν $i \in S$ και $U_i = 0$ αλλιώς. Διαλέγουμε τα $S$ έτσι ώστε τα χαρακτηριστικά τους διανύσματα, $U_1, U_2, \ldots, U_\ell$ να είναι γραμμικά ανεξάρτητα ή ισοδύναμα να ορίζουν ένα $\ell \times \ell$ πίνακα τάξης $\ell$ πάνω από το $\mathbb{Z}_2$.

Για να εξάγουμε το $s_1$, υπολογίζουμε $a_1, a_2, \ldots, a_\ell \in \{0, 1\}$ έτσι ώστε:

$$a_1 U_1 + \ldots + a_\ell U_\ell = (1, 0, 0, \ldots, 0) \pmod 2$$

Όταν οι υπολογισμοί γίνουν πάνω στους ακέραιους, τότε έχουμε:

$$a_1 U_1 + \ldots + a_\ell U_\ell = (2b_1 + 1, 2b_2, 2b_3, \ldots, 2b_\ell)$$

για γνωστές τιμές των $b_1, b_2, \ldots, b_\ell$. Έτσι υπολογίζουμε το $s_1$ από τη σχέση:

$$s_1 = \frac{T_1^{a_1} \cdots T_\ell^{a_\ell}}{u_1^{b_1} \cdots u_\ell^{b_\ell}} \pmod N$$

Οι υπολογισμοί είναι ανάλογοι για τα υπόλοιπα $s_i$. Το συνολικό κόστος για αυτό το τμήμα του αλγόριθμου είναι $O(\ell^2 \log \ell)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο.

Συνοψίζοντας λοιπόν, το συνολικό κόστος του αλγόριθμου ανάκτησης των μυστικών κλειδιών είναι $O(n\ell + \ell^2 \log \ell)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο.

□

## 4.3 Εγκυρότητα της επίθεσης Bellcore

### 4.3.1 Εισαγωγικά

Στο παραπάνω θεωρητικό μοντέλο παρατηρείται ότι το σχήμα Fiat-Shamir είναι πολύ ευάλωτο στην περίπτωση που η Alice αποδέχεται μονοσύνολα. Για το λόγο αυτό γίνεται η λογική υπόθεση ότι η Alice αρνείται να δεχθεί τέτοια μονοσύνολα. Ωστόσο παρουσιάζεται στο δεύτερο μέρος της επίθεσης τεχνική για την ανάκτηση των μυστικών κλειδιών της Alice ακόμη και στην περίπτωση όπου η Alice δέχεται μόνο σύνολα $S$ τέτοια ώστε $|S| \geq 2$. Η δυνατότητα της Alice να απορρίπτει μονοσύνολα ως προκλήσεις αποτέλεσε το έναυσμα για την εργασία μας. Η υπόθεση μας συνίσταται στο ότι η Alice μπορεί να κρίνει και να αποφασίζει ποιά σύνολα $S$ είναι αποδεκτά ως προκλήσεις. Αξιολογούμε λοιπόν την αποτελεσματικότητα της επίθεσης της Bellcore κάνοντας την υπόθεση ότι η Alice δέχεται μόνο συγκεκριμένα μεγέθη συνόλων $S$. Σημειώνουμε ότι η επίθεση της Bellcore δέχεται αξιωματικά ότι ένας πίνακας πλήρους τάξης είναι πάντα κατασκευάσιμος πάνω στο $\mathbb{Z}_2$.

Στη συνέχεια θα συμβολίζουμε με $G = \{n_1, n_2, \ldots, n_k\}$ το σύνολο των αποδεκτών μεγεθών του συνόλου $S$ από την Alice.

**Παράδειγμα 1**

Έστω άρτιο $\ell$ και $\{2, \ell - 1\} \subseteq G$. Είναι εύκολο να επαληθεύσει κανείς ότι ο παρακάτω πίνακας χαρακτηριστικών διανυσμάτων $B_e$ έχει τάξη $\ell$ πάνω στο $\mathbb{Z}_2$:

$$B_e = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{\ell-1} \\ b_{le} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & \ldots & 0 & 1 \\ & & \ddots & & \\ 0 & 0 & \ldots & 1 & 1 \\ 1 & 1 & \ldots & 1 & \mathbf{0} \end{bmatrix}$$

□

**Παράδειγμα 2**

Αντίστοιχα για περιττό $\ell$ και $\{2, \ell - 1\} \subseteq G$, ο παρακάτω πίνακας χαρακτηριστικών διανυσμάτων $B_o$ έχει τάξη $\ell$ πάνω στο $\mathbb{Z}_2$:

$$B_o = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{\ell-1} \\ b_{lo} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & \ldots & 0 & 1 \\ & & \ddots & & \\ 0 & 0 & \ldots & 1 & 1 \\ 1 & 1 & \ldots & 1 & \mathbf{1} \end{bmatrix}$$

□

Συμπερασματικά λοιπόν σε αυτές τις περιπτώσεις το μοντέλο επίθεσης της Bellcore είναι έγκυρο, αφού ένας πίνακας πλήρους τάξης είναι κατασκευάσιμος.

**Αντιπαράδειγμα**

Έστω $\ell = 3$ και $G = \{2\}$. Παρατηρούμε ότι σε αυτήν την περίπτωση $\{2, \ell\} \not\subseteq G$. Σε αυτό το παράδειγμα, υπάρχουν μόνο τρία χαρακτηριστικά διανύσματα: $(1,0,1), (0,1,1)$ και $(1,1,0)$. Επιπλέον ισχύει ότι $(1,0,1) + (0,1,1) = (1,1,0)$. Έτσι ο μόνος δυνατός $\ell \times \ell$ πίνακας

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

έχει τάξη 2 και όχι 3, όπως απαιτείται. Συνεπώς η επίθεση της Bellcore δεν είναι έγκυρη σε αυτήν την περίπτωση, καθώς δεν είναι δυνατό να κατασκευάσουμε ένα πίνακα πλήρους τάξης, όπως αξιώνεται. □

### 4.3.2 Χρήσιμα αποτελέσματα

Τα παραπάνω παραδείγματα υποδεικνύουν ότι υπάρχει κάποια σχέση μεταξύ του βάρους Hamming των χαρακτηριστικών διανυσμάτων, $w(U) = \sum U_i$. Θα εγκαθιδρύσουμε στη συνέχεια τη σχέση αυτή. Για την ανάλυσή μας θα χρησιμοποιήσουμε τους παρακάτω συμβολισμούς: με $V_2(\ell)$ θα συμβολίζουμε το σύνολο των διανυσμάτων του $\mathbb{Z}_2^\ell$ με άρτιο βάρος Hamming· με $X(\ell, k)$ θα συμβολίζουμε το υποσύνολο του $\mathbb{Z}_2^\ell$, το οποίο περιέχει όλα τα διανύσματα με βάρος Hamming ακριβώς $k$.

Αποδεικνύουμε καταρχήν δύο προτάσεις:

**Πρόταση 1.** *Για κάθε $a, b \in \mathbb{Z}_2^n$ το βάρος Hamming του αθροίσματός τους είναι:*

- **άρτιο**, *αν $w(a), w(b)$ είναι και τα δύο άρτια ή και τα δύο περιττά.*
- **περιττό**, *αλλιώς.*

*Απόδειξη.* Το άθροισμα δυαδικών διανυσμάτων είναι το δυαδικό αποκλειστικό-ή (binary XOR) των προσθετέων. Συνεπώς το βάρος Hamming του $a + b = a \oplus b$ είναι:

$$\begin{aligned} w(a \oplus b) &= \sum_{i=1}^{n} (a_i b_i^{'} + a_i^{'} b_i) = \sum_{i=1}^{n} \Big( a_i(1 - b_i) + (1 - a_i) b_i \Big) = \\ &= \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} a_i b_i + \sum_{i=1}^{n} b_i - \sum_{i=1}^{n} a_i b_i = w(a) + w(b) - 2 \sum_{i=1}^{n} a_i b_i \end{aligned}$$

Αφού ο όρος $2 \sum_{i=1}^{n} a_i b_i$ είναι άρτιος, το $w(a \oplus b)$ εξαρτάται μόνο από το $w(a)$ και $w(b)$. Αν είναι και τα δύο άρτια ή περιττά, το άθροισμά τους είναι άρτιο·

τότε είναι άρτιο και το $w(a \oplus b)$. Σε κάθε άλλη περίπτωση το άθροισμά τους είναι περιττό, άρα και το $w(a \oplus b)$ είναι περιττός αριθμός. $\square$

**Πρόταση 2.** *Ο $V_2(\ell)$ είναι υπόχωρος του $\mathbb{Z}_2^\ell$ με διάσταση $dim(V_2(\ell)) = \ell - 1$.*

*Απόδειξη.* Ο χώρος $\mathbb{Z}_2^\ell$ ορίζεται ως $< \mathbb{Z}_2^\ell, \mathbb{Z}_2, \oplus, \cdot >$, όπου $\oplus$ είναι το δυαδικό αποκλειστικό-ή (ή ισοδύναμα πρόσθεση κατά ισοϋπόλοιπο 2). Ο $V_2(\ell)$ είναι υπόχωρος, αφού:

- $(0, \ldots, 0) \in V_2(\ell)$
- Για οποιαδήποτε $u, v \in V_2(\ell)$, $u \oplus v \in V_2(\ell)$, από την Πρόταση 1
- $0 \cdot u = (0, \ldots, 0)$ και $1 \cdot u = u$ ανήκουν $V_2(\ell)$

Τα διανύσματα $\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b}_{\ell-\mathbf{1}}$ (οι πρώτες $\ell - 1$ γραμμές των πινάκων $B_e$ και $B_o$) είναι γραμμικά ανεξάρτητα:

$$\begin{aligned}
\mathbf{b_1} &= (1,0,0,\ldots,0,1)\\
\mathbf{b_2} &= (0,1,0,\ldots,0,1)\\
\mathbf{b_3} &= (0,0,1,\ldots,0,1)\\
&\vdots\\
\mathbf{b}_{\ell-\mathbf{1}} &= (0,0,0,\ldots,1,1)
\end{aligned}$$

Έστω τώρα ένα διάνυσμα $\mathbf{a} = (a_1, a_2, \ldots, a_\ell)$ του $V_2(\ell)$. Μπορούμε να εκφράσουμε το διάνυσμα αυτό ως γραμμικό συνδυασμό των $b_i$:

$$\mathbf{a} = a_1\mathbf{b_1} + a_2\mathbf{b_2} + \ldots + a_{l-1}\mathbf{b_{l-1}}$$

Αν και το άθροισμα περιέχει μόνο $\ell - 1$ διανύσματα $b_i$, $1 \leq i < l$ και η τιμή $a_\ell$ δε χρησιμοποιείται στους υπολογισμούς, το άθροισμα οδηγεί στη σωστή τιμή για το $a_\ell$. Θα το αποδείξουμε διακρίνοντας δύο περιπτώσεις:

- Αν $a_\ell = 1$, τότε το $w((a_1,\ a_2,\ \ldots, a_{\ell-1}))$ είναι περιττός αριθμός. Αφού $\mathbf{b_i}(\ell) = 1$ για κάθε $\mathbf{b_i}$, $1 \leq i < \ell$, στο αποτέλεσμα αθροίζεται περιττός αριθμός 1 στην τελευταία ($\ell$-οστή) θέση. Άρα $a_\ell = 1$, όπως απαιτείται.
- Αν $a_\ell = 0$, τότε το $w((a_1,\ a_2,\ \ldots, a_{\ell-1}))$ είναι άρτιος αριθμός. Με όμοιο σκεπτικό όπως προηγουμένως, έπεται ότι $a_\ell = 0$, όπως απαιτείται.

Συνοψίζοντας λοιπόν το σύνολο $\{\mathbf{b_1},\ \mathbf{b_2},\ \ldots,\ \mathbf{b}_{\ell-\mathbf{1}}\}$ περιέχει $\ell - 1$ γραμμικά ανεξάρτητα διανύσματα. Η θήκη (span) αυτών των διανυσμάτων ορίζει το $V_2(\ell)$. Εξ ορισμού λοιπόν, τα διανύσματα αυτά ορίζουν μία βάση του υπόχωρου και η διάστασή του είναι $\ell - 1$. $\square$

### 4.3.3 Το Προσεκτικό Σχήμα Fiat-Shamir

Ορίζουμε μία παραλλαγή του αυθεντικού σχήματος Fiat-Shamir, όπου αλλάζει ελαφρά το τρίτο βήμα (Response) του πρωτοκόλλου επικοινωνίας, όπως παρουσιάστηκε στην ενότητα 4.2.1:

**Ορισμός 3 (Το Προσεκτικό Σχήμα Fiat-Shamir).** *Το σχήμα αυθεντικοποίησης Fiat-Shamir επαυξημένο με ένα σύνολο $G$ καλείται «προσεκτικό», αν η Alice δέχεται στο τρίτο βήμα του πρωτοκόλλου σύνολα $S$, αν και μόνο αν $|S| \in G$.*

Από τον ορισμό βλέπουμε ότι αν $G = \{1, 2, \ldots, \ell\}$, τότε το σχήμα ταυτίζεται με το αυθεντικό σχήμα Fiat-Shamir. Αν τώρα $G \subset \{1, 2, \ldots, \ell\}$, υποστηρίζουμε ότι το σχήμα προσφέρει ισοδύναμη ασφάλεια με το αρχικό. Η ασφάλεια του σχήματος στηρίζεται στη δυσκολία παραγοντοποίησης ενός γινομένου πάνω στο $\mathbb{Z}_N$ και στο φαινόμενο διάχυσης (diffusion) του τυχαίου αριθμού $r$. Η ασφάλεια του αρχικού σχήματος δε στηρίζεται στον ακριβή αριθμό των παραγόντων ενός δεδομένου γινομένου. Ο Ορισμός 3 δεν αποκλείει οποιαδήποτε επιλογή ενός συγκεκριμένου $s_i$, αλλά εισάγει περιορισμούς στον αριθμό των παραγόντων μίας απάντησης $y$ του πρωτοκόλλου επικοινωνίας. Δεν υπάρχει κάποια γνωστή εργασία μέχρι στιγμής, όπου ο συνολικός αριθμός των παραγόντων ενός αριθμού στο $\mathbb{Z}_N$ να αποκαλύπτει οποιοδήποτε πληροφορία σχετική με τους παράγοντές του.

Το Προσεκτικό Σχήμα Fiat-Shamir παρέχει καλά χαρακτηριστικά άμυνας απέναντι στην επίθεση της Bellcore, όπως αποδεικνύει το παρακάτω θεώρημα:

**Θεώρημα 2.** *Αν η Alice υλοποιεί το Προσεκτικό Σχήμα Fiat-Shamir και το $G$ αποτελείται μόνο από άρτιους αριθμούς, τότε η επίθεση της Bellcore δεν είναι έγκυρη.*

*Απόδειξη.* Η επίθεση της Bellcore θα είναι έγκυρη αν μπορεί να κατασκευαστεί ένας $\ell \times \ell$ πίνακας με τάξη $\ell$, περιέχοντας σαν στήλες ή γραμμές, τα διανύσματα του $V_2(\ell)$. Η διάσταση του $V_2(\ell)$ είναι $\ell - 1$, σύμφωνα με την Πρόταση 2. Άρα οποιαδήποτε $\ell$ διανύσματα του $V_2(\ell)$ είναι γραμμικά εξαρτημένα και άρα οποιοσδήποτε πίνακας με στήλες ή γραμμές τέτοια διανύσματα θα έχει τάξη το πολύ $\ell - 1$. □

### 4.3.4 Το επεκτεταμένο μοντέλο επίθεσης Bellcore

Σύμφωνα με το Θεώρημα 2, το Προσεκτικό Σχήμα Fiat-Shamir παρέχει καλά χαρακτηριστικά ασφάλειας απέναντι στην επίθεση της Bellcore. Παρουσιάζουμε στη συνέχεια μία τροποποίηση της επίθεσης, η οποία μπορεί να ανακτήσει αρκετή μυστική πληροφορία από την Alice, ώστε να μπορεί στη συνέχεια ο επιτιθέμενος να την πλαστοπροσωπήσει. Αυτό αποδεικνύει ότι και το Προσεκτικό Σχήμα Fiat-Shamir είναι ευάλωτο σε ενεργές επιθέσεις.

Έστω ότι η Alice υλοποιεί το Προσεκτικό Σχήμα Fiat-Shamir χρησιμοποιώντας ως $G$ ένα σύνολο άρτιων αριθμών. Θεωρούμε ότι στο πρώτο βήμα της επίθεσης Bellcore εισάγονται λάθη κατά τη διάρκεια των υπολογισμών κατά τέτοιο τρόπο ώστε τα χαρακτηριστικά διανύσματα των συνόλων $S$ που δόθηκαν ως προκλήσεις να είναι γραμμικά ανεξάρτητα. Για να επιτευχθεί αυτό απαιτούνται $\ell - 1$ λάθη. Σε αυτήν την περίπτωση, μετά το πέρας της πρώτης φάσης του αλγορίθμου εξαγωγής των μυστικών κλειδιών, θα έχουμε στην κατοχή μας $\ell - 1$ γραμμικά ανεξάρτητα διανύσματα. Αυτό είναι εγγυημένο από την Πρόταση 2. Η πληροφορία αυτή είναι αρκετή για να πλαστοπροσωπήσει κάποιος την Alice, καθώς τα διανύσματα αυτά θα παράγουν όλο τον υπόχωρο $V_2(\ell)$.

**Παράδειγμα**

Για να γίνει κατανοητό αυτό, θα χρησιμοποιήσουμε το Αντιπαράδειγμα της ενότητας 4.3.1: Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι μετά από δύο επαναλήψεις του πρωτοκόλλου με λάθη, έχουμε ανακαλύψει τα $s_1 s_2$ και $s_1 s_3$. Τα αντίστοιχα χαρακτηριστικά διανύσματα είναι τα $(1,1,0)$ και $(1,0,1)$, τα οποία είναι γραμμικά ανεξάρτητα. Μπορούμε να εκφράσουμε το τρίτο διάνυσμα, το οποίο αντιστοιχεί στο $s_2 s_3$, ως γραμμικό συνδυασμό των δύο αυτών: $(0,1,1) = a_1(1,1,0) + a_2(1,0,1)$. Συνεπώς $a_1 = a_2 = 1$. Αντίστοιχα υπολογίζουμε ότι $b_1 = 1, b_2 = 0, b_3 = 0$. Άρα υπολογίζουμε το $s_2 s_3$ ως:

$$\frac{(s_1 s_2)^{a_1}(s_1 s_3)^{a_2}}{u_1^{b_1} u_2^{b_2} u_3^{b_3}} \pmod N = \frac{s_1^2 s_2 s_3}{u_1} \pmod N = s_2 s_3 \pmod N$$

Άρα μετά δύο επαναλήψεις του πρωτοκόλλου, έχουμε συγκεντρώσει όλες τις δυνατές απαντήσεις που μπορεί να δημιουργήσει η Alice. Υπενθυμίζουμε ότι είναι η Alice που ελέγχει τον τυχαίο αριθμό $r$ στο πρώτο βήμα του πρωτοκόλλου επικοινωνίας. □

### 4.3.5 Ασφάλεια του Προσεκτικού Σχήματος Fiat-Shamir

Στην προηγούμενη υποενότητα είδαμε ότι μία απλή τροποποίηση της επίθεσης Bellcore είναι αρκετή ώστε να παράγουμε αρκετή πληροφορία για να πλαστοπροσωπήσουμε την Alice. Για μία ρεαλιστική υλοποίηση όμως προκύπτουν θέματα που χρήζουν περαιτέρω ανάλυσης, τα οποία και εξετάζουμε.

Όταν η Alice υλοποιεί το Προσεκτικό Σχήμα Fiat-Shamir χρησιμοποιώντας ως $G$ ένα σύνολο που περιέχει μόνο άρτιους αριθμούς, δεν είναι δυνατή η εξαγωγή μοναδιαίων χαρακτηριστικών διανυσμάτων (δυαδικών διανυσμάτων που περιέχουν ένα μόνο 1). Αυτό σημαίνει ότι στο δεύτερο βήμα του αλγορίθμου εξαγωγής μυστικών κλειδιών δε μπορούμε να παράγουμε κάθε $s_i$ ξεχωριστά, όπως στην αρχική επίθεση της Bellcore. Στην καλύτερη περίπτωση μπορούμε να παράγουμε γινόμενα $s_i s_j$. Η παρατήρηση αυτή οδηγεί σε ένα ενδιαφέρον πρόβλημα για την πλαστοπροσωπία, καθώς ο επιτιθέμενος θα πρέπει να επιλέξει μία από τις ακόλουθες τακτικές: είτε θα υπολογίσει εκ των προτέρων όλα τα δυνατά γινόμενα είτε θα υλοποιήσει την εξαγωγή των μυστικών κλειδιών δυναμικά, κατά τη διάρκεια της συναλλαγής, χρησιμοποιώντας τα $\ell - 1$ γινόμενα που υπολόγισε στο πρώτο βήμα του αλγορίθμου.

Στην πρώτη περίπτωση, είναι δυνατόν με κατάλληλη επιλογή του $G$ να ορίζουμε ένα πολύ μεγάλο αριθμό γινομένων, ώστε να απαιτείται τελικά εκθετικός αριθμός υπολογισμών. Αν το $G$ περιέχει όλους τους άρτιους αριθμούς έως το $\ell$, $G = \{2, 4, 6, \ldots, \ell | \ell - 1\}$, τότε υπάρχουν

$$\sum_{i=2}^{\ell \ or \ \ell-1} \binom{\ell}{i}$$

δυνατά υποσύνολα, οπότε τελικά θα απαιτούνται $O(2^\ell)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο για μία πλήρη πλαστοπροσωπία.

Στη δεύτερη περίπτωση, τη δυναμική εξαγωγή των κλειδιών, δεν υπάρχει αυτό το πρόβλημα, καθώς κάθε φορά θα πρέπει να υπολογίζεται μόνο ένα γινόμενο. Ωστόσο θα πρέπει να υλοποιηθεί ο αλγόριθμος εξαγωγής των κλειδιών πάνω στην συσκευή που λειτουργεί ως Alice. Στην περίπτωση που αυτό είναι μία έξυπνη κάρτα, είναι δυνατό να τίθενται περιορισμοί από το σύστημα στο οποίο χρησιμοποιείται, ώστε να αποτυγχάνει η επίθεση. Για παράδειγμα η διαθέσιμη μνήμη ή επεξεργαστική ισχύς της κάρτας μπορεί να μην επαρκούν για την υλοποίηση του αλγόριθμου. Αυτό σημαίνει ότι θα πρέπει να χρησιμοποιηθεί ένας άλλος τύπος καρτών ή ένας εξομοιωτής μίας κάρτας. Και στις δύο περιπτώσεις όμως είναι πιθανόν το σύστημα να ανιχνεύει τέτοιες μορφές απάτης και να μην επιτρέπει την πραγματοποίηση των συναλλαγών. Αυτό μπορεί να γίνεται αν ο αναγνώστης λειτουργεί με την επίβλεψη κάποιου ανθρώπου.

Σε κάθε περίπτωση είναι φανερό ότι ακόμη και στην περίπτωση που χρησιμοποιείται το επεκτεταμένο μοντέλο επίθεσης της Bellcore, είναι δυνατό να υπάρξει προστασία του συστήματος από πλαστοπροσωπίες που οφείλονται στην εξαγωγή των μυστικών κλειδιών του σχήματος Fiat-Shamir από μία νόμιμη έξυπνη κάρτα.

## 4.4 Το μοντέλο ενεργών επιθέσεων S-TAM

Στην προηγούμενη ενότητα είδαμε ότι υπάρχουν διαμορφώσεις συστημάτων, οι οποίες αντιστέκονται στα γνωστά μοντέλα επιθέσεων, τόσο το αυθεντικό της Bellcore όσο και το επεκτεταμένο, το οποίο εισάγαμε. Στην ενότητα αυτή παρουσιάζουμε ένα νέο θεωρητικό μοντέλο επιθέσεων, το S-TAM. Το S-TAM είναι έγκυρο τόσο όταν εφαρμόζεται στο αυθεντικό σχήμα, όσο και στο Προσεκτικό Σχήμα Fiat-Shamir. Το S-TAM σε κάθε περίπτωση μπορεί να ανακτήσει τα μυστικά κλειδιά της Alice σε πολυωνυμικό χρόνο.

### 4.4.1 Μοντέλο λαθών

Θα θεωρήσουμε ότι κατά τη διάρκεια της εκτέλεσης του πρωτοκόλλου Fiat-Shamir είναι δυνατό να αναστραφεί παροδικά ένα δυαδικό ψηφίο, στο τρίτο βήμα. Πιο συγκεκριμένα θεωρούμε ότι αυτή η αναστροφή μπορεί να συμβεί σε οποιοδήποτε $s_i$, πριν η Alice το χρησιμοποιήσει για τον υπολογισμό της απάντησης $y = r\prod_{i \in S} s_i$. Είναι ενδιαφέρον να παρατηρήσουμε ότι σε αυτήν την περίπτωση ο επιτιθέμενος Bob θα πρέπει να λύσει το σύνθετο πρόβλημα απομόνωσης χώρου και χρόνου. Πράγματι, σε αυτό το μοντέλο λαθών ο Bob δε μπορεί να ξέρει τον απόλυτο χρόνο όπου χρησιμοποιείται το κάθε $s_i$ που τον ενδιαφέρει. Σε αυτό το πλαίσιο, η υπόθεσή μας για το μοντέλο λαθών είναι ισχυρότερη από την αντίστοιχη για την επίθεση της Bellcore, γιατί απαιτείται ακριβής συγχρονισμός με τη συσκευή που λειτουργεί ως Alice· αυτό σημαίνει στη γενική περίπτωση ότι η πιθανότητα εισαγωγής λάθους θα είναι μικρότερη από ότι για την επίθεση της Bellcore. Πιστεύουμε ωστόσο ότι η υπόθεσή μας είναι ρεαλιστική, όταν η συσκευή είναι η έξυπνη κάρτα, καθώς έχει μικρή επεξεργαστική ισχύ.

### 4.4.2 Η επίθεση S-TAM

Η επίθεση που προτείνουμε βασίζεται στο παραπάνω μοντέλο λαθών. Περιγράφεται δε στο παρακάτω θεώρημα:

**Θεώρημα 3 (Επίθεση S-TAM).** *Έστω $N$ ένα ακέραιο υπόλοιπο μεγέθους $n$ δυαδικών ψηφίων και γινόμενο δύο πρώτων αριθμών. Έστω $\ell$ η παράμετρος ασφαλείας του πρωτοκόλλου Fiat-Shamir. Δεδομένων $\ell$ εκτελέσεων του πρωτοκόλλου με λάθη, τα μυστικά $s_1, \ldots, s_\ell$ μπορούν να ανακτηθούν σε όσο χρόνο απαιτείται για να γίνουν $O(n\ell \log \ell)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο. (modular multiplications).*

*Απόδειξη.* Έστω ότι ένα λάθος συνέβη κατά την εκτέλεση του πρωτοκόλλου, σύμφωνα με το μοντέλο λαθών της υποενότητας 4.4.1. Ο Bob μπορεί να εντοπίσει ότι όντως συνέβη ένα λάθος, στο τέταρτο βήμα του πρωτοκόλλου. Υποθέτουμε ότι το λάθος συνέβη στο $s_j$. Για να ανακτήσουμε το $s_j$ εργαζόμαστε ως ακολούθως:

Αφού το λάθος ήταν μία δυαδική αναστροφή, το $s_j$ άλλαξε σε $s_j \pm 2^i$ για κάποιο $i$ με $0 \le i \le n-1$. Ο Bob γνωρίζει από τα βήματα του πρωτοκόλλου:

**Βήμα 1:**

$$r^2 \pmod N$$

**Βήμα 4:**

$$\hat{y} = r(s_j \pm 2^i) \prod_{k \in S - s_j} s_k \pmod N$$

Οι παρακάτω απλές πράξεις επιτρέπουν την ανάκτηση του $s_j$, δεδομένου ότι το λάθος συνέβη όντως στο $s_j$ (από την απάντηση $\hat{y}$ δε γνωρίζουμε σε ποιό $s_j$ συνέβη το λάθος):

$$\begin{aligned} C_1 &= \frac{\hat{y}^2}{r^2} \pmod N = (s_j \pm 2^i)^2 \prod_{k \in S - s_k} s_k^2 \pmod N & (4.1) \\ C &= \frac{C_1}{\prod_{i \in S} u_i} \pmod N = \frac{u_j + 2^{2i} \pm 2^{i+1} s_j}{u_j} \pmod N & (4.2) \\ s_j &= \pm \frac{u_j(C-1) - 2^{2i}}{2^{i+1}} \pmod N & (4.3) \end{aligned}$$

Για τον υπολογισμό της 4.1 απαιτούνται τρεις πολλαπλασιασμοί κατά ισοϋπόλοιπο. Για τον υπολογισμό της 4.2 απαιτούνται δύο πολλαπλασιασμοί κατά ισοϋπόλοιπο, καθώς την τιμή $\prod_{i \in S} u_i$ την έχουμε υπολογίσει στο τέταρτο βήμα του πρωτοκόλλου Fiat-Shamir (Verify) και μπορούμε να τη χρησιμοποιήσουμε απευθείας. Τέλος για τον υπολογισμό της 4.3 απαιτούνται $O(n)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο, για να εντοπίσουμε τη σωστή τιμή του $s_j$, βρίσκοντας την ορθή θέση σφάλματος $i$. Συνεπώς το κόστος αυτών των υπολογισμών είναι $O(n)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο.

Δεδομένου ότι ο Bob δε γνωρίζει σε ποιό $s_j$ έγινε το λάθος, θα πρέπει να δοκιμάσει όλα τα $|S|$ δυνατά $s_j$ για να το εντοπίσει. Άρα η συνολική πολυπλοκότητα για να εντοπίσουμε ένα $s_j$ είναι $O(n\ell)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο.

Δεδομένων τώρα $\ell$ επαναλήψεων του πρωτοκόλλου με λάθη, έτσι ώστε να συμβεί ένα λάθος σε κάθε $s_1, s_2, \ldots, s_\ell$, ο Bob μπορεί να ανακτήσει όλα τα μυστικά $s_j$ της Alice, στο χρόνο που απαιτούνται για να γίνουν $O(n\ell \log \ell)$ πολλαπλασιασμοί κατά ισοϋπόλοιπο. □

## 4.5 Αποτελέσματα

Στον πίνακα 4.1 συνοψίζεται η αποτελεσματικότητα για όλα τα μοντέλα επιθέσεων που εξετάσαμε. Σημειώνουμε ότι η πολυπλοκότητα για την επαυξημένη επίθεση Bellcore είναι η χείριστη δυνατή και αναφέρεται στην περίπτωση όπου το $G$ περιέχει όλους τους άρτιους αριθμούς έως το $\ell$ και ο αλγόριθμος εξαγωγής των κλειδιών υλοποιηθεί εκτός της κάρτας. Είναι ενδιαφέρον να παρατηρήσουμε ότι ασυμπτωτικά το μοντέλο επίθεσης S-TAM έχει τη χειρότερη πολυπλοκότητα. Ωστόσο είναι εγγυημένο να δουλέψει σε κάθε περίπτωση και συνεπώς μπορεί να χρησιμοποιηθεί όταν τα άλλα μοντέλα δε λειτουργούν.

Το κριτήριο για πολυπλοκότητα του κάθε μοντέλου ήταν η ακριβότερη σε χρόνο πράξη που χρειάζεται να γίνει. Αυτή είναι ο πολλαπλασιασμός κατά ισοϋπόλοιπο. Πειράματα προσομοίωσης σε ένα τυπικό προσωπικό υπολογιστή έδειξαν ότι ο χρόνος εκτέλεσης του αλγόριθμου εξαγωγής των κλειδιών με βάση την επίθεση Bellcore ήταν της τάξης των μερικών δευτερολέπτων ή λεπτών

[4]. Αυτό μας οδηγεί σε ένα σημαντικό συμπέρασμα: αν μία έξυπνη κάρτα υλοποιεί το σχήμα αυθεντικότητας Fiat-Shamir και το μοντέλο λαθών είναι εφικτό, τότε πρέπει να δίνεται ιδιαίτερη προσοχή στη διαμόρφωση του συνολικού συστήματος, ώστε να αποφεύγεται διαρροή ευαίσθητων πληροφοριών από την κάρτα.

| Επίθεση | Το $G$ περιέχει | Απαιτούμενα λάθη | Πολυπλοκότητα (πολλαπλασιασμοί κατά ισοϋπόλοιπο) |
|---|---|---|---|
| Bellcore | περιττούς | $\ell$ | $O(n\ell + \ell^2 \log \ell)$ |
| Bellcore | άρτιους | ΔΕ ΛΕΙΤΟΥΡΓΕΙ | ΔΕ ΛΕΙΤΟΥΡΓΕΙ |
| Επαυξημένη Bellcore | άρτιους | $\ell - 1$ | $O(2^\ell)$ |
| S-TAM | οτιδήποτε | $\ell$ | $O(n\ell \log \ell)$ |

Πίνακας 4.1: Αποτελεσματικότητα των μοντέλων ενεργών επιθέσεων

## 4.6 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάσαμε το μοντέλο ενεργών επιθέσεων της Bellcore για την εξαγωγή των μυστικών πληροφοριών που χρησιμοποιούνται στο σχήμα ταυτοποίησης Fiat-Shamir. Αποδείξαμε ότι με μία μικρή τροποποίηση του πρωτοκόλλου επικοινωνίας, μπορούμε να κατασκευάσουμε συστήματα ανθεκτικά σε αυτήν την επίθεση. Στη συνέχεια παρουσιάσαμε ένα επαυξημένο μοντέλο επιθέσεως, το οποίο, υπό συγκεκριμένες προϋποθέσεις, είναι δυνατό να παρακάμψει αυτό το μηχανισμό προστασίας. Τέλος εισάγαμε ένα νέο μοντέλο επιθέσεων, το S-TAM, το οποίο είναι έγκυρο σε κάθε περίπτωση και μπορεί να χρησιμοποιηθεί στις περιπτώσεις όπου η αυθεντική ή η επαυξημένη επίθεση της Bellcore δεν είναι έγκυρη. Καταλήγουμε λοιπόν στο συμπέρασμα ότι τα συστήματα όπου τα μοντέλα λαθών είναι εφικτά, ειδικά δε συστήματα που περιέχουν έξυπνες κάρτες, είναι ευάλωτα σε ενεργές επιθέσεις και άρα θα πρέπει να λαμβάνεται ιδιαίτερη μέριμνα για την αποφυγή τέτοιων επιθέσεων.

# Κεφάλαιο 5

# Έξυπνες κάρτες Java Card

*"The art of war teaches us to rely not on the likelihood of the enemy is not coming, but on our readiness to receive him; not on the chance of his not attacking, but rather on the face that we have made our position unassainable."*

—*The Art of War*, Sun Tzu

Η τρίτη γενιά έξυπνων καρτών περιλαμβάνει τις κάρτες στις οποίες το λογισμικό μπορεί να ανανεωθεί μετά την έκδοση τους. Στην ενότητα 2.1.10 παρουσιάσαμε τον κύκλο ζωής μίας κάρτας. Οποιοδήποτε πρόβλημα παρουσιαστεί στο λογισμικό μίας κάρτας μετά την έκδοσής της, στο παρόν μοντέλο δε μπορεί να αντιμετωπιστεί. Κατά συνέπεια ο εκδότης της κάρτας θα πρέπει να την αποσύρει και να εκδώσει νέα, με το διορθωμένο λογισμικό. Σε συστήματα όπου χρησιμοποιούνται χιλιάδες ή και εκατομμύρια έξυπνων καρτών, η κίνηση αντικατάστασης έχει μεγάλο κόστος, σε χρήματα και χρόνο. Ειδικά αν το πρόβλημα παρακάμπτει κάποια δικλείδα ασφαλείας του συνολικού συστήματος, το κόστος είναι σημαντικά μεγαλύτερο, καθώς επηρεάζεται και η φήμη της εταιρείας.

Η ραγδαία εξάπλωση της χρήσης έξυπνων καρτών ως ασφαλή υποσυστήματα σε συνδυασμό με την γεωγραφική διασπορά των χρηστών, καθιστά σημαντικό ζήτημα την απουσία σφαλμάτων στο λογισμικό των έξυπνων καρτών, αφού εκδοθούν. Παράλληλα η εξάπλωση αυτή αναμένεται να δημιουργήσει το φαινόμενο smart card juggling [5], δηλαδή ο τελικός χρήστης να είναι υποχρεωμένος να έχει στην κατοχή του μία πληθώρα έξυπνων καρτών, καθεμία από τις οποίες θα του παρέχει μία υπηρεσία ή ένα σύνολο υπηρεσιών από ένα συγκεκριμένο εκδότη. Έτσι θα είναι υποχρεωμένος να εναλλάσσει συνεχώς έξυπνες κάρτες προκειμένου να εκτελεί απρόσκοπτα τις εργασίες του.

Ένα παράδειγμα smart card juggling είναι το ακόλουθο: ένας χρήστης σε μία εταιρεία χρησιμοποιεί μία έξυπνη κάρτα για να μπορέσει να αποκτήσει πρόσβαση στον υπολογιστή του. Για την πραγματοποίηση μίας αγοράς μέσω του Διαδικτύου, θα πρέπει να χρησιμοποιήσει μία δεύτερη κάρτα, την πιστωτική, την οποία έχει εκδώσει η τράπεζα με την οποία συνεργάζεται. Προκειμένου να στείλει ένα εμπιστευτικό ηλεκτρονικό μήνυμα θα πρέπει να χρησιμοποιήσει μία τρίτη κάρτα, η οποία έχει εκδοθεί από έναν οργανισμό που πιστοποιεί την ταυτότητα του χρήστη, ώστε να μπορέσει να δημιουργήσει μία ψηφιακή υπογραφή. Για να χρησιμοποιήσει τέλος ένα συγκεκριμένο λογισμικό, θα πρέπει να εισάγει μία τέταρτη έξυπνη κάρτα, η οποία πιστοποιεί ότι είναι νόμιμος κάτοχος του προϊόντος. Σίγουρα ο εργαζόμενος αυτός δεν είναι καθόλου ευχαριστημένος από το περιβάλλον εργασίας του και η παραγωγικότητά του είναι αρκετά μειωμένη.

Στην ενότητα 2.2.2 παρουσιάσαμε την πλατφόρμα Java Card της εταιρείας SUN. Η Java Card παρέχει τη δυνατότητα να ενσωματωθούν όλες αυτές οι εφαρμογές μέσα σε μία μόνο κάρτα, διευκολύνοντας σημαντικά τη χρήση των έξυπνων καρτών. Παρέχει επίσης τη δυνατότητα ανανέωσης ή προσθήκης λογισμικού μετά την έκδοση μίας κάρτας και τις απαραίτητες δικλείδες ασφαλείας, ώστε να μην επιτρέπεται η αλληλεπίδραση μεταξύ εφαρμογών διαφορετικών κατασκευαστών που βρίσκονται μέσα σε μία κάρτα.

## 5.1 Μηχανισμοί ασφάλειας της Java Card

Η πλατφόρμα Java Card, ως γνήσιος απόγονος της τεχνολογίας Java, παρέχει εγγενώς αυξημένα χαρακτηριστικά ασφάλειας ως γλώσσα προγραμματισμού αλλά και περιβάλλον εκτέλεσης εφαρμογών. Η ιδιότητα αυτή προήλθε από την ανάγκη προστασίας τόσο του προγραμματιστή από τα κοινά λάθη, όσο και του χρήστη, ο οποίος καλείται να εκτελέσει εφαρμογές που μεταφέρονται από το Διαδίκτυο και τον Παγκόσμιο Ιστό, οπότε δε μπορεί να γνωρίζει την προέλευση και την ηθική τους. Η πλατφόρμα Java Card παρέχει μηχανισμούς ασφάλειας σε τέσσερα διακριτά επίπεδα:

- Προγραμματισμός
- Εικονική μηχανή
- Περιβάλλον εκτέλεσης
- Εγκατάσταση εφαρμογών

### Προγραμματισμός

Το Java Card API ακολουθεί τις βασικές αρχές της γλώσσας Java και παραμένει μία γλώσσα χωρίς δείκτες (pointers) και παρόμοιες προγραμματιστικές «ευκολίες», οι οποίες είναι συνήθης πηγή προβλημάτων ασφαλείας.

### Εικονική μηχανή

Η εικονική μηχανή εκτός από τη διερμήνευση των εντολών εκτελεί και μία δεύτερη λειτουργία. Αυτή είναι ο έλεγχος της λογικής ορθότητας του προγράμματος που εκτελείται. Δεδομένου ότι σε μία κάρτα εγκαθίστανται εφαρμογές από πολλούς και πιθανόν άγνωστους κατασκευαστές, δεν είναι δυνατό να γνωρίζει η εικονική μηχανή το πώς ακριβώς έχουν παραχθεί. Το Java Card API είναι αρκετά προστατευτικό, ώστε να υπάρχουν εσκεμμένα λάθη, ωστόσο μία εφαρμογή μπορεί να έχει παραχθεί χωρίς τη χρήση του πρότυπου μεταγλωττιστή της SUN. Θα μπορούσε να μην έχει χρησιμοποιηθεί καν μεταγλωττιστής, αλλά να έχει παραχθεί με κάποιο πλάγιο τρόπο το «εκτελέσιμο» της εφαρμογής. Επίσης σε κάθε περίπτωση δε θα πρέπει να παραμελούμε ότι κάθε λογισμικό ενδέχεται να έχει κάποιο λογικό σφάλμα (bug)· ο εκάστοτε μεταγλωττιστής είναι και αυτός ένα λογισμικό.

Στην πλατφόρμα Java η λειτουργία ελέγχου της λογικής ορθότητας ενός προγράμματος είναι δουλειά μίας ξεχωριστής ίνας (thread), του `Security Manager`. Σε συσκευές περιορισμένων πόρων όμως, όπως οι έξυπνες κάρτες, δεν υπάρχει πάντα η δυνατότητα πολυνηματικής επεξεργασίας. Για το λόγο αυτό, προτιμήθηκε η ενσωμάτωση αυτής της λειτουργικότητας στην Εικονική Μηχανή.

### Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης της Java Card (Java Card Runtime Environment, JCRE), είναι το τρίτο επίπεδο ασφάλειας. Περιλαμβάνει την υλοποίηση της εικονικής μηχανής καθώς και τις κλάσεις του Java Card API. Το JCRE είναι υπεύθυνο για τον έλεγχο μίας εφαρμογής κατά τη διάρκεια εκτέλεσής της. Για κάθε εφαρμογή Java Card (applet), ορίζεται μία «περίμετρος δικαιοδοσίας», κατά αναλογία με το sandbox στη Java. Όταν εγκαθίσταται μία εφαρμογή σε μία έξυπνη κάρτα, ορίζονται ταυτόχρονα και τα δικαιώματά της. Για παράδειγμα ορίζονται οι διευθύνσεις μνήμης που επιτρέπεται να προσπελάσει και τα αρχεία στα οποία έχει δικαίωμα πρόσβασης. Κατά την εκτέλεση της εφαρμογής, κάθε εντολή που διερμηνεύεται από την εικονική μηχανή, ελέγχεται ταυτόχρονα από το JCRE για την ορθότητά της και για το αν έχει δικαίωμα εκτέλεσης.

### Εγκατάσταση εφαρμογών

Η δυνατότητα εγκατάστασης εφαρμογών στην κάρτα, αφού έχει εκδοθεί, είναι το σημαντικότερο χαρακτηριστικό αυτής της τεχνολογίας. Λόγω της ανομοιομορφίας των δυνατοτήτων των έξυπνων καρτών, αλλά και της ασυμφωνίας μεταξύ των κατασκευαστών για μία ασφαλή μεθοδολογία εγκατάστασης, η προδιαγραφή για τη Java Card δεν ορίζει τον ακριβή τρόπο εγκατάστασης. Έτσι κάθε κατασκευαστής μπορεί να υλοποιήσει όποια μέθοδο κρίνει καλύτερη και αποδοτικότερη, σύμφωνα πάντα με τις δυνατότητες της κάρτας που κατασκευάζει.

Αναγνωρίζοντας το συγκεκριμένο κενό, ο οργανισμός Visa παρουσίασε την προδιαγραφή για έξυπνες κάρτες Visa Open Platform [58], τον Απρίλιο του 1998. Η προδιαγραφή VOP ορίζει μία αρχιτεκτονική για τη διαχείριση εφαρμογών σε κάρτες πολλαπλών εφαρμογών. Αυτό σημαίνει ότι ορίζονται πρότυπα για την ασφαλή φόρτωση και εγκατάσταση εφαρμογών σε τέτοιες κάρτες. Δεδομένου ότι ούτε η SUN, ούτε το Java Card Forum όρισαν έως σήμερα ένα ανοικτό πρότυπο για την ασφαλή φόρτωση εφαρμογών, η δεύτερη έκδοση του VOP, το Open Platform 2.0, έχει γίνει το de facto πρότυπο στην αγορά· είναι τόσο σημαντικό, ώστε οι κατασκευαστές υλοποιούν την πλατφόρμα Java Card και το Visa OP 2.0 στο ίδιο ολοκληρωμένο. Ο συνδυασμός της Java Card με το Visa OP ήταν ο κύριος λόγος για τον οποίο ο οργανισμός ETSI προτίμησε την πλατφόρμα αυτή για να ενσωματωθεί στο SIM API, μία επέκταση των προδιαγραφών της κάρτας SIM που χρησιμοποιείται στο σύστημα κινητής τηλεφωνίας GSM [56].

## 5.2 Θέματα ασφάλειας στη Java Card

Η πλατφόρμα Java Card σε συνδυασμό με το Visa OP παρέχει μία ασφαλή υποδομή για τη λειτουργία έξυπνων καρτών πολλαπλών εφαρμογών. Ένα κρίσιμο σημείο, η λογική εγκυρότητα της εικονικής μηχανής έχει επιβεβαιωθεί με τυπικές μεθόδους [33] [44]. Οι τυπικές μέθοδοι (formal methods) χρησιμοποιούν μαθηματικές αποδείξεις για να επιβεβαιώσουν μία υλοποίηση είναι μία έγκυρη μετάφραση μίας προδιαγραφής.

Ένα τρωτό σημείο στην πλατφόρμα είναι η ύπαρξη μεθόδων native. Τα πιθανά τρωτά σημεία αυτή τη στιγμή είναι δύο: η υλοποίηση του Java Card API και ο διαμοιρασμός αντικειμένων (object sharing). Εξετάζουμε στη συνέχεια αυτά τα θέματα.

### 5.2.1 Native methods

Οι μέθοδοι native είναι ένας μηχανισμός εκτέλεσης εντολών του λειτουργικού συστήματος μέσα από τη γλώσσα Java. Στην παραδοσιακή αρχιτεκτονική της Java ένα bytecode διερμηνεύεται από την εικονική μηχανή, η οποία στη συνέχεια εκτελεί τις αντίστοιχες κλήσεις λειτουργικού συστήματος. Με τον τρόπο αυτό διασφαλίζεται η ανεξαρτησία και η μεταφερσιμότητα μίας εφαρμογής Java από πλατφόρμα σε πλατφόρμα. Για διάφορους λόγους έχει επιβληθεί η ύπαρξη native methods στη γλώσσα Java. Με τον τρόπο αυτό, ένα bytecode μπορεί να εκτελέσει εντολές του λειτουργικού συστήματος, χωρίς τη μεσολάβηση της εικονικής μηχανής και του περιβάλλοντος εκτέλεσης.

Λόγω της ανομοιομορφίας των χαρακτηριστικών των έξυπνων καρτών, αλλά και των περιορισμένων πόρων που διαθέτουν, οι native method ενσωματώθηκαν στην πλατφόρμα Java Card. Με τον τρόπο αυτό ένα applet μπορεί να επιταχύνει σημαντικά τα χρονοβόρα τμήματα του κώδικά του, εκμεταλλευόμενο τα ιδιαίτερα χαρακτηριστικά της συγκεκριμένης κάρτας. Ταυτόχρονα βέβαια θυσιάζεται η αρχή της μεταφερσιμότητας της εφαρμογής, καθώς οι κλήσεις που γίνονται αφορούν το συγκεκριμένο λειτουργικό σύστημα.

Η υποστήριξη native method στην πλατφόρμα Java Card δημιουργεί προβλήματα ασφαλείας, καθώς τα applet, στο συγκεκριμένο τμήμα όπου τα χρησιμοποιούν, δεν υφίστανται τον έλεγχο και τους περιορισμούς που επιβάλλει το JCRE. Κατά συνέπεια έχουν την ελευθερία να προβούν σε οποιαδήποτε ενέργεια επιθυμούν. Για παράδειγμα μπορούν να διαβάσουν οποιαδήποτε περιοχή της μνήμης θέλουν, ακόμη και άλλων εφαρμογών που βρίσκονται μέσα στην κάρτα. Όταν όμως μιλάμε για ανοικτά και διαλειτουργικά συστήματα, αυτό είναι καταστροφικό, καθώς ένα κακόβουλο applet μπορεί να διαβάσει οποιαδήποτε πληροφορία περιέχεται σε ένα άλλο applet. Οι πληροφορίες αυτές μπορεί να είναι ευαίσθητα προσωπικά ή εμπορικά δεδομένα του χρήστη ή ακόμη και τα κρυπτογραφικά κλειδιά που χρησιμοποιούνται από αυτό το applet.

### 5.2.2 Υλοποίηση του Java Card API

Ένα μέρος του Java Card API είναι η υλοποίηση των κρυπτογραφικών αλγορίθμων DES, RSA, SHA-1 και MD5. Η υλοποίηση αυτή είναι ευθύνη του κατασκευαστή της κάρτας. Είναι δυνατό να υπάρχουν σχεδιαστικές ατέλειες ή αμελείς υλοποιήσεις αυτών των αλγορίθμων. Ωστόσο μία τέτοια απειλή θα αφορά ένα συγκεκριμένο μοντέλο έξυπνης κάρτας. Αν υπάρχουν τέτοια σημεία, η ασφάλεια όλων των εφαρμογών είναι αμφισβητήσιμη, καθώς οι κρυπτογραφικές υπηρεσίες παρέχονται ως υποδομή σε αυτές.

Ένα σημαντικό ζήτημα επίσης είναι η υλοποίηση πολλαπλών σχημάτων ασφαλείας χρησιμοποιώντας κοινά κλειδιά. Όπως έχει δειχθεί στο [28], είναι δυνατόν να εκτελέσουμε δύο ασφαλή πρωτόκολλα με τέτοιο τρόπο, ώστε το ένα να καταρρίπτει την ασφάλεια του άλλου, αν χρησιμοποιούνται κοινά κλειδιά. Το ζήτημα αυτό είναι δυνατό να υφίσταται ιδιαίτερα σε συστήματα με έξυπνες κάρτες πολλαπλών εφαρμογών, καθώς υπάρχουν περιορισμοί στο μέγιστο αριθμό κλειδιών που μπορεί να αποθηκεύσει μία κάρτα. Επίσης αρκετές εφαρμογές προτιμούν, για μεγαλύτερη ασφάλεια, να χρησιμοποιούν μία υπάρχουσα υποδομή κρυπτογραφίας, αντί να αναπτύσσουν μία δική τους. Με τον τρόπο αυτό όμως θα χρησιμοποιούνται κοινά κλειδιά από πολλές εφαρμογές και μάλιστα διαφορετικών κατασκευαστών. Κατά συνέπεια είναι εφικτό το σενάριο επίθεσης, όπου ένας κατασκευαστής εφαρμογών, χρησιμοποιεί την εφαρμογή του για να εκτελέσει ένα κρυπτογραφικό πρωτόκολλο επιτιθέμενος στο πρωτόκολλο ενός άλλου κατασκευαστή.

### 5.2.3 Διανομή αντικειμένων

Η πλατφόρμα Java Card 2.1 [54] παρέχει τη δυνατότητα διανομής αντικειμένων (object sharing), με το μηχανισμό Shareable Object Interface, SIO. Ο μηχανισμός αυτός κρίθηκε αναγκαίος, ώστε να μπορούν να απεικονίζονται στα applet μίας έξυπνης κάρτας εμπορικές σχέσεις μεταξύ διαφορετικών εταιρειών και για τη συνεργασία διαφορετικών applet της ίδιας εταιρείας. Ένα παράδειγμα της χρησιμότητας αυτής είναι το παρακάτω, το οποίο θα χρησιμοποιήσουμε και στη συνέχεια.

Ένα πιστωτικό ίδρυμα έχει εκδώσει με τη μορφή applet ένα ηλεκτρονικό πορτοφόλι ή μία πιστωτική κάρτα. Μία αεροπορική εταιρεία έχει εκδώσει ένα applet ηλεκτρονικών εισιτηρίων, το οποίο ενσωματώνει ένα σχήμα πίστης (loyalty scheme) και ένα σχήμα «συχνών πτήσεων» (frequent-flying program). Όλα αυτά τα applet είναι εγκατεστημένα σε μία κάρτα ενός πελάτη. Στα πλαίσια μίας εμπορικής συμφωνίας μεταξύ του πιστωτικού οργανισμού και της αεροπορικής εταιρείας, κάθε φορά που ένας πελάτης χρησιμοποιεί την πιστωτική κάρτα ή το ηλεκτρονικό πορτοφόλι για να αγοράσει ένα εισιτήριο της αεροπορικής εταιρείας, κερδίζει «μονάδες πίστης» (loyalty units).

Για να υλοποιηθεί αυτή η λειτουργία σε μία Java Card, το ηλεκτρονικό πορτοφόλι καλεί ένα SIO του applet της αεροπορικής εταιρείας, μέσω του οποίου αυξάνει τις μονάδες πίστης σε εκείνο το applet. Αυτό είναι απαραίτητο, καθώς αν το ηλεκτρονικό πορτοφόλι προσπαθούσε να καλέσει απευθείας τις μεθόδους του applet ηλεκτρονικών εισιτηρίων θα αποτύγχανε, γιατί το JCRE θα διαπίστωνε ότι δεν επιτρεπόταν η συγκεκριμένη κλήση, αφού δεν εμπίπτει στην περίμετρο ασφαλείας του.

Ο μηχανισμός διανομής αντικειμένων είναι ουσιώδης για την υλοποίηση ανοικτών, διαλειτουργικών συστημάτων με έξυπνες κάρτες. Ωστόσο, παρά τη χρησιμότητα του, εντοπίζουμε τέσσερα προβλήματα ασφαλείας. Τα προβλήματα αυτά είναι σημαντικά, καθώς δεν υπάρχουν αυτή τη στιγμή ικανοποιητικά αντίμετρα. Έτσι, οι εκδότες applet για να επιτύχουν το επιθυμητό επίπεδο ασφάλειας, είναι υποχρεωμένοι να δημιουργούν σχέσεις εμπιστοσύνης με τους εκδότες που συνεργάζονται και να εκδίδουν κάρτες με συγκεκριμένες εφαρμογές. Παρουσιάζουμε στη συνέχεια τα τέσσερα προβλήματα και τις προτεινόμενες λύσεις. Στα παραδείγματα, η εφαρμογή Α είναι το applet του πιστωτικού οργανισμού και η εφαρμογή Β είναι το applet της αεροπορικής εταιρείας. Για όπου χρειαστεί, η εφαρμογή Γ θα είναι ένα τρίτο applet, εγκατεστημένο μέσα στην ίδια κάρτα.

### Αναδιανομή των αντικειμένων

Όταν η εφαρμογή Α έχει αποκτήσει ένα SIO της εφαρμογής Β, τότε έχει πρόσβαση στις μεθόδους και τα πεδία της. Αυτό σημαίνει ότι μπορεί να αναδιανείμει τις πληροφορίες που αποκτά από τη Β, σε οποιαδήποτε άλλη εφαρμογή Γ. Δημιουργείται έτσι ένα πρόβλημα ασφάλειας, καθώς η εφαρμογή Β δε συμφωνεί απαραίτητα με τη δημιουργία αυτής της σχέσης. Ωστόσο δε μπορεί να το αντιμετωπίσει· δεν υπάρχει κάποιος τρόπος ώστε να ορίσει ότι η εφαρμογή Α δε μπορεί να μοιράσει στη συνέχεια τις πληροφορίες που αποκτά.

Για να αντιμετωπιστεί το πρόβλημα αυτό απαιτούνται αλλαγές στο λειτουργικό σύστημα της κάρτας ή στους μηχανισμούς του περιβάλλοντος εκτέλεσης (JCRE). Θα πρέπει δηλαδή να οριστούν μηχανισμοί ελέγχου ροής της πληροφορίας, έτσι ώστε να μη μπορεί η εφαρμογή Α να δίνει τις πληροφορίες που αποκτά από το Β στην εφαρμογή Γ. Ωστόσο ένα τέτοιο σχήμα δεν είναι αρκετό, καθώς θα πρέπει να εξεταστεί στη συνέχεια το πώς θα ενσωματωθεί αυτή η λογική στο συνολικό σύστημα όπου συμμετέχει η έξυπνη κάρτα. Θα πρέπει να ελέγχεται και η ροή της πληροφορίας μεταξύ του applet A και του off-card τμήματος της εφαρμογής A.

### Πρόσβαση σε όλες τις μεθόδους διεπαφής μίας κλάσης

Έστω ότι η εφαρμογή B έχει ορίσει δύο μεθόδους διεπαφής, οι οποίες είναι διαθέσιμες μέσω ενός SIO. Δεν είναι δυνατό να ορίσουμε διαφορετικά δικαιώματα πρόσβασης σε αυτές τις μεθόδους για διαφορετικές εφαρμογές. Για παράδειγμα, έστω ότι η A πρέπει να αλληλεπιδρά με τη μέθοδο του σχήματος πίστης και η Γ με τη μέθοδο του σχήματος συχνών πτήσεων. Αν η A τώρα γνωρίζει την ύπαρξη της μεθόδου για τη Γ, τότε δεν υπάρχει τρόπος να της απαγορευθεί η κλήση. Ο μόνος περιορισμός που μπορεί να επιβάλει το JCRE είναι η πρόσβαση στις μεθόδους της B μέσω ενός SIO, το οποίο είναι ακριβώς αυτό που θα κάνει η A.

Για να λυθεί αυτό το πρόβλημα, κάθε applet θα πρέπει να ελέγχει το αναγνωριστικό AID της καλούσας εφαρμογής και να πράττει ανάλογα. Όπως θα δούμε παρακάτω, αυτό ενδέχεται να δημιουργήσει διαχειριστικά προβλήματα.

### Πλαστοπροσωπία του AID

Η απόφαση για την πρόσβαση μέσω ενός SIO στις μεθόδους ενός αντικειμένου του B μπορεί να βασιστεί το πολύ στο AID της καλούσας εφαρμογής A, αφού αυτό είναι η μόνη διαθέσιμη πληροφορία. Ωστόσο είναι εφικτό να τεθεί το AID ενός κακόβουλου applet ίδιο με αυτό της A, που είναι γνωστό ότι έχει πρόσβαση σε συγκεκριμένες μεθόδους της B. Το κακόβουλο applet μπορεί στη συνέχεια να εγκατασταθεί στην κάρτα αντί της A. Έτσι μπορεί ελεύθερα να χρησιμοποιήσει το SIO για κακόβουλες πράξεις, για παράδειγμα για να αποσπάσει προσωπικές ή επιχειρηματικές πληροφορίες από το B.

Το πρόβλημα αυτό είναι πολύ σημαντικό. Οι λύσεις που προτείνονται αυτή τη στιγμή απαιτούν περιορισμούς στον τρόπο εγκατάστασης των applet, όπως για παράδειγμα να επιτρέπεται η εγκατάσταση υπογεγραμμένου κώδικα από έμπιστες πηγές. Ωστόσο αυτό περιορίζει δραματικά την ευελιξία της πλατφόρμας Java Card. Για παράδειγμα μία τέτοια λύση απαγορεύει στον κάτοχο της κάρτας να εγκαταστήσει νέες εφαρμογές για δικό του όφελος.

### Μελλοντικές αναφορές σε διανεμητέα αντικείμενα

Η διανομή αντικείμενων προϋποθέτει ότι το applet-εξυπηρέτης γνωρίζει εκ των προτέρων όλα τα AID των applet-πελατών, με τα οποία θα μοιράζεται συγκεκριμένες διεπαφές. Αυτή είναι μία πολύ ισχυρή απαίτηση για οποιοδήποτε εξυπηρέτη. Είναι λογικό να υποθέσουμε ότι καθώς νέες υπηρεσίες αναπτύσσονται και νέες συμφωνίες επισυνάπτονται, νέοι πελάτες θα απαιτούν να έχουν πρόσβαση στις διεπαφές ενός εξυπηρέτη. Αντίστοιχα το πρόβλημα υφίσταται όταν μία συμφωνία ακυρωθεί ή λήξει χωρίς να ανανεωθεί. Στην υπάρχουσα πλατφόρμα, και στις δύο περιπτώσεις, κάτι τέτοιο απαιτεί να ξαναγραφτεί και να επανεκδοθεί το applet-εξυπηρέτης, ώστε να περιέχει τη νέα λίστα με τα έγκυρα AID. Αυτό μπορεί να είναι ένα ανυπέρβλητο εμπόδιο όταν το applet-εξυπηρέτης είναι ευρέως διανεμημένο, όπως θα συνέβαινε σε ένα σύστημα μεγάλης κλίμακας.

### 5.2.4 Μία βελτίωση – δύο προβλήματα

Από τα προηγούμενα είναι φανερό ότι η δυνατότητα για διανομή αντικειμένων δημιουργεί προβλήματα ασφάλειας, τα οποία είναι πολύ σημαντικά ώστε να αγνοηθούν προς χάριν της ευκολίας που προσφέρουν για ένα ανοικτό και διαλειτουργικό σύστημα. Στο [41] προτείνεται η χρήση αντιπροσώπων (delegate) για τη διαχείριση των SIO. Κάθε εξυπηρέτης ορίζει έναν αντιπρόσωπο για κάθε μέθοδο που επιθυμεί να διανείμει. Ο αντιπρόσωπος είναι μία διαφορετική κλάση, η οποία έχει το ίδιο AID με τον εξυπηρέτη. Ο αντιπρόσωπος ταυτοποιεί την καλούσα κλάση, χρησιμοποιώντας ένα πρωτόκολλο πρόκλησης-απάντησης, το οποίο βασίζεται στη γνώση ενός κοινού μυστικού με την καλούσα κλάση. Η πρόσβαση στις μεθόδους του αντιπροσώπου επιτρέπεται μόνο μετά την επιτυχή ταυτοποίηση.

Το προτεινόμενο σχήμα λύνει μόνο δύο από τα τέσσερα προβλήματα που θέσαμε: την πρόσβαση σε όλες τις μεθόδους μίας κλάσης και την πλαστοπροσωπία του AID. Όπως ήδη έχουμε δει το πρόβλημα της αναδιανομής των αντικειμένων αντιμετωπίζεται μόνο με αλλαγές στο JCRE. Το πρόβλημα των μελλοντικών αναφορών στα αντικείμενα αντιμετωπίζεται μερικώς. Πιο συγκεκριμένα δε χρειάζεται να αναβαθμίζεται η κλάση κάθε φορά που αλλάζει μία σχέση μεταξύ δύο εφαρμογών, παρά μόνο οι αντίστοιχοι αντιπρόσωποι.

Το συγκεκριμένο σχήμα που προτείνεται, το οποίο και είναι και το μοναδικό μέχρι τώρα στη σχετική βιβλιογραφία, έχει δύο σημαντικά προβλήματα ασφαλείας. Το πρώτο πηγάζει από το γεγονός ότι η ταυτοποίηση γίνεται από τον αντιπρόσωπο μίας κλάσης και όχι από την ίδια την κλάση. Με τον τρόπο αυτό, η κλάση δε μπορεί να γνωρίζει ότι ο αντιπρόσωπός της είναι αυθεντικός. Η μόνη ταυτοποίηση που χρησιμοποιείται είναι το AID του αντιπροσώπου, το οποίο θα πρέπει να είναι ίδιο με το AID της κλάσης που αντιπροσωπεύει. Όπως είδαμε στην υποενότητα 5.2.3, θα μπορούσαμε να κατασκευάσουμε έναν τέτοιο αντιπρόσωπο, τον οποίο θα εγκαθιστούμε στην κάρτα αντί του κανονικού. Ο αντιπρόσωπος αυτός μπορεί να παρακάμψει τον έλεγχο αυθεντικότητας και να δίνει ελεύθερη πρόσβαση στη συγκεκριμένη μέθοδο που χρησιμοποιείται.

Η χρήση του AID είναι η πηγή και του δεύτερου προβλήματος. Πιο συγκεκριμένα το AID είναι μέρος του μυστικού που γνωρίζει ο αντιπρόσωπος και η καλούσα εφαρμογή. Αυτό συμβαίνει γιατί αντιπροσωπεύεται η μέθοδος και όχι η σχέση μεταξύ δύο κλάσεων. Έτσι, αν για κάποιο λόγο μία σχέση πάψει να υπάρχει, αυτό που συμβαίνει είναι ότι αφαιρείται το αντίστοιχο AID από τα διαθέσιμα, τα οποία μπορούν να χρησιμοποιηθούν στο σχήμα ταυτοποίησης.

Τίποτα όμως δε μπορεί να εμποδίσει τη φόρτωση μίας κλάσης με πλαστό αλλά έγκυρο AID. Σύμφωνα με το σχήμα ταυτοποίησης που προτείνεται, η κλάση αυτή θα μπορέσει να αποκτήσει πρόσβαση στις συγκεκριμένες μεθόδους.

Από τα παραπάνω επιβεβαιώνεται για μία ακόμη φορά ότι η ασφάλεια ενός συστήματος είναι μία αλυσίδα· το επίπεδο της ασφάλειας καθορίζεται από τον πιο αδύναμο κρίκο της αλυσίδας. Στην περίπτωση μας, ο κρίκος αυτός είναι το AID. Από τη στιγμή που συνεχίζει να χρησιμοποιείται αυτούσιο, συνεχίζουν να υφίστανται προβλήματα ασφαλείας, τα οποία βασίζονται σε αυτό.

## 5.3 Ασφαλής διανομή αντικειμένων

Στην ενότητα αυτή παρουσιάζουμε μία απλή αρχιτεκτονική διανομής αντικειμένων για την πλατφόρμα Java Card. Η αρχιτεκτονική αυτή δεν απαιτεί αλλαγές στο Java Card API, όπως στο [41]. Όπως θα αποδείξουμε, η αρχιτεκτονική μας αντιμετωπίζει επιτυχώς τα δύο προβλήματα που αναφέραμε παραπάνω. Επιπλέον, αν το επιτρέπει η υποδομή της κάρτας, μπορεί να αντιμετωπίσει εγγενώς, ως ένα σημείο, το πρόβλημα των μελλοντικών αναφορών σε διανεμόμενα αντικείμενα.

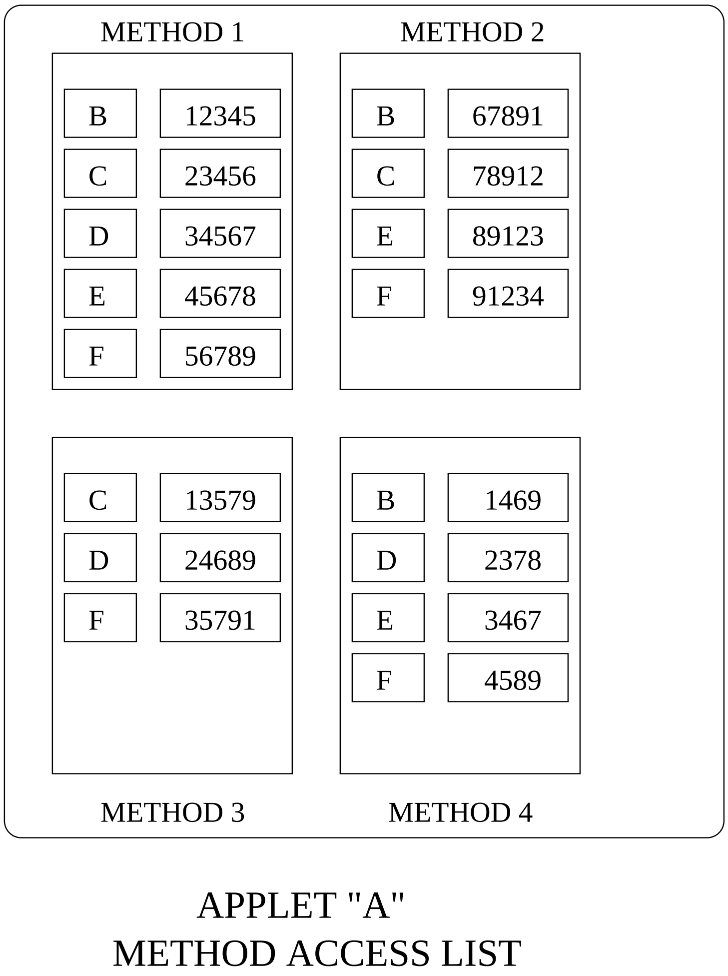
### 5.3.1 Περιγραφή της αρχιτεκτονικής

Στην αρχιτεκτονική μας υπάρχουν δύο applet, το A και B σε ρόλο εξυπηρέτη και πελάτη αντίστοιχα. Ο εξυπηρέτης είναι το applet, το οποίο παρέχει διανεμόμενες διεπαφές και ο πελάτης είναι το applet που επιθυμεί να τις χρησιμοποιήσει. Δεν απαγορεύεται ένα applet να έχει και τους δύο ρόλους, αλλά σε κάθε συνεδρία μπορεί να έχει μόνο έναν από τους δύο.

Το applet A έχει στην κατοχή του μία λίστα για κάθε διανεμόμενη διεπαφή με ένα AID πελάτη και ένα μυστικό κλειδί, το οποίο αντιστοιχεί στο συγκεκριμένο πελάτη και στη συγκεκριμένη διεπαφή, όπως φαίνεται στο σχήμα 5.1. Το μυστικό κλειδί χρησιμοποιείται σε ένα σχήμα ταυτοποίησης μηδενικής γνώσης, όπως για παράδειγμα το σχήμα Fiat-Shamir, το οποίο παρουσιάσαμε στο κεφάλαιο 4. Για να αποκτήσει πρόσβαση στη διεπαφή του A, οι δύο κλάσεις εκτελούν το σχήμα ταυτοποίησης ώστε το B να αποδείξει στο A την ταυτότητά του. Αν το πρωτόκολλο εκτελεστεί σωστά, τότε επιτρέπεται η πρόσβαση στη διεπαφή, αλλιώς η αίτηση απορρίπτεται. Φυσικά, η αίτηση απορρίπτεται από την αρχή, αν το B δεν είναι στη λίστα διανομής της συγκεκριμένης διεπαφής του A.

### 5.3.2 Απόδειξη ασφάλειας

Η αρχιτεκτονική που προτείνουμε, αντιμετωπίζει επιτυχώς τα δύο προβλήματα του σχήματος της ενότητας 5.2.4. Το εκάστοτε κλειδί αντιστοιχεί σε ζεύγη πελάτη-διεπαφής· κάθε διαφορετικό ζεύγος έχει διαφορετικό κλειδί. Με τον τρόπο αυτό αν ένα applet αποκτά πρόσβαση σε μία μέθοδο, δε μπορεί να




Σχήμα 5.1: Αρχιτεκτονική διανομής αντικειμένων

αποκτήσει πρόσβαση και σε μία άλλη, αν δεν ταυτοποιηθεί με βάση το νέο κλειδί. Επίσης η πλαστοπροσωπία του AID δεν υφίσταται. Αν μία κλάση δηλώσει πλαστό AID, τότε δε θα μπορέσει να εκτελέσει με επιτυχία το σχήμα ταυτοποίησης, καθώς δε θα γνωρίζει τα σωστά κλειδιά. Από την άλλη πλευρά, αν μία κλάση εγκατασταθεί σε μία κάρτα ως A, τότε θα μπορεί να ανακτήσει τα μυστικά κλειδιά της B, αφού το σχήμα είναι μηδενικής γνώσης· δε θα μάθει κάτι παραπάνω από όσα ήξερε μέχρι εκείνη τη στιγμή.

### 5.3.3 Παρατηρήσεις – Επεκτάσεις

Η αρχιτεκτονική που προτείνουμε έχει το πρόβλημα των μελλοντικών αναφορών στα αντικείμενα. Οποιαδήποτε αλλαγή στις σχέσεις μεταξύ δύο applet, συνεπάγεται την ανανέωση του κώδικα του A. Μία κάρτα, η οποία είναι απομονωμένη από το συνολικό σύστημα, δε μπορεί να ενημερωθεί για τις νέες εξελίξεις μεταξύ των σχέσεων. Αν όμως συμμετάσχει στο σύστημα, τότε μπορεί να ενημερωθεί άμεσα· σε αυτήν την περίπτωση, από πλευράς ευχρηστίας είναι προτιμότερο να ανανεώνεται μία μόνο εφαρμογή με όλες τις αλλαγές, παρά να απαιτείται η εγκατάσταση πολλών αντιπροσώπων.

Η κατάσταση αυτή μπορεί να βελτιωθεί σε κάρτες που διαθέτουν ρολόι πραγματικού χρόνου, όπως για παράδειγμα το iButton της εταιρείας Dallas Semiconductors [29]. Σε αυτή την περίπτωση εισάγουμε ένα τρίτο πεδίο σε κάθε εγγραφή, το οποίο δηλώνει την ημερομηνία λήξης της σχέσης. Έτσι το A μπορεί να ελέγχει αυτό το πεδίο, ώστε, ακόμη και αν δεν έχει ανανεωθεί πρόσφατα, προληπτικά να απαγορεύει την πρόσβαση στις μεθόδους του. Το πεδίο αυτό μπορεί να υπάρχει και στις κάρτες χωρίς τέτοιο ρολόι, απλά θα πρέπει να λαμβάνεται ειδική μέριμνα, ώστε να αποθηκεύεται μία αδιάφορη τιμή σε αυτό.

## 5.4 Σύνοψη

Στο κεφάλαιο αυτό ασχοληθήκαμε με την ασφάλεια έξυπνων καρτών Java Card. Οι επιθέσεις πλάγιου καναλιού που παρουσιάσαμε στα προηγούμενα κεφάλαια είναι εφαρμόσιμες και σε αυτές. Η δυνατότητα ανανέωσης του κώδικα περιορίζει την μακροπρόθεσμη επιτυχία τέτοιων επιθέσεων σε αυτές τις κάρτες. Η πλατφόρμα Java Card ωστόσο έχει ενσωματώσει χάριν ευκολίας ορισμένους μηχανισμούς, οι οποίοι με τους κατάλληλους χειρισμούς μπορούν να οδηγήσουν σε επιτυχείς επιθέσεις. Παρουσιάσαμε τις γνωστές επιθέσεις καθώς και νέες που εντοπίσαμε στα πλαίσια της εργασίας μας. Παράλληλα δείξαμε ότι η μόνη αρχιτεκτονική που έχει προταθεί αυτή τη στιγμή στη βιβλιογραφία λύνει μόνο μερικά από αυτά τα προβλήματα και όχι πάντα με επιτυχία. Τέλος προτείναμε μία νέα αρχιτεκτονική, η οποία αντιμετωπίζει επιτυχώς αυτά τα προβλήματα και προτείναμε επεκτάσεις που βασίζονται σε συγκεκριμένα χαρακτηριστικά κάποιων μοντέλων.

# Κεφάλαιο 6

# Επίλογος

*«Στο τέλος της ανηφόρας υπάρχει η κούραση, που συνοδεύεται από το αδύνατο που έγινε δυνατό. Στην κατηφόρα κουράζεσαι για κάτι που ο καθένας μας θα μπορούσε να το κάνει.»*

—*Αριστομένης Προβελέγγιος*

## 6.1 Συμπεράσματα

Στην εργασία αυτή εστιάσαμε στη λογική ασφάλεια που προσφέρουν συστήματα έξυπνων καρτών, σε περιβάλλοντα όπου μπορούν να υλοποιούνται επιθέσεις πλάγιου καναλιού.

Για τα συστήματα καρτών μνήμης με μονάδα αυθεντικότητας, προτείναμε ένα θεωρητικό μοντέλο κατανάλωσης ισχύος, το οποίο ενσωματώνει όλη τη διαθέσιμη γνώση για τη λειτουργία της μονάδας. Με τον τρόπο αυτό περιγράφουμε όσο το δυνατό πληρέστερα τις μεταβολές της κατανάλωσης ισχύος κάθε χρονική στιγμή. Επιπλέον αναπτύξαμε μία ανοικτή και επεκτάσιμη βιβλιοθήκη επικοινωνίας με κάρτες μνήμης με χρήση αναγνώστη έξυπνων καρτών. Με τον τρόπο αυτό παρέχουμε μία κοινή υποδομή για την υλοποίηση παθητικών επιθέσεων σε όλες τις οικογένειες έξυπνων καρτών.

Για τα συστήματα έξυπνων καρτών που χρησιμοποιούν το πρωτόκολλο ταυτοποίησης Fiat-Shamir, αποδείξαμε ότι το μοντέλο ενεργών επιθέσεων της Bellcore δεν είναι πάντα έγκυρο. Στη συνέχεια επεκτείναμε το μοντέλο επιθέσεων, ώστε να είναι έγκυρο σε κάθε διαμόρφωση συστήματος. Παρουσιάσαμε επίσης προβλήματα υλοποίησης των επιθέσεων αυτών στις συγκεκριμένες διαμορφώσεις. Τέλος, προτείναμε ένα εναλλακτικό μοντέλο επιθέσεων, το οποίο είναι έγκυρο σε κάθε διαμόρφωση συστήματος. Το μοντέλο αυτό, αν και βασίζεται σε ισχυρότερες υποθέσεις, είναι ελκυστικό, καθώς παρακάμπτει τα προβλήματα υλοποίησης του προηγούμενου μοντέλου.

Για τα συστήματα έξυπνων καρτών που βασίζονται στην τεχνολογία Java Card, παρουσιάσαμε μία σειρά επιθέσεων που άρουν την ασφάλεια των δεδομένων. Οι επιθέσεις χρησιμοποιούν το μηχανισμό διαμοιρασμού αντικειμένων μεταξύ των εφαρμογών της έξυπνης κάρτας. Στη συνέχεια εξετάσαμε ένα σχήμα που έχει προταθεί στη βιβλιογραφία για την αποφυγή μερικών επιθέσεων. Αποδείξαμε ότι και σε αυτήν την περίπτωση μπορούμε με τον ίδιο μηχανισμό να παρακάμψουμε το σχήμα ασφαλείας. Σε κάθε περίπτωση λοιπόν ο διαμοιρασμός αντικειμένων είναι ένας επισφαλής μηχανισμός και θα πρέπει να χρησιμοποιείται με ιδιαίτερη προσοχή σε ανοικτά και διαλειτουργικά περιβάλλοντα.

Η εργασία μας επιβεβαίωσε ότι η ασφάλεια ενός συστήματος είναι μία αλυσίδα· ο πιο αδύναμος κρίκος της αλυσίδας είναι αυτός που καθορίζει και το συνολικό επίπεδο ασφάλειας που προσφέρει το σύστημα. Η προσθήκη νέων χαρακτηριστικών και λειτουργιών σε ένα σύστημα θα πρέπει να γίνεται με ιδιαίτερη προσοχή και σε κάθε περίπτωση θα πρέπει να επανεξετάζεται η συνολική ασφάλεια του συστήματος. Η επανεξέταση αυτή θα πρέπει να γίνεται σε κάθε περίπτωση, ακόμη και όταν σε πρώτη ματιά δε σχετίζονται άμεσα με την ασφάλεια του συστήματος.

## 6.2 Μελλοντικές κατευθύνσεις

Στα πλαίσια της μεταπτυχιακής μας εργασίας, εμφανίστηκαν νέα ερωτήματα, τα οποία χρήζουν απάντησης. Οι απαντήσεις αυτές δε θα μπορούσαν να δοθούν στον περιορισμένο χρόνο και χώρο της εργασίας μας.

Στον τομέα των παθητικών επιθέσεων σε κάρτες μνήμης, θα είχε ιδιαίτερο ενδιαφέρον να γίνουν οι απαραίτητες μετρήσεις και να δοκιμασθεί η αξιοπιστία του μοντέλου ή και να τροποποιηθεί κατάλληλα, αν κριθεί σκόπιμο. Είναι επίσης ενδιαφέρον να εξετάσει κανείς πώς μπορούν να χρησιμοποιηθούν επιθέσεις βασισμένες σε κατανάλωση ισχύος, σε άλλα συστήματα ασφαλείας, πέραν των έξυπνων καρτών και των κρυπτογραφικών αλγορίθμων.

Στον τομέα των ενεργών επιθέσεων, είναι ενδιαφέρον να εξεταστεί η επέκταση τους σε άλλα πρωτόκολλα και αλγόριθμους κρυπτογραφίας. Για τα μοντέλα επιθέσεων που εξετάσαμε θα ήταν χρήσιμο να εντοπιστεί, αν υπάρχει, κάτω φράγμα πολυπλοκότητας στον αριθμό λαθών που απαιτείται για την ανακατασκευή των μυστικών κλειδιών.

# Βιβλιογραφία

[1] Mehdi Laurent Akkar, Règis Bevan, Paul Dischamp και Didier Moyart. Power Analysis, What in Now Possible .... Στο *Advances in Cryptology - ASIACRYPT 2000*, τόμος 1976 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 489–502. Springer, 2000.

[2] Ross J. Anderson και Markus G. Kuhn. Tamper Resistance – a Cautionary Note. Στο *The Second USENIX Workshop on Electronic Commerce Proceedings*, σελίδες 1–11, Oakland, California, 1996.

[3] Ross J. Anderson και Markus G. Kuhn. Low cost attacks on tamper resistant devices. Στο *Security Protocols, 5th International Workshop, Proceedings*, τόμος 1361 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 125–136. Springer, 1997.

[4] E.P. Antoniadis, D.N. Serpanos, A. Traganitis και A.G. Voyiatzis. Software Simulation of Active Attacks on Cryptographic Systems. Τεχνική Αναφορά υπ. αρίθμ. CSD-TR-2001-01, Department of Computer Science, University of Crete, Greece, 2001.

[5] Tuomas Aura και Dieter Gollmann. Software License Management with Smart Cards. Στο *Proceedings of the First USENIX Workshop on Smartcard Technology*, σελίδες 75–85, Chicago, Μάιος 1999. USENIX Association.

[6] F. Bao, R. Deng, Y. Han, A.D. Narasimhalu και T. Ngair. Breaking Public Key Cryptosystems on Tamper Resistant Devices in the Presence of Transient Faults. Στο *Security Protocol Workshop '97, LNCS 1361.* Springer-Verlag, 1997.

[7] Eli Biham και Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. Στο *Advances in Cryptology - CRYPTO'97*, τόμος 1294 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 513–525. Springer, 1997.

[8] Eli Biham και Adi Shamir. Power analysis of the key scheduling of the aes candidates. Στο *Second Advanced Encryption Standard (AES)*

*Candidate Conference*, Rome, Italy, Μάρτιος 1999. `http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm`.

[9] Dan Boneh. Twenty Years of Attack on the RSA Cryptosystem. *Notices of American Mathematical Society*, 46(2):203–213, 1999.

[10] Dan Boneh, Richard A. DeMillo και Richard J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. Στο *Advances in Cryptology - Eurocrypt'97*, τόμος 1233 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 37–51. Springer, 1997.

[11] Dan Boneh, Richard A. DeMillo και Richard J. Lipton. On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 14(2):101–119, 2001.

[12] B. Canvel και C.T.J. Dodson. Public Key Cryptosystem Timing Analysis: Evaluating Obscuring Techniques. Στο *Advances in Cryptology - CRYPTO 2000*, τόμος 1880 στο *Lecture Notes in Computer Science (LNCS)*. Springer, 2000. Rump Session Presentation.

[13] Suresh Chari, Charanjit Jutla, Josyula R. Rao και Pankaj Rohatgi. A cautionary note regarding evaluation of AES candidates on smart-cards. Στο *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, Μάρτιος 1999. `http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm`.

[14] Cristophe Clavier, Jean Sébastian Coron και Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. Στο *Cryptographic Hardware and Embedded Systems, Second International Workshop, CHES 2000*, τόμος 1965 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 252–263. Springer, 2000.

[15] Jean Sébastian Coron και Louis Goubin. On Boolean and Arithmetic Masking against Differential Power Analysis. Στο *Cryptographic Hardware and Embedded Systems, Second International Workshop, CHES 2000*, τόμος 1965 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 231–237. Springer, 2000.

[16] J. F. Dhem, F. Koeune, P. A. Leroux, Mestré, Jean Jaques Quisquater και J. L. Willems. A Practical Implementation of the Timing Attack. Τεχνική Αναφορά υπ. αρίθμ. CG-1998/1, UCL Crypto Group, DICE, Université Catholique de Louvain, Belgium, 1998.

[17] EMV Co. EMV standards, 1999. ΥΡΛ: `ηττπ://ωωω.εμςομ.ςομ/`.

[18] Uriel Feige, Amos Fiat και Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 1(2):77–94, 1988.

[19] Gemplus. *GemCcore-Based Reader Interface Library Programmer's Guide*, Μάιος 1997. Έκδοση 1.0.

[20] Gemplus. *GemCcore ChipSet Controller Software*, Φεβρουάριος 1998. Έκδοση 2.0.

[21] Gemplus. *Description of the GemCcore and OROS Command Interpreter*, Μάιος 1999. Έκδοση 1.0.

[22] Gemplus. *GemCore Technical Specifications*, Αύγουστος 2000. Έκδοση 2.0.

[23] H. Handschuh, P. Paillier και J. Stern. Probing Attacks on Tamper-Resistant Devices. Στο *Cryptographic Hardware and Embedded Systems, First International Workshop,CHES'99*, τόμος 1717 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 303–315. Springer, 1999.

[24] Java Card Forum. `http://www.javacardforum.org/`, 2001.

[25] Marc Joye και Jean Jaques Quisquater. Attacks on systems using chinese remaindering. Τεχνική Αναφορά υπ. αρίθμ. CG-1996/9, Universite Catholique de Louvain, 1996.

[26] Marc Joye και Jean Jaques Quisquater. Faulty RSA Encryption. Τεχνική Αναφορά υπ. αρίθμ. CG-1997/8, Universite Catholique de Louvain, 1997.

[27] J. Kelsey, B. Schneier, D. Wagner και C. Hall. Side Channel Cryptanalysis of Product Ciphers. Τεχνική Αναφορά υπ. αρίθμ., Counterpane Systems, 1998. διαθέσιμη από το URL: `http://www.counterpane.com/`.

[28] John Kelsey, Bruce Schneier και David Wagner. Protocol Interactions and the Chosen Protocol Attack. Στο *Security Protocols, 5th International Workshop, Proceedings*, τόμος 1361 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 91–104. Springer, 1997.

[29] Kingpin. A practical introduction to the Dallas Semiconductor iButton, χ.χ. Research Report. Διαθέσιμο στο URL: `http://www.atstake.com/research/reports/practical_introduction_to_ibutton.pdf`.

[30] P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems. Στο *Advances in Cryptology - CRYPTO '96*, τόμος 1109 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 104–113. Springer, 1996.

[31] Paul C. Kocher, Joshua Jaffe και Benjamin Jun. Differential Power Analysis. Στο *Advances in Cryptology - CRYPTO'99*, τόμος 1666 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 388–397. Springer, 1999.

[32] O. Kömmerling και M.G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. Στο *Proceedings of the First USENIX Workshop on Smartcard Technology*, σελίδες 9–20, Μάιος 1999.

[33] Jean Louis Lanet και Antoine Requet. Formal Proof of smart card applets correctness. Στο *Third Smart Card Research and Advanced Application Conference, CARDIS'98*. Springer, 1998.

[34] A.K. Lenstra. Memo on RSA Signature Generation in the Presense of Faults, 1996. Manuscript available from author, `arjen.lenstra@citicorp.com`.

[35] David P. Macher. Fault Induction Attacks, Tamper Resistance, and Hostile Reverse Engineering in Perspective. Στο *Financial Cryptography, FC '97 Proceedings*, επιμελητής: R. Hirschfeld, τόμος 1318 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 109–121. Springer, 1997.

[36] Maosco. `http://www.maosco.com/`, 2001.

[37] Ritta Mayer-Sommer. Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smartcards. Στο *Cryptographic Hardware and Embedded Systems, Second International Workshop, CHES 2000*, τόμος 1965 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 78–92. Springer, 2000.

[38] Alfred J. Menezes, Paul C.van Oorschot και Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press Series on Discrete Mathematics and its Applications. CRC Press, 1996.

[39] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. Στο *Cryptographic Hardware and Embedded Systems, Second International Workshop, CHES 2000*, τόμος 1965 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 238–251. Springer, 2000.

[40] T.S. Messerges, E.A. Dabbish και R.H. Sloan. Investigations of Power Analysis Attacks on Smartcards. Στο *Proceedings of the First USENIX Workshop on Smartcard Technology*, Μάιος 1999.

[41] M. Montgommery και K. Krishna. Secure Object Sharing in Java Card. Στο *Proceedings of the First USENIX Workshop on Smartcard Technology*, Μάιος 1999.

[42] OpenCard Consortium. `http://www.opencard.org/`, 2001.

[43] PC/SC Workgroup. `http://www.pcscworkgroup.com/`, 2001.

[44] Joachim Posegga και Harald Vogt. Byte Code Verification for Java Smart Card Based on Model Checking. Στο *ESORICS*, Lecture Notes in Computer Science (LNCS), σελίδες 175–190. Springer, 1998.

[45] Wolfgang Rankl και Wolfgang Effing. *Smart Card Handbook.* John Willey and Sons, 2η έκδοση, 2000. ISBN: 0471988758.

[46] Ron Rivest, Adi Shamir και Leonard Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[47] Wener Schindler. A Timing Attack against RSA with the Chinese Remainder Theorem. Στο *Cryptographic Hardware and Embedded Systems, Second International Workshop, CHES 2000*, τόμος 1965 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 93–108. Springer, 2000.

[48] Bruce Schneier. *Applied Cryptography.* John Wiley and Sons, Inc., 2η έκδοση, 1996.

[49] Adi Shamir. Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies. Στο *Cryptographic Hardware and Embedded Systems, Second International Workshop, CHES 2000*, τόμος 1965 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 71–77. Springer, 2000.

[50] Claude Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423,623–565, Θυλψ, Οςτοβερ 1948.

[51] Peter Smulders. The Threat of Information Theft by Reception of Electromagnetic Radiation from RS-232 Cables. *Computers & Security*, 9:53–58, 1990.

[52] SUN Microsystems, Inc. Java Card 2.1.1 Runtime Environment Specification. `http://java.sun.com/products/javacard/javacard21.html`, χ.χ.

[53] SUN Microsystems, Inc. Java Card 2.1.1 Virtual Machine Specification. `http://java.sun.com/products/javacard/javacard21.html`, χ.χ.

[54] SUN Microsystems, Inc. Java Card API 2.1.1 Specification. `http://java.sun.com/products/javacard/javacard21.html`, χ.χ.

[55] SUN Microsystems, Inc. SUN microsystems, inc. announces java card API. Press Release, Οκτώβριος 1996. `http://java.sun.com/pr/1996/oct/pr961029-02.html`.

[56] SUN Microsystems, Inc. SUN Microsystems' java card technology available as GSM standard in mobile phones. Press Release, Νοέμβριος 1999. `http://java.sun.com/pr/1999/11/pr991116-02.html`.

[57] Wimvan Eck. Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk? *Computers & Security*, 4:269–286, 1985.

[58] Visa. Visa Open Platform. `http://www.visa.com/nt/suppliers/open/main.html`, 2001.

[59] Steve H. Weingart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defences. Στο *Cryptographic Hardware and Embedded Systems, Second International Workshop, CHES 2000*, τόμος 1965 στο *Lecture Notes in Computer Science (LNCS)*, σελίδες 302–317. Springer, 2000.

[60] Yuliang Zheng και Tsutomo Matsumoto. Breaking Smart Card Implementations of ElGamal Signature and Its Variants. 1996. Παρουσιάστηκε στο rump session του συνεδρίου AsiaCrypt '96.

Η ΠΑΡΟΥΣΑ ΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΟΘΕΤΗΘΗΚΕ ΗΛΕΚΤΡΟΝΙΚΑ. ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΤΥΠΟΓΡΑΦΙΑΣ, ΤΑ ΟΠΟΙΑ ΑΝΑΠΤΥΧΘΗΚΑΝ, ΒΕΛΤΙΩΘΗΚΑΝ Η ΕΜΠΛΟΥΤΙΣΤΗΚΑΝ ΑΠΟ ΦΟΙΤΗΤΕΣ ΤΟΥ ΤΜΗΜΑΤΟΣ.

ΤΟ ΜΕΓΑΛΥΤΕΡΟ ΜΕΡΟΣ ΤΟΥ ΚΕΙΜΕΝΟΥ ΣΥΝΤΑΧΘΗΚΕ ΣΤΟΝ ΚΕΙΜΕΝΟΓΡΑΦΟ VIMH (VI IMPROVED, HELLENIC). Η ΕΛΛΗΝΙΚΗ ΕΠΙΛΟΓΗ ΤΟΥ ΚΕΙΜΕΝΟΓΡΑΦΟΥ ΑΝΑΠΤΥΧΘΗΚΕ ΑΠΟ ΤΟΝ ΜΙΧΑΛΗ ΦΛΟΥΡΗ, ΟΝΤΑΣ ΜΕΤΑΠΤΥΧΙΑΚΟΣ ΦΟΙΤΗΤΗΣ ΤΟΥ ΤΜΗΜΑΤΟΣ.

Ο ΟΡΘΟΓΡΑΦΙΚΟΣ ΕΛΕΓΧΟΣ ΤΟΥ ΚΕΙΜΕΝΟΥ ΕΓΙΝΕ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ ISPELLH, ΤΟ ΟΠΟΙΟ ΑΠΟΤΕΛΕΣΕ ΘΕΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ.

Η ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΤΕΛΙΚΗΣ ΜΟΡΦΗΣ ΤΟΥ ΕΓΓΡΑΦΟΥ ΕΓΙΝΕ ΜΕ ΤΟ ΠΡΟΓΡΑΜΜΑ LaTeX $2_\varepsilon$, ΤΟ ΟΠΟΙΟ ΑΝΕΠΤΥΞΕ Ο LESLIE LAMPORT, ΣΤΗΡΙΖΟΜΕΝΟΣ ΣΤΗΝ ΕΡΓΑΣΙΑ ΤΟΥ DONALD E. KNUTH, ΤΟ TeX. ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ ΕΠΙΣΗΣ Η ΕΛΛΗΝΙΚΗ ΕΠΙΛΟΓΗ ΤΟΥ ΠΑΚΕΤΟΥ BABEL, ΤΗΝ ΟΠΟΙΑ ΑΝΕΠΤΥΞΕ Ο ΑΠΟΣΤΟΛΟΣ ΣΥΡΟΠΟΥΛΟΣ, ΓΙΑ ΤΗΝ ΟΡΘΗ ΑΠΟΔΟΣΗ ΤΟΥ ΚΕΙΜΕΝΟΥ ΚΑΤΑ ΤΟΥΣ ΚΑΝΟΝΕΣ ΤΗΣ ΕΛΛΗΝΙΚΗΣ ΤΥΠΟΓΡΑΦΙΑΣ.