ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
Επικοινωνιακά Συστήματα και Τεχνολογίες

Τυπική ανάλυση πρωτοκόλλων ασφαλών πληρωμών με Χρωματισμένα Δίκτυα Petri

Ονοματεπώνυμο: Ταρασιάδης Μιλτιάδης

ΑΕΜ : 79

Επιβλέπον: κ. Παναγιώτης Κατσαρός

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα της μεταπτυχιακής μου εργασίας κ. Παναγιώτη Κατσαρό Λέκτορα του τμήματος πληροφορικής Α.Π.Θ., για την πολύτιμη βοήθεια και την καθημερινή υποστήριξή του.

Επίσης, τον υπ. Διδάκτορα του τμήματος πληροφορικής Α.Π.Θ. κ. Μπασαγιάννη Στυλιανό, για την πολύ καλή συνεργασία.

Τέλος, ιδιαίτερα του γονείς μου για την συνεχή υποστήριξη καθ' όλη την διάρκεια των σπουδών μου.

1 Περιεχόμενα

Κεφάλαιο 2

Εισαγωγή στις πληρωμές μέσω κινητών συσκευών

2 Εισαγωγή στις πληρωμές μέσω κινητών συσκευών

Μια από τις πιο διαδεδομένες χρήσεις ηλεκτρονικών συσκευών στις μέρες μας, είναι η χρήση προσωπικών κινητών συσκευών, όπως είναι τα κινητά τηλέφωνα και τα νεοεισερχόμενα στην αγορά των τηλεπικοινωνιών, PDAs. Την αλματώδη εξέλιξη των συσκευών αυτών ακολουθεί η ανάπτυξη υπηρεσιών και εφαρμογών ειδικά προσαρμοσμένων σε συσκευές που χαρακτηρίζονται από ιδιαίτερα χαρακτηριστικά και απαιτήσεις.

Οι συσκευές αυτές είναι εξαιρετικά αποδοτικές σε λειτουργίες εξουσιοδότησης και διαχείρισης ηλεκτρονικών πληρωμών ακόμη και τραπεζικών συναλλαγών, προσφέροντας “ασφάλεια” αλλά πάνω από όλα άμεση και γρήγορη εξυπηρέτηση. Μπορούν να συγκριθούν ακόμη και με τις ευρέως διαδεδομένες online πληρωμές που πραγματοποιούνται με τη χρήση Η/Υ, μέσω internet.

Μερικά από τα πλεονεκτήματα που μπορεί ένας χρήστης να αποκομίσει από τις υπηρεσίες ηλεκτρονικών πληρωμών των παραπάνω συσκευών, έχουν είδη ενσωματωθεί στις νέες εξελιγμένες προσωπικές κινητές συσκευές, κάποιες από αυτές μπορούν να χρησιμοποιηθούν με ένα μικρό κόστος ενώ άλλες είναι πολύ πιθανό να ενσωματωθούν σε κινητές συσκευές τα επόμενα χρόνια.

Η χρήση εξυπηρετικών αλλά και “αξιόπιστων” υπηρεσιών από κινητές συσκευές, είναι σχεδόν βέβαιο ότι θα αλλάξει ριζικά το μέλλον των ηλεκτρονικών πληρωμών, των τραπεζικών συναλλαγών αλλά και την παγκόσμια βιομηχανία των τηλεπικοινωνιών. Για τον λόγο αυτό σε αυτήν την εργασία θα ασχοληθούμε παρακάτω με ένα πρωτόκολλο ηλεκτρονικών πληρωμών μέσω κινητών συσκευών και θα ελέγξουμε την ασφάλεια που παρέχεται μοντελοποιώντας το πρωτόκολλο με την βοήθεια των Colored Petri Nets και το εργαλείο CPN Tool.

2.1 Εφαρμογές

Οι εκπληκτικές δυνατότητες που παρέχει η τεχνολογία M-Commerce σε συνδυασμό με την αλματώδη ανάπτυξη της κινητής τηλεφωνίας παρέχουν ένα μεγάλο πλήθος υπηρεσιών. Μία συνοπτική λίστα των υπηρεσιών αυτών είναι η παρακάτω :

- Τραπεζικές συναλλαγές.
- Χρηματιστηριακές συναλλαγές.
- Υπηρεσίες μέσω Internet (Web-browsing, ενημέρωση, κλπ.).
- Ψυχαγωγία (downloading μουσικών κομματιών και video-clips, αποστολή ηλεκτρονικών καρτών κλπ).
- Αποστολή γραπτών μηνυμάτων (SMS, e-mail).
- Υπηρεσίες "πλοήγησης"
- Υπηρεσίες για εταιρείες /Οργανισμούς (Virtual Private Network-VPN, Intranet).

Αναλυτικότερα οι προσφερόμενες υπηρεσίες είναι οι ακόλουθες:

Χρηματοοικονομικές

Οι υπηρεσίες αυτές μπορούν να αποτελέσουν ένα πολύτιμο εργαλείο για τους επαγγελματίες του κλάδου. Απευθύνονται, κυρίως, σε άτομα που η φύση της εργασίας τους απαιτεί έγκυρη και έγκαιρη ενημέρωση σε θέματα που αφορούν τόσο στην Ελληνική όσο και στην παγκόσμια οικονομική κοινότητα. Ουσιαστικά λοιπόν προσδίδονται νέες δυνατότητες ενημέρωσης ακόμα και στους απλούς επενδυτές, οι οποίοι χωρίς να παρακολουθήσουν κάποια τηλεοπτική εκπομπή, χωρίς να χρησιμοποιήσουν το Διαδίκτυο ή να διαβάσουν κάποια οικονομική εφημερίδα, ενημερώνονται άμεσα και οικονομικά, μέσω του κινητού τους τηλεφώνου, οποιαδήποτε στιγμή της ημέρας το θελήσουν.

Γενική ενημέρωση

Η ευκολία χρήσης, καθώς και η ποικιλία των πληροφοριών που προσφέρονται αποτελούν τα χαρακτηριστικά στοιχεία των υπηρεσιών αυτών. Δεν μπορεί να οριστεί αυστηρά καθορισμένη ομάδα αποδεκτών μια και η φύση των πληροφοριών που προσφέρονται είναι τέτοια, ώστε να μπορούν να φανούν χρήσιμες σε καθέναν. Τέτοιες πληροφορίες μπορεί να αφορούν τις

σημαντικότερες τελευταίες ειδήσεις, τα τελευταία αθλητικά νέα, την πρόγνωση του καιρού, τα θέατρα, τους κινηματογράφους, τα νούμερα της τελευταίας κλήρωσης του ΛΟΤΤΟ και του ΤΖΟΚΕΡ, την νικήτρια στήλη του ΠΡΟΠΟ, ωροσκόπια κλπ.

Αποστολή & λήψη δεδομένων & τηλεομοιοτυπία (fax)

Η σύγχρονη μορφή των επαγγελμάτων απαιτεί συχνά την άμεση πρόσβαση σε πληροφορίες. Η ασύρματη μετάδοση δεδομένων με τη χρήση ενός κινητού τηλεφώνου και ενός φορητού υπολογιστή μπορεί να προσφέρει αξιόλογες λύσεις σε επαγγελματίες, ιδίως τις ώρες που βρίσκονται εκτός γραφείου. Οποιαδήποτε στιγμή χρειαστεί, στελέχη επιχειρήσεων, χρηματιστές, οικονομικοί σύμβουλοι, εκπρόσωποι πωλήσεων μπορούν να έχουν πρόσβαση στο εταιρικό δίκτυο και σε βάσεις δεδομένων, αλλά και να χειριστούν τη διαπροσωπική και επαγγελματική τους επικοινωνία μέσω fax ή ηλεκτρονικό ταχυδρομείο (e-mail). Το μικρό μέγεθος του εξοπλισμού (κινητό τηλέφωνο και φορητό PC) αποτελεί επιπρόσθετο πλεονέκτημα στη λειτουργικότητα και την αποτελεσματικότητα της συγκεκριμένης υπηρεσίας.

Παιχνίδια και παρακολούθηση χώρου

Μπορεί τα παρεχόμενα παιχνίδια σήμερα να είναι ιδιαίτερα απλά και να αποτελούν μεταφορές κλασικών παιχνιδιών υπάρχουν όμως και περίπλοκες εφαρμογές που επιτρέπουν τη συμμετοχή πολλών χρηστών από κάθε μεριά του κόσμου. Μια ιδιαίτερη κατηγορία παιχνιδιών αποτελούν τα WAP casino. Η κατηγορία αυτή βρίσκεται ακόμη σε πρώιμο στάδιο αλλά αναμένεται να γνωρίσει ραγδαία ανάπτυξη στο μέλλον.

Από την άλλη πλευρά ο χρήστης μπορεί να εποπτεύει μέσω του κινητού ένα απομακρυσμένο χώρο (λαμβάνει snapshots του εποπτευόμενου χώρου ανά τακτά χρονικά διαστήματα, το εύρος των οποίων μπορεί να μεταβληθεί μόνο όταν παρατηρηθεί από το λογισμικό κάποια μεταβολή κίνησης). Οι υπηρεσίες αυτές βρίσκονται προς το παρόν σε πιλοτικό στάδιο και απευθύνεται σε εταιρείες αφού η δημιουργία ενός εξυπηρέτη wap παρουσιάζει δυσκολίες, ενώ η τιμή της δικτυακής κάμερας είναι αρκετά υψηλή. Στον τομέα της παρακολούθησης αναμένεται να αναπτυχθούν εφαρμογές που θα επιτρέπουν

στο χρήστη να έχει μεγαλύτερο έλεγχο στις λειτουργίες της κάμερας. Επιπρόσθετα με την έλευση συσκευών κινητής τηλεφωνίας με οθόνες υψηλότερης ανάλυσης και έγχρωμες η παραπάνω δυνατότητα θα αποκτήσει μεγαλύτερη λειτουργικότητα.

Ταξίδια & **Τουρισμός**

Πολλές τεχνολογίες, όπως το WAP, το i-mode, το iappli προσφέρουν απεριόριστες δυνατότητες στο χρήστη που είτε σχεδιάζει τις διακοπές του είτε βρίσκεται ήδη στο μέσο κάποιου ταξιδιού. Οι υπηρεσίες που παρέχονται είναι ποικίλες και καλύπτουν κάθε πτυχή ενός ταξιδιού. Μια μεγάλη κατηγορία ιστοτόπων ανα τον κόσμο είναι προσβάσιμες από τις νέες κινητές συσκευές της αγοράς και ασχολούνται με τα μέσα μετάβασης στην επιθυμητή χώρα ή πόλη και μερικοί ιστοτόποι μάλιστα προσφέρουν τη δυνατότητα κράτησης εισιτηρίων σε πραγματικό χρόνο.

Υπάρχουν σελίδες που παρέχουν στον ταξιδιώτη κάθε είδους πληροφορία αναφορικά με την υπό επίσκεψη χώρα ή πόλη. Περιλαμβάνονται πληροφορίες για τα αξιοθέατα, τόπους διαμονής, εστιατόρια, χώρους διασκέδασης ενώ δεν λείπουν και οι σελίδες μέσω των οποίων και με χρήση πιστωτικής κάρτας μπορεί ο χρήστης να κάνει κράτηση σε ξενοδοχείο ή να νοικιάσει αυτοκίνητο.

Ιατρικές υπηρεσίες

Από τις βασικότερες και σπουδαιότερες και προφανώς από τις νεότερες υπηρεσίες που προσφέρονται στο χρήστη κινητής τηλεφωνίας. Η υπηρεσία αυτή ενώνει το χρήστη και τον ιατρό ή οποιαδήποτε άλλον του τομέα υγείας παρέχοντας μια αδιάκοπη επικοινωνία μεταξύ ασθενών και θεράποντων ιατρών αλλά και μεταξύ νοσοκομείων και οργανισμών για πιο αποτελεσματική αντιμετώπιση προβλημάτων υγείας. Ο ασθενής μπορεί να έχει πρόσβαση στον ιατρικό του φάκελο, να δημιουργεί στατιστικά στοιχεία, να επικοινωνεί με ιατρούς για κλείσιμο ραντεβού ή ανανέωση συνταγής ή για έκτατες καταστάσεις, να έχει τη δυνατότητα ελέγχου της διαδικασίας που αφορά τη διατήρηση της υγείας του. Από την άλλη πλευρά και ο ιατρός έχει τη δυνατότητα να παίρνει στοιχεία και πληροφορίες από τον ιατρικό φάκελο

του ασθενούς, από τα προσωπικά στοιχεία του ασθενούς ή από το νοσοκομείο και να αντιμετωπίζει αποδοτικότερα τα οποιαδήποτε περιστατικά.

WAP Διαδικτυακές Πύλες (portals)

Τον τελευταίο καιρό οι διαδικτυακές πύλες έχουν γίνει ιδιαίτερα δημοφιλείς στο χώρο του Διαδικτύου. Μέσω των πυλών αυτών ο χρήστης μπορεί να βρει οργανωμένες ανά κατηγορίες, πλήθος από ηλεκτρονικές σελίδες χωρίς να χρειάζεται ο ίδιος να αναλώνει χρόνο για αυτό που ψάχνει. Δεν είναι λίγες οι εταιρείες που έχουν θέσει σε λειτουργία WAP διαδικτυακές πύλες. Οι WAP διαδικτυακές πύλες στη συντριπτική πλειοψηφία τους ακολουθούν ιδιαίτερα λιτή σχεδίαση έτσι ώστε στη μικρή οθόνη του κινητού τηλεφώνου να απεικονίζονται όσο το δυνατό περισσότεροι τόποι. Οι περισσότερες από τις WAP διαδικτυακές πύλες συναντώνται και σε HTML έκδοση ώστε ο χρήστης να μπορεί να κάνει έρευνα και από το σπίτι του χωρίς τις ακριβές χρεώσεις των ασύρματων δικτύων.

Άλλες εφαρμογές και υπηρεσίες που παρέχονται μέσω m-commerce είναι επιγραμματικά οι ακόλουθες:

- Χρηματοοικονομικές συναλλαγές(M-Banking, M-Broking, Mobile Πληρωμές)
- Αγορές προϊόντων
- Mobile Δημοπρασίες
- Mobile Παιχνίδια, Music and Video

Το βασικό χαρακτηριστικό όλων των παραπάνω υπηρεσία – εφαρμογών, είναι ότι για να χρησιμοποιήσει κάποιος χρήστης μία υπηρεσία θα πρέπει να καταβάλει και κάποιο αντίτιμο, δηλαδή να πληρώσει. Άρα βασικό μέλημα μιας τέτοιας τεχνολογίας είναι ο ασφαλής τρόπος πληρωμής μέσω της κινητής συσκευής. Παρακάτω θα δούμε τις απαραίτητες εγγυήσεις και την ασφάλεια που πρέπει να διέπουν τέτοιου είδους συναλλαγές.

2.2 Εγγυήσεις και ασφάλεια (επιγραμματικά)

Όταν μιλάμε για ασφάλεια (security) και ιδιαίτερα για καταστάσεις που έχουν να κάνουμε με πρωτόκολλα ασφαλείας (security protocols), πρέπει να έχουμε στο μυαλό μας τα παρακάτω. Ο βασικός σκοπός κάθε ασφαλούς πρωτοκόλλου και κάθε συστήματος που θέλει να χαρακτηρίζεται ως "ασφαλές", είναι να παρέχει χαρακτηριστικά ποιότητας υπηρεσιών (quality of services). Αυτά τα χαρακτηριστικά έχουν διαφορετική διάσταση για κάθε σύστημα, και κάθε ένα από αυτά έχει ένα δικό του τρόπο υλοποίησης μέσα σε ένα σύστημα επικοινωνιών. Τα χαρακτηριστικά αυτά, ενός "ασφαλούς" πρωτοκόλλου είναι τα ακόλουθα σύμφωνα με τα [6] και [7]:

- **Αυθεντικότητα (Authentication)**: δυνατότητα απόδειξης της ταυτότητας των χρηστών του συστήματος
- **Μυστικότητα (Privacy)**: διασφάλιση της μη ανάγνωσης των δεδομένων από μη εξουσιοδοτημένους χρήστες
- **Ακεραιότητα (Integrity)**: αποτροπή μετατροπής των δεδομένων από μη εξουσιοδοτημένους χρήστες
- **Μη αποποίηση (Non – repudiation)**: η δυνατότητα απόδειξης της αποστολής ή της παραλαβής ενός μηνύματος

Ένα "ασφαλές" πρωτόκολλο θα πρέπει να είναι σχεδιασμένο ώστε να παρέχει ένα ή περισσότερα από τα παραπάνω χαρακτηριστικά ανάλογα με το σκοπό για τον οποίο δημιουργείται. Ο τρόπος και τα τεχνολογικά μέσα με τα οποία το πρωτόκολλο θα μπορεί να παρέχει τέτοιου είδους υπηρεσίες ποικίλει και κάποιοι από αυτούς τους μηχανισμούς είναι οι ψηφιακές υπογραφές (digital signature), η συμμετρική κρυπτογράφιση (encipherment), οι συναρτήσεις hash (hash functions) και ανταλλαγή μηνυμάτων αυθεντικοποίησης (authentication exchange) [6].

2.3 Προβλήματα που σχετίζονται με την ασφάλεια πληρωμών μέσω κινητών συσκευών

Τα τελευταία χρόνια στον τομέα των τηλεπικοινωνιών, έχουν κάνει την εμφάνισή τους ποικίλα περιβάλλοντα εφαρμογών κινητής τηλεφωνίας, σε διάφορες γεωγραφικές περιοχές σε όλο τον κόσμο. Στην Ευρώπη έχουμε

επικεντρωθεί στο Wireless Application Protocol (WAP), στην Ιαπωνία έχει μεγάλη επιτυχία το i-mode και στην βόρεια Αμερική υπάρχουν διάφορα άλλα συστήματα. Παρόλο την πληθώρα αυτών των τεχνολογιών, υπάρχουν κάποια κοινά προβλήματα, τα οποία σχετίζονται με την ασύρματη επικοινωνία και καθιστούν προβληματικές τις πληρωμές μέσω κινητών συσκευών.

Η παρεχόμενη "ασφάλεια", ειδικότερα στο επίπεδο εφαρμογών (Application Layer), είναι πολύ χαμηλότερη από την υπάρχουσα ασφάλεια των ενσύρματων δικτύων. Επιπρόσθετα, η απόδοση αλλά και η επεξεργαστική ισχύς των συσκευών αυτών είναι σαφώς περιορισμένη σε σχέση με τα υπάρχοντα συστήματα Η/Υ που χρησιμοποιούνται για ηλεκτρονικές συναλλαγές μέσω internet.

Στη συνέχεια θα αναλύσουμε τα προβλήματα που προκύπτουν σε θέματα ασφαλείας και σε θέματα απόδοσης των κινητών συσκευών που χρησιμοποιούνται σε ηλεκτρονικές ασύρματες συσκευές [3].

2.3.1 Περιορισμοί Ασφάλειας

Οι περισσότερες εφαρμογές ηλεκτρονικού εμπορίου σε ασύρματα περιβάλλοντα αποτελούνται τουλάχιστον από τρεις συμμετέχοντες: τον κάτοχο της κινητής συσκευής (mobile **user**), τον φορέα κινητής τηλεφωνίας (mobile **operator**) και τον παροχέα των υπηρεσιών ηλεκτρονικών συναλλαγών (service **provider**). Αυτό απαιτεί δύο "ασφαλής" επικοινωνίες, μία του κατόχου της κινητής συσκευής (mobile user) με τον φορέα κινητής τηλεφωνίας (mobile operator) και μία του φορέα κινητής τηλεφωνίας (mobile operator) με τον παροχέα των υπηρεσιών ηλεκτρονικών συναλλαγών (service provider). Για να επιτευχθεί μία τέτοια "ασφαλής" συναλλαγή, θα πρέπει και ο κάτοχος της κινητής συσκευής (mobile user) και ο παροχέας των υπηρεσιών ηλεκτρονικών συναλλαγών (service provider) να έχουν απόλυτη εμπιστοσύνη προς τον φορέα κινητής τηλεφωνίας (mobile operator). Η εμπιστοσύνη αυτή απαιτείται για τους ακόλουθους λόγους:

- Κανένας από τους δύο τελικούς συμβαλλόμενους (user και provider) δεν μπορεί να είναι σίγουρος ότι η επικοινωνία πίσω από

τον proxy server του φορέα κινητής τηλεφωνίας (mobile operator) συνεχίζει να είναι “ασφαλής”.

- Το περιεχόμενο των μηνυμάτων κατά την διαδικασία της επικοινωνίας, αποκρυπτογραφούνται και κρυπτογραφούνται στον proxy server του φορέα κινητής τηλεφωνίας (mobile operator), κάνοντας ευάλωτη μία πιθανή επίθεση από έναν εξωτερικό παράγοντα (hacker) ή από ένα εσωτερικό παράγοντα (τον “ασφαλή” φορέα κινητής τηλεφωνίας).
- Δεν είναι δυνατή μία επικοινωνία end-to-end η οποία να εμπεριέχει και το χαρακτηριστικό της αυθεντικότητας (authentication).

Η κυρίαρχη τεχνολογία που χρησιμοποιείται στην σύνδεση φορέα κινητής τηλεφωνίας (mobile operator) με τον παροχέα των υπηρεσιών ηλεκτρονικών συναλλαγών (service provider) πραγματοποιείται με το ασφαλές πρωτόκολλο SSL. Ο τρόπος με τον οποίο μεταδίδονται ηλεκτρονικά μηνύματα μεταξύ του φορέα κινητής τηλεφωνίας (mobile operator) και του κατόχου της κινητής συσκευής (mobile user), εξαρτάται από το περιβάλλον εφαρμογής της ασύρματης επικοινωνίας και από την κλάση (class) της κινητής συσκευής. Στο WAP χρησιμοποιείται το πρωτόκολλο Wireless Transport Layer Security (WTLS), στο HDML χρησιμοποιείται μία φόρμα του κρυπτογραφημένου Handheld Device Transport Protocol, στο i-mode δεν παρέχει καμία υπηρεσία κρυπτογράφησης δεδομένων και στο i-appli είναι δυνατόν να χρησιμοποιηθεί SSL επικοινωνία με την κινητή συσκευή αλλά μόνο σε Java κινητές συσκευές. Τέλος, η GoAmerica, η Palm και άλλες εταιρίες του χώρου, χρησιμοποιούν τα δικά του ιδιωτικά πρωτόκολλα ασφαλείας, που κατά κύριο λόγο βασίζονται σε elliptic curve cryptosystems.

Μια περίπτωση που μπορεί να χρησιμοποιηθεί end-to-end ασφάλεια στο WAP είναι με την χρήση WMLScript with cryptographic API. Όμως υπάρχει μόνο ένα WMLScript cryptographic API, το signText(), το οποίο παρέχει στους χρήστες την δυνατότητα να υπογράφουν ψηφιακά τα μηνύματα τους. Δυστυχώς όμως, τα προγράμματα WMLScript στέλνονται μέσω του gateway του operator και έτσι δεν υπάρχει καμία εγγύηση για την ακεραιότητα της διαδικασίας. Ακόμη και αν ο operator είναι απόλυτης εμπιστοσύνης, υπάρχει

πάντα ο τρόπος επίθεσης. Από το γεγονός ότι το WAP δεν έχει πιστοποίηση υψηλού επιπέδου, ο χρήστης δεν έχει κανέναν τρόπο να αποδείξει την ταυτότητα του αποστολέα της αίτησης υπογραφής, κάτι που τον κάνει ευάλωτο στην κλασική επίθεση man-in-the-middle. Για παράδειγμα μπορεί ο εισβολέας να προσποιηθεί ότι είναι ένα νόμιμος server, να ζητήσει από τον χρήστη να υπογράψει ένα μήνυμα και να το χρησιμοποιήσει εν αγνοία του χρήστη σε μία πλαστή συναλλαγή. Στην περίπτωση μιας πιστοποίησης με κωδικό πρόσβασης, μπορεί πολύ εύκολα ο εισβολέας να δημιουργήσει ένα περιβάλλον WML που να μοιάζει απόλυτα με το περιβάλλον του νόμιμου server και να του ζητήσει να δώσει τον κωδικό του ώστε να τον χρησιμοποιήσει αργότερα προς όφελος του.

Μερικές εταιρίες ανάπτυξης εφαρμογών βασίζονται σε εξειδικευμένες εφαρμογές με σκοπό την παροχή ασφάλειας μέσω επιπρόσθετων μεθόδων πιστοποίησης, αλλά κάτι τέτοιο έχει ιδιαίτερες απαιτήσεις από την συσκευή του χρήστη, κάτι που δεν μπορούν να παρέχουν οι μαζικές – εμπορικές κινητές συσκευές της αγοράς, τουλάχιστον για τα επόμενα χρόνια. Για αυτές τις απαιτήσεις των κινητών συσκευών σε πόρους και επεξεργαστική ισχύ θα μιλήσουμε στην επόμενη παράγραφο.

2.3.2Περιορισμοί Απόδοσης και Πόρων

Εκτός από τους περιορισμούς που είδαμε σε θέματα ασφάλειας των κινητών συστημάτων πληρωμών, πρέπει να έχουμε υπ' όψιν μας και τα προβλήματα που σχετίζονται με την χαμηλή απόδοση των ασύρματων συσκευών μας αλλά και την μικρή σχετικά επεξεργαστική ισχύ τους.

Πρώτα από όλα, ο διακομιστής των υπηρεσιών στο ασύρματο δίκτυο έχει χαμηλότερες δυνατότητες εν σύγκριση με τους αντίστοιχους ενσύρματους διακομιστές, σε τομείς όπως ταχύτητα μεταφοράς μηνυμάτων (bandwidth), τις καθυστερήσεις μετάδοσης μηνυμάτων (longer latencies) και σαφώς τα περισσότερα λάθη στην αποστολή μηνυμάτων (errors and conflicts). Επιπλέον, λόγω της μαζικής παραγωγής και διάθεσης στην αγορά φθηνών κινητών συσκευών, υπάρχουν περιορισμοί που έχουν σχέση με την είσοδο και

έξοδο των δεδομένων (μικρά πληκτρολόγια και οθόνες), μικρή επεξεργαστική ισχύς και ελάχιστη μνήμη.

Για τον λόγο αυτό, τα πρωτόκολλα ηλεκτρονικών πληρωμών που έχουν ήδη αναπτυχθεί στα ενσύρματα δίκτυα, δεν είναι εφαρμόσιμα αν δεν τροποποιηθούν κατάλληλα, σε ασύρματα συστήματα. Για να παρουσιάσουμε το πρόβλημα αυτό, θα χρησιμοποιήσουμε ένα παράδειγμα μίας συναλλαγής με το ευρέως διαδεδομένο ασφαλές πρωτόκολλο στις ηλεκτρονικές συναλλαγές, το SET.

Από μετρήσεις που έχουν γίνει στο πρωτόκολλο SET για τον υπολογισμό του μέσου όρου του μέγεθος ενός μηνύματος σε μία τυπική συναλλαγή μεταξύ του κατόχου της κινητής συσκευής και του εμπόρου, έχουν δείξει ότι το μέγεθος αυτό είναι 18608 bytes. Σε μία σύνδεση των 9600 bps, η αποστολή αυτού του μεγέθους μηνύματος θα διαρκέσει περίπου 15 δευτερόλεπτα. Να σημειώσουμε ότι τα 9600 bps είναι η τυπική ταχύτητα μεταφοράς που χρησιμοποιείται σήμερα στις τεχνολογίες WAP και i-mode. Επίσης στον χρόνο αυτό δεν έχουμε συμπεριλάβει ούτε τον χρόνο επεξεργασίας του μηνύματος από τον κάτοχο της κινητής συσκευής και τον έμπορο, ούτε τον χρόνο αποστολή και λήψης των μηνυμάτων που απαιτούνται για να ξεκινήσει η ηλεκτρονική συναλλαγή – πληρωμή. Το συμπέρασμα είναι ότι τα κινητά συστήματα επικοινωνίας δεν έχουν μεγάλες δυνατότητες στον τομέα της επικοινωνίας.

Επιπρόσθετα, σύμφωνα με το πρωτόκολλο SET, η διαδικασία της συναλλαγής απαιτεί από την συσκευή του πελάτη (κάτοχο της κινητής συσκευής) τους ακόλουθους κρυπτογραφικούς υπολογισμούς:

- τέσσερις υπολογισμούς τυχαίων αριθμών
- οκτώ SHA-1 hash υπολογισμούς
- μία υπογραφή ιδιωτικού κλειδιού με 1024 bit RSC
- τρεις με οκτώ αποκρυπτογραφήσεις δημόσιου κλειδιού με 1024 bit RSA
- και μία διαδικασία σύμφωνη με το Optimal Asymmetric Encryption Padding (OAEP)

Παρόλο που τέτοιου είδους κρυπτογραφικές λειτουργίες είναι εφικτό να πραγματοποιηθούν στις συσκευές που μελετάμε, πολλές από αυτές δεν υποστηρίζονται ή δεν διατίθενται από αντίστοιχες εφαρμογές σήμερα. Επιπρόσθετα, η επεξεργασία και η εκτέλεση των παραπάνω λειτουργιών είναι ιδιαίτερα χρονοβόρα σε περίπτωση που η κινητή συσκευή δεν είναι εξοπλισμένη με ειδικό hardware. Σε αυτήν την καθυστέρηση έρχονται να προστεθούν και οι καθυστερήσεις από τις διαδικασίες επεξεργασίας των συστημάτων του εμπόρου και της τράπεζας αλλά και τον χρόνο που απαιτείται για την μεταξύ τους επικοινωνία.

Κεφάλαιο 3

Πρωτόκολλα - Συστήματα “ασφαλών” πληρωμών μέσω κινητών συσκευών

3 Πρωτόκολλα - Συστήματα "ασφαλών" πληρωμών μέσω κινητών συσκευών

Ο βασικός σκοπός της έρευνας στον τομέα του m-Commerce Payments είναι να βρεθεί ένας τρόπος ώστε το πρωτόκολλο επικοινωνίας που θα κυριαρχήσει στις κινητές συσκευές, να συνδυάζει ***μικρό χρόνο απόκρισης*** για τον χρήστη και παράλληλα να εξασφαλίσει όλες τις ***απαιτήσεις ασφάλειας, πιστοποίησης και αυθεντικότητας***. Αυτό διότι, από την μία πλευρά το μικρό bandwidth, η μικρή επεξεργαστική ισχύς και από την άλλη η ευκολία της υποκλοπής ηλεκτρονικών μηνυμάτων από το ευάλωτο μέσο μετάδοσης των ασύρματων επικοινωνιών, που είναι ο αέρας (ατμόσφαιρα), καθιστούν τις παραπάνω απαιτήσεις ιδιαίτερα δύσκολες στην πράξη.

Όταν λοιπόν κάποιος προσπαθεί να υλοποιήσει ένα πρωτόκολλο προσαρμοσμένο πάνω στις πληρωμές μέσω κινητών συσκευών, ίσως είναι καλό να χρησιμοποιήσει ένα υπάρχον, αποδοτικό και ασφαλές πρωτόκολλο από τις ενσύρματες πληρωμές, παρά να δημιουργήσει ένα εξολοκλήρου νέο πρωτόκολλο. Αυτό σκέφτηκαν οι Konrad Wrona και Guido Zavagli το 1999 [4] και πρότειναν την προσαρμογή του ευρέως αποδεκτού πρωτοκόλλου SET στις πληρωμές μέσω κινητών συσκευών. Είναι όμως δυνατόν να προσαρμοστεί το πρωτόκολλο SET στις πληρωμές μέσω κινητών συσκευών; Από όσο απέδειξαν οι Konrad Wrona και Guido Zavagli αυτό είναι εφικτό. Επικεντρώθηκαν περισσότερο στην εξασφάλιση των παρακάτω κριτηρίων:

- **Απλότητα** (Simplicity)
- **Σύντομοι χρόνοι απόκρισης** (συγκρίσιμοι με αυτούς του e-commerce) (Short Response Time)
- **Ασφάλεια** (Security)
- **Εξυπηρέτηση** (Convenience)

Στην πράξη έδωσαν περισσότερη βαρύτητα στο 2ο και στο 3ο κριτήριο, παραβλέποντας τα άλλα δύο. Στην επόμενη παράγραφο θα δούμε τις τρεις παραλλαγές που πρότειναν μαζί με άλλα προτεινόμενα πρωτόκολλα.

3.1 Αναφορά στα πιο γνωστά πρωτόκολλα - ερευνητικά προγράμματα

Λόγω όλων των παραπάνω περιορισμών που πρέπει να ικανοποιούν τα πρωτόκολλα πληρωμών κινητών συσκευών, δημιουργήθηκαν δύο κατηγορίες πρωτοκόλλων:

- Συστήματα που χρησιμοποιούν διακομιστή μεσολάβησης **(proxy solutions)**, στα οποία είναι εφικτή η χρήση ενός υπάρχον πρωτοκόλλου ηλεκτρονικών πληρωμών με χρήση proxy server ενός ενσύρματου δικτύου.
- Συστήματα που δεν χρησιμοποιούν διακομιστή μεσολάβησης **(non-proxy solutions)**, τα οποία έχουν αναπτυχθεί ειδικά για τις δυνατότητες των κινητών συσκευών.

3.1.1 Συστήματα πληρωμών με proxy

Η κατηγορία αυτή έχει αναπτυχθεί ιδιαίτερα από το γεγονός ότι υπάρχει ήδη εγκατεστημένη μία ακριβή υποδομή στο χώρο του ηλεκτρονικού εμπορίου, η οποία δεν μπορεί να αλλάξει εύκολα και ανέξοδα από την μια μέρα στην άλλη. Για να αλλάξει η υπάρχουσα τεχνολογία των ηλεκτρονικών πληρωμών θα πρέπει, εκτός από το να αλλάξουν τα πρωτόκολλα συναλλαγών, να αλλάξει και όλο το υπάρχον λογισμικό ώστε να εξασφαλιστούν και οι ασύρματες επικοινωνίες.

Μια απλή και σχετικά μικρού κόστους λύση, είναι να χρησιμοποιηθεί ένας proxy server ανάμεσα από τον χρήστη της κινητής συσκευής και την υποδομή της πληρωμής (payment infrastructure). Ο ρόλος του proxy server θα είναι να συμπεριφέρεται ως ένας συνηθισμένος πελάτης ως προς την υπάρχουσα ενσύρματη υποδομή ενώ παράλληλα θα είναι υπεύθυνος της πιστοποίησης (authentication) του χρήστη της κινητής συσκευής και της εξουσιοδότησης (authorization) της πληρωμής.

Πολλά συστήματα που χρησιμοποιούν διακομιστή μεσολάβησης (proxy solutions) έχουν αναπτυχθεί τα τελευταία χρόνια και από ότι δείχνουν οι εξελίξεις, η τεχνολογία προχωράει προς αυτήν την κατηγορία για τους λόγους

που αναφέραμε νωρίτερα. Παρακάτω θα αναλύσουμε μερικά από αυτά τα συστήματα – πρωτόκολλα.

3.1.1.1 Standard SET για κινητές συσκευές

Σε αυτήν την παραλλαγή του πρωτοκόλλου SET, ουσιαστικά η κινητή συσκευή παίρνει τον ρόλο ενός Η/Υ. Όμως λόγω του ότι δεν υπάρχει ακόμη η απαραίτητη τεχνολογία για να λειτουργήσει μία κινητή συσκευή με το SET και οι υποτιθέμενοι χρόνοι απόκρισης θα είναι πολύ μεγάλοι, η εξέλιξη παρέχεται από τις επόμενες παραλλαγές του SET.

3.1.1.2 3D SET ή SET Wallet Server για κινητές συσκευές

Με σκοπό λοιπόν την υλοποίηση ηλεκτρονικών πληρωμών μέσω κινητών συσκευών και του πρωτοκόλλου SET, προτάθηκε από τους Konrad Wrona και Guido Zavagli το πρωτόκολλο SET Wallet Server. Έτσι ο πελάτης πρέπει απλά να πιστοποιήσει τον εαυτό του στον SET Wallet Server αποστέλλοντας του τις πληροφορίες του λογαριασμού της πιστωτικής του κάρτας. Κάνοντας την υπόθεση ότι υπάρχει σχέση εμπιστοσύνης ανάμεσα στον πελάτη και τον SET Wallet Server, η παραλλαγή αυτή έχει πολλά πλεονεκτήματα όταν χρησιμοποιείται σε πληρωμές μέσω κινητών συσκευών:

- Η κινητή συσκευή δεν χρειάζεται να επεξεργαστεί τις συναλλαγές του SET μόνη της αλλά προωθεί την όλη διαδικασία στον SET Wallet Server και απλά παίρνει πίσω μια απάντηση ως επιβεβαίωση της πληρωμής.
- Η μετάδοση μηνυμάτων με ασύρματη επικοινωνία περιορίζεται σε λίγα μόνο μηνύματα, άρα μειώνονται και οι πιθανότητες υποκλοπής.
- Μπορεί να επιτευχθεί καλύτερη απόδοση, απλά αυξάνοντας τις υπολογιστικές δυνατότητες του SET Wallet Server.
- Σε περίπτωση που χρειαστεί αναβάθμιση σε μια νεότερη έκδοση SET, αυτό θα γίνει μόνο στον SET Wallet Server.

Από την άλλη όμως υπάρχουν και κάποια μειονεκτήματα:

- Ο χρήστης της κινητής συσκευής πρέπει να έχει πλήρη εμπιστοσύνη στον SET Wallet Server.

- Αν συμβεί μία επίθεση στον SET Wallet Server όλες οι πληροφορίες του πελάτη, το ιδιωτικό κλειδί του, το δημόσιο κλειδί του και το ψηφιακό του πιστοποιητικό του υποκλέπτονται με μια κίνηση.
- Ο SET Wallet Server μπορεί να χρησιμοποιήσει τις πληροφορίες που συλλέγει από τις κινήσεις του πελάτη και να δημιουργήσει προφίλ χρηστών, τα οποία να χρησιμοποιεί για άλλους εμπορικούς σκοπούς.

3.1.1.3 Split SET Server για κινητές συσκευές

Η παραλλαγή αυτή του SET είναι μία υβριδική μορφή ανάμεσα από το Standard SET και του SET Wallet Server. Το πλεονέκτημά της είναι ότι ο πελάτης δεν απαιτείται να έχει πλήρη εμπιστοσύνη στον Split SET Server διότι το ιδιωτικό του κλειδί το κατέχει μόνο ο ίδιος μέσα στην κινητή συσκευή ενώ το δημόσιο και το ψηφιακό του πιστοποιητικό το έχει ο Split SET Server.

3.1.1.4 Mobile Chip Electronic Commerce για κινητές συσκευές

Το πρωτόκολλο αυτό είναι μια προσαρμογή του Chip Electronic Commerce που επίσης βασίζεται σε proxy server. Η κινητή συσκευή αποτελείται από δύο μέρη, το Mobile Chip Electronic Commerce Client και το Mobile Chip Electronic Commerce Server (proxy). Ο Server αναλαμβάνει να φέρει εις πέρας τις λειτουργίες του πρωτοκόλλου που απαιτούν κάποια υπολογιστική ισχύ, όπως να ελέγχει τα ψηφιακά πιστοποιητικά των συμβαλλόμενων, να δημιουργεί και να αποστέλλει τα μηνύματα στον έμπορο. Ο ρόλος του client είναι να παρέχει την ασφάλεια που χρειάζεται στην συναλλαγή όπως την πιστοποίησης (authentication) του χρήστη της κινητής συσκευής και την εξουσιοδότησης (authorization) της πληρωμής μέσω ενός cryptogram EMV που είναι ενσωματωμένο στην smart card.

3.1.2 Συστήματα πληρωμών χωρίς proxy

Αν και αυτή η κατηγορία δεν έχει αναπτυχθεί αρκετά για τους λόγους που αναφέραμε προηγούμενα, σε κάποιες επιχειρήσεις που δεν είχαν ήδη εγκατεστημένο ένα σύστημα e-commerce, είναι ίσως ευκολότερο και οικονομικότερο να χρησιμοποιήσουν ένα πρωτόκολλο το οποίο θα εξυπηρετεί μόνο της πληρωμές κινητών συσκευών παρά να τροποποιήσουν ένα ήδη υπάρχον. Τις περισσότερες φορές σε αυτήν την κατηγορία χρησιμοποιείται ένα μοντέλο τριών συμβαλλόμενων μεριών, με τον παροχέα πληρωμών

(payment provider) να παίζει τον ρόλο και του issuer και του acquirer. Κάποια πρωτόκολλα – συστήματα αυτής της κατηγορίας περιγράφονται παρακάτω.

3.1.2.1 Paybox

Το σύστημα αυτό διατίθεται μόνο στην Γερμανία αλλά ο στόχος είναι να διαδοθεί και να χρησιμοποιηθεί και σε άλλες ευρωπαϊκές χώρες. Το σύστημα αυτό επιτρέπει την πραγματοποίηση μιας αγορά μέσω ενός κινητού τηλεφώνου από ένα κατάστημα στο διαδίκτυο, από χρήστες άλλων κινητών τηλεφώνων και από άλλους μεταπωλητές (π.χ. οδηγούς ταξί). Το σύστημα αποτελείται από τρία μέρη και ο χρήστης της κινητής συσκευής χρειάζεται ένα τραπεζικό λογαριασμό ή μία πιστωτική κάρτα από την οποία θα γίνει η πληρωμή.

3.1.2.2 Mobile Operator Payment Systems

Τα τελευταία χρόνια γίνεται μια προσπάθεια από τους παροχείς υπηρεσιών κινητής τηλεφωνίας να μπουν δυναμικά στην αγορά των πληρωμών μέσω κινητών τηλεφώνων. Αυτό εκ πρώτης όψεως φαίνεται λογικό διότι έχουν στη διάθεσή τους:

a) Μία μεγάλη βάση πελατών
b) Μία υποδομή πληρωμής λογαριασμών
c) Απόλυτο έλεγχο μέσω της SIM κάρτας και άρα δεν υπάρχει θέμα ασφάλειας

Ωστόσο, για να ικανοποιηθεί το βασικό αίτημα των πελατών της εταιρίας, για αγορές από όλα τα ηλεκτρονικά καταστήματα του κόσμου, θα πρέπει να συνεργαστούν πολλοί ενδιάμεσοι παροχείς, κάτι που περιπλέκει τα δεδομένα.

3.2 Αναλυτική περιγραφή των εγγυήσεων ασφάλειας που πρέπει να χαρακτηρίζουν τα πρωτόκολλα "ασφαλών" πληρωμών

Στην παράγραφο αυτή θα δώσουμε μια πιο αναλυτική περιγραφή των εγγυήσεων ασφαλείας της παραγράφου 2.2 επικεντρώνοντας την ανάλυσή

μας σε πρωτόκολλα “ασφαλών” πληρωμών. Οι βασικοί στόχοι ενός πρωτοκόλλου ασφαλών πληρωμών είναι οι παρακάτω:

- ***Εμπιστευτικότητα*** *(confidentiality):* Διασφάλιση της προσπελασιμότητας της πληροφορίας μόνο από όσους έχουν τα απαραίτητα δικαιώματα.
- ***Ακεραιότητα*** *(integrity):* Διαφύλαξη της ακρίβειας και της πληρότητας της πληροφορίας και των μεθόδων επεξεργασίας αυτής.
- ***Διαθεσιμότητα*** *(availability):* Διασφάλιση της προσπελασιμότητας της πληροφορίας σε εξουσιοδοτημένους χρήστες όποτε απαιτείται.
- ***Έλεγχος αυθεντικότητας*** *(authentication):* Εξακρίβωση της ταυτότητας του χρήστη είτε με passwords είτε με προσωπικούς αριθμούς αναγνώρισης (Personal Identification Numbers - PIN’s) και διάφορα άλλα.
- ***Μη αποποίηση της ευθύνης*** *(non – repudiation):* Ολοκλήρωση συναλλαγής όπου κάποιος μετά δεν μπορεί να ισχυρισθεί ότι δεν συμμετείχε σ’ αυτήν.
- ***Εξουσιοδότηση*** *(authorization):* Παραχώρηση δικαιωμάτων στο χρήστη από τον ιδιοκτήτη

Εκτός όμως από τις παραπάνω εγγυήσεις, ένα πρωτόκολλο ασφαλών πληρωμών πρέπει να παρέχει και τις παρακάτω εγγυήσεις [9]:

Money conservation [10] ή όπως αλλιώς είναι γνωστή αυτή η ιδιότητα, **money atomicity**. Αυτή η βασική εγγύηση συναλλαγών, εξασφαλίζει ότι δεν υπάρχει πιθανότητα δημιουργίας ή καταστροφής χρημάτων κατά την διάρκεια της μεταφοράς ηλεκτρονικών χρηματικών ποσών. Η παραβίαση αυτής της εγγύησης μπορεί να συμβεί σε μία κατάσταση όπου έχει «πέσει» το website του πωλητή, από μία κατάσταση αποτυχημένης συναλλαγής είτε από μία επίθεση εισβολέα σε κάποιο επίπεδο του πρωτοκόλλου.

Μία άλλη πολύ σημαντική εγγύηση που πρέπει να παρέχει ένα ασφαλές πρωτόκολλο ηλεκτρονικών πληρωμών είναι, η **αποφυγή διπλών χρεώσεων** (**double spending**) [11]. Ο βασικός στόχος αυτής της εγγύησης ασφαλείας

είναι να μην εκτελεστεί μία παραγγελία πάνω από μία φορά χωρίς την έγκριση του αγοραστή. Σε ένα freshness ή ένα replay attack μπορεί να επιτευχθεί διπλή χρέωση από έναν εισβολέα, χρησιμοποιώντας μηνύματα από προηγούμενες νόμιμες συναλλαγές. Ένας βασικός μηχανισμός αντιμετώπισης τέτοιου είδους επιθέσεων είναι η χρήση ενός αριθμού (session id) ο οποίος θα διασφαλίζει το πόσο πρόσφατο (freshness) είναι ένα μήνυμα και κατά πόσο είναι μέρος της συγκεκριμένης συναλλαγής.

Μία ακόμη πολύ σημαντική εγγύηση είναι η διασφάλιση ότι δεν υπάρχει πιθανότητα να πληρώσει ο πελάτης για κάποιο προϊόν και να μην το παραλάβει αλλά και αντίστροφα. Αυτή η εγγύηση ονομάζεται **goods atomicity** [10] και σε κάθε διμερής συναλλαγή ενός πρωτοκόλλου ασφαλών πληρωμών πρέπει και ο πελάτης και ο πωλητής να ελέγχουν την ιδιότητα αυτή ώστε να μην παραβιαστεί σε περίπτωση που «πέσει» το website του πωλητή, σε περίπτωση αποτυχημένης συναλλαγής είτε από μία επίθεση εισβολέα. Σε περίπτωση που οι συμμετέχοντες στην συναλλαγή είναι περισσότεροι από δύο η εγγύηση αυτή πρέπει να διασφαλιστεί ανάμεσα σε όλους τους συμβαλλόμενους. Αν σκεφτούμε για παράδειγμα ότι κάποιος πελάτης πληρώνει για ένα αεροπορικό εισιτήριο μόνο σε περίπτωση που πραγματοποιηθεί επιτυχώς η αντίστοιχη συναλλαγή του με έναν άλλο πωλητή για την διαμονή του σε ξενοδοχείο. Στην περίπτωση αυτή είναι απαραίτητη η εξασφάλιση της ιδιότητας **good atomicity.**

Distributed payment atomicity [11]. Η εγγύηση αυτή εξασφαλίζει να συμπεριληφθούν αλληλεπιδράσεις μεταξύ διαφορετικών και ανεξάρτητων συμβαλλόμενων σε μία απλή συναλλαγή. Ένας τρόπος για να επιτευχθεί αυτή η εγγύηση σε ετερογενή περιβάλλοντα, όπου οι εφαρμογές χρησιμοποιούν πρωτόκολλα επικοινωνίας χωρίς ποικιλία στις συναλλαγές, είναι το Transaction Internet Protocol – TIP [12]. Το TIP έχει την δυνατότητα να αποτρέπει διάφορα είδη επιθέσεων, όπως δύο είδη denial of service attacks, transaction corruption attack, packet-sniffing attack και man-in-the-middle attack. Για τον λόγο αυτό πολλά συστήματα πληρωμών χρησιμοποιούν το TIP ώστε να εξασφαλίσουν και αυτήν την εγγύηση πληρωμών.

Certified delivery [10]. Η εγγύηση αυτή απαιτεί την εξασφάλιση των δύο εγγυήσεων money conservation και goods atomicity αλλά παράλληλα απαιτεί όλοι οι συμβαλλόμενοι να μπορούν να αποδείξουν ευαίσθητες πληροφορίες της συναλλαγής. Έτσι όταν συμβεί κάτι κατά την διάρκεια της συναλλαγής, οι συμβαλλόμενοι μπορούν να αποδείξουν τι συνέβη στην online συναλλαγή αντιπαραθέτοντας τα στοιχεία του offline.

Η εγγύηση Goods atomicity παρέχει μια υψηλού επιπέδου ιδιότητα **δίκαιης ανταλλαγής (fair exchange)** [13] και [14]. Τι σημαίνει δίκαιη ανταλλαγή μεταξύ πελάτη και πωλητή; Ότι κανένας συμβαλλόμενος δεν μπορεί να αποκομίσει κάποιο αγαθό από έναν άλλο, εξαπατώντας τον. Να σημειώσουμε ότι ένας πελάτης εξαπατά έναν πωλητή όταν αποκτήσει τα αγαθά που θέλει χωρίς ο πωλητής να λάβει το αντίτιμο. Από την άλλη ένας πελάτης εξαπατά ένας πελάτη όταν αυτός πληρωθεί για μία παραγγελία αλλά δεν αποστείλει τα αντίστοιχα αγαθά στον πελάτη. Τέλος να επισημάνουμε ότι η εγγύηση αυτή διασφαλίζει ότι πραγματοποιήθηκε η συναλλαγή χρημάτων για κάποιο αγαθό αλλά δεν εξασφαλίζει ότι τα αγαθά που παρήγγειλε ο πελάτης είναι ακριβώς αυτά για τα οποία πλήρωσε με την παραγγελία του.

Μία ακόμη εγγύηση υψηλού επιπέδου είναι η **Protection of participants' interests [15]** με την οποία διασφαλίζεται ότι οι συμβαλλόμενοι της συναλλαγής θα αποκομίσουν αυτά που νόμιμα δικαιούνται. Αυτό συμβαίνει πολλές φορές όταν χωρίς να υπάρχει η πρόθεση να εξαπατήσει ο ένας τον άλλο, κάποιος από τους συμβαλλόμενους δεν αποκόμισε από την συναλλαγή ότι θα ήθελε.

Όλες οι παραπάνω εγγυήσεις που αναπτύξαμε αλλά και πολλές ακόμη, πρέπει να λαμβάνονται υπόψιν κατά την διάρκεια ανάπτυξης ενός αφαλούς πρωτοκόλλου ηλεκτρονικών πληρωμών. Επίσης πάνω σε αυτές τις ιδιότητες βασίζονται και πολλές ερευνητικές εργασίες [9] που προσομοιώνουν τα υπάρχοντα πρωτόκολλα και τα ελέγχουν προς την κατεύθυνση αυτή.

3.3 Τυπική ανάλυση πρωτοκόλλων "ασφαλών" πληρωμών με χρωματισμένα δίκτυα Petri

Για την τυπική ανάλυση συστημάτων «ασφαλών» ηλεκτρονικών πληρωμών υπάρχουν πολλές γλώσσες μοντελοποίησης και προσομοίωσης συνοδευόμενες από αντίστοιχα εργαλεία. Υπάρχουν όμως και τα Χρωματισμένα δίκτυα Petri. Τα Χρωματισμένα δίκτυα Petri δεν υπερέχουν όλων των άλλων γλωσσών μοντελοποίησης. Απλά είναι εξαιρετικά χρήσιμα και μαζί με άλλες γλώσσες μοντελοποίησης παίζουν σημαντικό ρόλο στη σχεδίαση και ανάλυση προηγμένων συστημάτων. Μερικοί από τους λόγους που χρησιμοποιούνται Χρωματισμένα δίκτυα Petri στην τυπική ανάλυση πρωτοκόλλων «ασφαλών» πληρωμών θα αναφερθούν παρακάτω. Ωστόσο δεν θα πρέπει να παραλείψουμε την χρησιμότητα και όλων των άλλων μεθόδων τυπικής ανάλυσης.

Τα Χρωματισμένα δίκτυα Petri παριστάνονται γραφικά. Η γραφική απεικόνιση είναι διαισθητικά πολύ ελκυστική. Επειδή τα διαγράμματα των Χρωματισμένων δικτύων Petri μοιάζουν με πολλά από τα σχέδια που κάνουν οι σχεδιαστές και οι μηχανικοί, καθώς κατασκευάζουν και αναλύουν ένα σύστημα. Επιπρόσθετα, τα Χρωματισμένα δίκτυα Petri γίνονται πολύ εύκολα κατανοητά, ακόμα και από άτομα που δεν είναι εξοικειωμένα με αυτά.

Τα Χρωματισμένα δίκτυα Petri περιγράφουν με σαφήνεια και τις καταστάσεις και τις δράσεις, αντίθετα με τις περισσότερες γλώσσες περιγραφής συστημάτων που περιγράφουν είτε τις καταστάσεις, είτε τις δράσεις, αλλά όχι και τα δύο μαζί. Χρησιμοποιώντας Χρωματισμένα δίκτυα Petri, ο αναγνώστης μπορεί εύκολα να μετατοπίσει το επίκεντρο της προσοχής του, από τις καταστάσεις στις δράσεις, και αντίστροφα.

Τα Χρωματισμένα δίκτυα Petri έχουν μια σημασιολογία που δομείται πάνω στην πραγματικά ταυτόχρονη εκτέλεση αντί πάνω στη **χρονική διαμεσολάβηση (interleaving)**. Σε μια χρονική διαμεσολάβηση είναι αδύνατο να υπάρχουν δύο δράσεις στο ίδιο βήμα, κι έτσι αυτό σημαίνει ότι οι δράσεις μπορούν να εκτελεστούν μόνο η μία μετά την άλλη, με οποιαδήποτε σειρά. Είναι ευκολότερο να δουλέψει κανείς με πραγματικά ταυτόχρονη

εκτέλεση, γιατί είναι πιο κοντά στον τρόπο με τον οποίο σκέφτονται οι άνθρωποι για τις ταυτόχρονες δράσεις.

Τα Χρωματισμένα δίκτυα Petri προσφέρουν ιεραρχικές περιγραφές. Αυτό σημαίνει ότι ένα μεγάλο Χρωματισμένο δίκτυο Petri μπορεί να δομηθεί συνδέοντας πολλά μικρά Χρωματισμένα δίκτυα Petri μεταξύ τους, με ένα καλά ορισμένο τρόπο. Οι ιεραρχικές δομές των Χρωματισμένων δικτύων Petri παίζουν παρόμοιο ρόλο με αυτόν των υπορουτινών και των διαδικασιών στις γλώσσες προγραμματισμού. Η ύπαρξη Ιεραρχικών Χρωματισμένων δικτύων Petri κάνει δυνατή τη μοντελοποίηση μεγάλων συστημάτων με ένα εύκολο και σπονδυλωτο τρόπο.

Τα Χρωματισμένα δίκτυα Petri συνδυάζουν την περιγραφή του ελέγχου και του συγχρονισμού με την περιγραφή του χειρισμού δεδομένων (data manipulation). Δηλαδή ένας μοναδικός γράφος παρουσιάζει το περιβάλλον, τις συνθήκες ενεργοποίησης και τα απότελεσματα μιας δράσης. Πολλές άλλες περιγραφικές γλώσσες χρησιμοποιούν γράφους που περιγράφουν μόνο το περιβάλλον μιας δράσης, ενώ η συμπεριφορά της περιγράφεται ξεχωριστά.

Τα Χρωματισμένα δίκτυα Petri προσφέρουν διαδραστικές προσομοιώσεις, των οποίων τα αποτελέσματα παρουσιάζονται απευθείας στο διάγραμμα του Χρωματισμένου δικτύου Petri. Με την προσομοίωση είναι δυνατή η **αποσφαλμάτωση (debugging)** ενός μεγάλου μοντέλου καθώς κατασκευάζεται, όπως ακριβώς ένας προγραμματιστής διορθώνει τα ξεχωριστά τμήματα ενός προγράμματος, αφού τα τελειώσει. Επίσης μπορούν να ελεγχθούν οι τιμές δεδομένων των μετακινούμενων μαρκών.

Τα Χρωματισμένα δίκτυα Petri έχουν μια καλά ορισμένη σημασιολογία η οποία ορίζει με σαφήνεια τη συμπεριφορά κάθε δικτύου. Λόγω αυτής της σημασιολογίας, γίνεται δυνατή η προσομοίωση των Χρωματισμένων δικτύων Petri και η εφαρμογή των μεθόδων ανάλυσης.

Τα Χρωματισμένα δίκτυα Petri έχουν πολύ λίγες, αλλά ισχυρές αρχές. Ο ορισμός των Χρωματισμένων δικτύων Petri είναι αρκετά σύντομος και

δομείται πάνω σε καθιερωμένες έννοιες, τις οποίες γνωρίζουν πολλοί σχεδιαστές συστημάτων, από τα απλά μαθηματικά και από τις γλώσσες προγραμματισμού. Αυτό σημαίνει ότι είναι σχετικά εύκολη η εκμάθηση χρήσης των Χρωματισμένων δικτύων Petri και ότι είναι δυνατό να αναπτυχθούν ισχυρές μέθοδοι ανάλυσης.

Τα Χρωματισμένα δίκτυα Petri έχουν ένα αριθμό από τυπικές μεθόδους ανάλυσης με τη βοήθεια των οποίων, μπορούν να αποδειχθούν οι ιδιότητες των Χρωματισμένων δικτύων Petri. Οι δύο πιο σημαντικές μέθοδοι ανάλυσης είναι η ανάλυση του χώρου καταστάσεων και τα **αμετάβλητα διανύσματα θέσεων (place invariants)**.

Τα Χρωματισμένα δίκτυα Petri έχουν ένα σύνολο από εργαλεία λογισμικού που υποστηρίζουν τη σχεδίαση, την προσομοίωση και την ανάλυση τους [18]. Η ύπαρξη τέτοιων εργαλείων είναι πολύ σημαντική για την πρακτική χρήση των Χρωματισμένων δικτύων Petri.

Για όλους τους παραπάνω λόγους, επιλέξαμε τα Χρωματισμένα δίκτυα Petri με σκοπό να μοντελοποιήσουμε ένα πρωτόκολλο «ασφαλών» πληρωμών μέσω κινητών συσκευών. Χρησιμοποιήσαμε για την σχεδίαση, την μοντελοποίηση και την τυπική ανάλυση του πρωτοκόλλου το εργαλείο CPN Tool [18]. Στη συνέχεια, με την βοήθεια της γλώσσας ML και της δυνατότητας του εργαλείου CPN Tool για τυπική ανάλυση, προχωρήσαμε στον έλεγχο των εγγυήσεων ασφαλείας που πρέπει να διέπουν το πρωτόκολλό μας.

3.4 Σενάρια επιθέσεων και μοντέλα επιθέσεων εισβολέα

Κατά την ανάλυση ενός «ασφαλούς» πρωτοκόλλου πληρωμών θα πρέπει να συμπεριλάβουμε όλα τα πιθανά σενάρια επιθέσεων από κάποιον εισβολέα. Κάτι που όμως είναι πολύ δύσκολο διότι υπάρχουν πάρα πολλές πιθανές επιθέσεις. Αν προσπαθούσαμε να περιγράψουμε την έννοια του εισβολέα (intruder) θα λέγαμε ότι είναι κάποιος που θέλει να υπονομεύσει την ασφάλεια του πρωτοκόλλου και αν τα καταφέρει να επωφεληθεί από αυτό.

Κάποιες από τις ενέργειες που μπορεί να κάνει ένας εισβολέας σύμφωνα με το [8] είναι:

- Να αποτρέψει την αποστολή κάποιου μηνύματος
- Να δημιουργήσει ένα αντίγραφο του μηνύματος και να το αποθηκεύσει σε μια προσωρινή μνήμη
- Να υποκλέψει ένα μήνυμα αποτρέποντας την αποστολή του και παράλληλα να δημιουργήσει ένα αντίγραφό του
- Να τροποποιήσει ένα μήνυμα
- Να καταστρέψει ένα μήνυμα
- Να ξαναστείλει ένα παλιό μήνυμα
- Να καθυστερήσει την αποστολή ενός μηνύματος
- Να αναδιατάξει την σειρά ενός μηνύματος

Όλες οι παραπάνω ενέργειες δεν είναι ανεξάρτητες. Τις περισσότερες φορές ένας εισβολέας συνδυάζει δύο ή και περισσότερες από τις παραπάνω ενέργειες με σκοπό να γίνει όσο το δυνατόν πιο ισχυρός και να καταφέρει να παραβιάσει τελικά το πρωτόκολλο. Παρακάτω αναφέρουμε μια σύντομη κατάταξη των επιθέσεων σε πρωτόκολλα ηλεκτρονικών πληρωμών [19].

Elementary protocol flaws. Σε αυτήν την κατηγορία ανήκουν όλες οι επιθέσεις εισβολέων που συμβαίνουν σε πρωτόκολλα τα οποία παρέχουν ελάχιστη ή και καθόλου προστασία ενάντια σε επιθέσεις.

Password/key guessing flaws. Στην κατηγορία αυτή οι επιθέσεις προς το πρωτόκολλο προέρχονται από το γεγονός ότι οι χρήστες πολύ συχνά επιλέγουν τα password τους από ένα μικρό σύνολο κοινών λέξεων. Ακόμη και σε περίπτωση που το password δημιουργείται αυτόματα από ένα ψευδο-τυχαίο κλειδί (pseudo-random key), είναι πολύ πιθανό το κλειδί αυτό να είναι με τέτοιο τρόπο κατασκευασμένο που ο εισβολέας να μπορεί να το αναπαράγει. Αυτό έχει ως αποτέλεσμα ο χώρος αναζήτησης του κλειδιού να μειώνεται αισθητά σε σύγκριση με όλα τα πιθανά κλειδιά που μπορεί να χρησιμοποιήθηκαν. Να σημειωθεί ότι για τον παραπάνω λόγο η κατηγορία αυτή ονομάζεται και dictionary attacks ή και verifiable-text attacks.

Stale message flaws. Η κατηγορία αυτή είναι γνωστή και ως Replay Attack. Εδώ ο εισβολέας σε αντίθεση με την άμεση επίθεση προς ένα ασφαλές πρωτόκολλο, προσπαθεί να χρησιμοποιήσει γνήσια μηνύματα του πρωτοκόλλου τα οποία ούτε μπορεί να τα διαβάσει ούτε να τα δημιουργήσει. Υπάρχουν τρεις περιπτώσεις:

- Να χρησιμοποιηθούν μηνύματα από την μία εκτέλεση του πρωτοκόλλου σε μία άλλη εκτέλεση (Run External Attack)
- Να χρησιμοποιηθούν μηνύματα από ίδια την εκτέλεση του πρωτοκόλλου (Run Internal Attack)

Τα δύο παραπάνω σενάρια συμβαίνουν μόνο όταν το πρωτόκολλο δεν έχει κανένα μηχανισμό για session keys ώστε να εντοπιστεί η παλαιότητα του μηνύματος

- Να επαναμεταδοθούν κάποια από τα μηνύματα του πρωτοκόλλου (Message Destination Attack)

Parallel session flaws ή Oracle session attack. Σε αυτήν την περίπτωση, ο εισβολέας έχει την ικανότητα να αντλήσει τις απαραίτητες πληροφορίες μεταδίδοντας κατάλληλα μηνύματα του πρωτοκόλλου. Στη συνέχεια μπορεί και εκτελεί το πρωτόκολλο ταυτόχρονα πάνω από μία φορά με βασικό σκοπό να μπορέσει να διπλοχρεώσει τον πελάτη ή να καταφέρει να παραποιήσει κάποιο μήνυμα του πρωτοκόλλου.

Internal Protocol Flaws. Η κατηγορία αυτή των επιθέσεων προκύπτει όταν ένας από του συμβαλλόμενους της συναλλαγής αποτύχει να ολοκληρώσει όλες τις απαιτούμενες αποστολές μηνυμάτων.

Cryptosystem Flaws. Οι αλγόριθμοι κρυπτογράφησης και τα σχετικά πρωτόκολλα σχεδιάζονται και χρησιμοποιούνται με σκοπό να παρέχουν authentication και confidentiality στα δεδομένα. Όταν κάποια υλοποίηση εξασφαλίζει όλα τα χαρακτηριστικά που απαιτεί ο αλγόριθμος και το

πρωτόκολλο, αλλά δεν διασφαλίζει authentication και confidentiality στα δεδομένα, τότε λέμε ότι συμβαίνει cryptosystem – related flaws.

Πολλές από τις παραπάνω κατηγορίες και σενάρια επιθέσεων θα τις ελέγξουμε με το μοντέλο του εισβολέα που έχουμε υλοποιήσει στο CP-Net και περιγράφεται αναλυτικά στην παράγραφο 4.1.8.

Κεφάλαιο 4

Ένα “ασφαλές” πρωτόκολλο πληρωμών μέσω κινητών συσκευών

4 Ένα "ασφαλές" πρωτόκολλο πληρωμών μέσω κινητών συσκευών

Στο κεφάλαιο αυτό θα επικεντρωθούμε σε ένα πρωτόκολλο «ασφαλών» πληρωμών μέσω κινητών συσκευών [20]. Το πρωτόκολλο αυτό έχει σχεδιαστεί έχοντας λάβει υπόψιν όλες τις προδιαγραφές που πρέπει να πληρεί ένα πρωτόκολλο «ασφαλών» πληρωμών μέσω κινητών συσκευών. Τις προϋποθέσεις ασφαλείας αλλά και τους περιορισμούς απόδοσης και πόρων που έλαβαν υπόψιν του οι σχεδιαστές του τις έχουμε αναλύσει διεξοδικά στις παραγράφους 3.2 και 3.3.

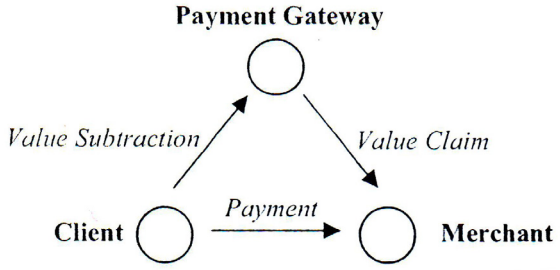
Τα βασικά σημεία του πρωτοκόλλου είναι ότι:

1. χρησιμοποιεί συμμετρική κρυπτογραφία, συναρτήσεις Hash και key generations, με σκοπό την μείωση της απαιτούμενης επεξεργαστικής ισχύς (συγκριτική αξιολόγηση στον πίνακα 1).

Cryptographic Operations		**SET**	**iKP**	**Το πρωτόκολλο που μελετήσαμε**
1. Public-key encryptions	C	1	1	-
	M	1	-	-
	PG	1	-	-
2. Public-key decryptions	C	-	-	-
	M	1	-	-
	PG	2	1	-
3. Signature generations	C	1	1	-
	M	3	1	-
	PG	1	1	-
4. Signature verifications	C	2	3	-
	M	2	2	-
	PG	1	2	-
5. Symmetric-key encryptions/decryptions	C	2	-	4
	M	-	-	5
	PG	1	-	2
6. Hash functions	C	3	2	2
	M	2	4	-
	PG	-	1	-
7. Keyed-hash functions	C	-	-	2
	M	-	1	2
	PG	-	-	1
8. Key generations	C	-	-	2
	M	-	-	1
	PG	-	-	1

Πίνακας 1*. Συγκριτική αξιολόγηση των κρυπτογραφικών λειτουργιών των πρωτοκόλλων SET,iKP και του ασφαλούς πρωτοκόλλου που μελετήσαμε*

2. Το μοντέλο αποτελείται από 5 συμβαλλόμενα μέρη: τον client «C», τον merchant «M», τον issuer «I» που είναι η τράπεζα του πελάτη, τον acquirer «A» που είναι η τράπεζα του εμπόρου και τον payment gateway «PG» που είναι ο ενδιάμεσος κρίκος ανάμεσα στις τράπεζες του πελάτη και του εμπόρου, σε ένα ιδιωτικό ασφαλές δίκτυο αλλά ανάμεσα στον ίδιο τον πελάτη και τον έμπορο. Μία σχηματική αναπαράσταση του μοντέλου φαίνεται στο σχήμα 1



Σχήμα 1*. Σχηματική αναπαράσταση του μοντέλου*

3. Τα βασικά βήματα της ανταλλαγής μηνυμάτων του πρωτοκόλλου φαίνονται στο σχήμα 2

Merchant Registration Protocol

[1] C→M: $\{ID_C, X, n\}_K$

[2] M→C: h(n, X)

Payment Protocol

[3] C→M: ID_c, I, TIDReq, MIDReq

[4] M→C: $\{TID, ID_M\}_{Xi}$

[5] C→M: $\{OI, Price, ID_C, ID_I,$
$MAC[(Price, h(OI), ID_M,), Y_i]\}_{Xi,}$
$MAC[(OI, Price, ID_C, ID_I), X_{i+1}]$

[6] M→PG: $\{MAC[(Price, h(OI), ID_M), Y_i], h(OI), i,$
$TID, Price, ID_C, ID_I\}_{Zj}, j, ID_M,$
$MAC[(h(OI), i, TID, ID_C, ID_I), Z_{j+1}]$

(Κάτω από το ιδιωτικό δίκτυο των τραπεζών)

[7] PG→I: $MAC[(Price, h(OI), ID_M,), Y_i], h(OI), i,$
$TID, Price, ID_C, ID_M, h(Z_{j+1})$

[8] PG→A: Price, ID_M

[9] I, A→PG: Yes/No, $\{h(OI), Yes/No, h(Z_{j+1})\}_{Yi}$,
$h(Yes/No, h(OI), h(Y_i))$

[10] PG→M: $\{$ Yes/No, $\{h(OI), Yes/No, h(Z_{j+1})\}_{Yi}$,
$h(Yes/No, h(OI), h(Y_i))\}_{Zj+1}$

[11] M→C: $\{\{h(OI), Yes/No, h(Z_{j+1})\}_{Yi}\}_{Xi+1}$

Σχήμα 2*. Τα βασικά βήματα της ανταλλαγής μηνυμάτων του πρωτοκόλλου*

4. Οι συμβολισμοί επεξηγούνται στον πίνακα 2

Σύμβολο	Επεξήγηση – ρόλος
{C, M, PG, I, A}	Το σύνολο των μελών Client, Merchant, Payment Gateway, Issuer και Acquirer αντίστοιχα
ID_p	Η ταυτότητα του P. Περιέχει πληροφορίες επικοινωνίας για τον P
TID	Η ταυτότητα της συναλλαγής συμπεριλαμβανομένης της ώρας και της ημέρας της συναλλαγής
OI	Πληροφορίες συναλλαγής. OI = {TID, h(OD), Price} όπου το OD είναι η περιγραφή της συναλλαγής και Price η τιμή της
Yes/No	Η κατάσταση της συναλλαγής
TIDReq	Αίτηση για TID
MIDReq	Αίτηση για MID
$\{M\}_x$	Το μήνυμα M συμμετρικά κρυπτογραφημένο με το κλειδί x
h(X)	Η hash function του μηνύματος X
MAC(X,K)	Message Authentication Code (MAC) του μηνύματος X με το κλειδί K

***Πίνακας 2**. Συμβολισμοί του πρωτοκόλλου*

5. Έχει γίνει η παραδοχή ότι ο PG και ο M έχουν ένα κοινό μυστικό κλειδί Z και ο C με τον I έχουν ένα κοινό μυστικό κλειδί Y.

Περισσότερες λεπτομέρειες για την φύση και την λειτουργία του πρωτοκόλλου μπορείτε να βρείτε στο [20]. Εμείς στη συνέχεια θα αναλύσουμε την υλοποίηση – μοντελοποίηση του πρωτοκόλλου με χρήση CP-Nets.

4.1 Περιγραφή του χρωματισμένου δικτύου Petri για το πρωτόκολλο

4.1.1 Γενικές παραδοχές και δομή του μοντέλου

Στο μοντέλο μας προτείνουμε τις παρακάτω βασικές κατηγορίες για τα places που θα χρησιμοποιήσουμε στη συνέχεια:

- Places που αναπαριστούν τις πληροφορίες των συμβαλλόμενων (IDs)
- Places που αναπαριστούν τα κανάλια επικοινωνίας

- Places που αναπαριστούν κλειδιά με τα οποία πρόκειται να κρυπτογραφηθούν μηνύματα
- Places που αναπαριστούν πληροφορίες για την επιτυχή ή όχι εκτέλεση του πρωτοκόλλου.
- Places που αναπαριστούν μηνύματα ή τμήμα μηνυμάτων του πρωτοκόλλου.
- Places που αναπαριστούν πληροφορίες της συναλλαγής

Στον πίνακα 3 υπάρχουν όλοι οι τύποι δεδομένων που χρησιμοποιήθηκαν από το μοντέλο μας. Το colset KEY χρησιμοποιήθηκε για αναπαραστήσει τα κλειδιά με τα οποία γίνεται η κρυπτογράφηση των μηνυμάτων. Το colset SEC_KEY αποτελείται από ένα KEY και έναν ακέραιο αριθμό και αναπαριστά το κλειδί με το αντίστοιχο session number. Το colset sharedKey εμπεριέχει όλους τους πιθανούς τύπους κλειδιού ενός μηνύματος. Το colset sharedKeyList είναι μία λίστα από κλειδιά. Ο τύπος MAC χρησιμοποιείται σε places που τα tokens παίρνουν τιμή το αποτέλεσμα του Message Authentication Code το οποίο αποτελείτε από ένα string και ένα τύπο κλειδιού. Το colset ID βοηθάει στην αποθήκευση των Ids και είναι enumerated. Οι τύποι PRICE, OI, OD περιέχουν πληροφορίες για την τιμή, την παραγγελία και μια περιγραφή της παραγγελίας αντίστοιχα και είναι strings. Το colset REQUEST έχει δύο πιθανές τιμές για να αναπαραστήσει τις δύο αιτήσεις για το ID της παραγγελίας και το ID του merchant αντίστοιχα. Το colset MSG_PART είναι ένα union από πόλους πιθανούς τύπους που μπορούν να το αποτελούν και όπως λέει το όνομά του είναι ένα μέρος του συνολικού μηνύματος.

```
colset E = with e;
colset INT = int;
colset BOOL = bool;
colset STRING = string;
colset KEY=with K | Kcm | Kci | Kpgm;
colset SEC_KEY=product KEY*INT;
colset sharedKey=union sesKey:KEY
                       +secKey:SEC_KEY
                       +NO_KEY;
colset sharedKeysList=list sharedKey;
colset MAC=product STRING*SEC_KEY;
colset ID=with cID | mID | tID | iID | noID;
colset PRICE = STRING;
colset OI=STRING;
colset OD=STRING;
colset REQUEST=with TIDReq | MIDReq;
colset APPROVED=with Yes | No;
colset MSG_PART=union sesNo:INT
                      +pid:ID
                      +od:OD
                      +oi:OI
                      +pr:PRICE
                      +req:REQUEST
                      +app:APPROVED
                      +k:SEC_KEY
                      +str:STRING
                      +maced:MAC;
colset MESSAGE=list MSG_PART;
colset encrMSG=product MESSAGE*sharedKey;
colset dispatchedMSG=list encrMSG;
colset encrDispatchedMSG=product dispatchedMSG*sharedKey;
colset encrDispMSG=list encrDispatchedMSG;
colset HASH_PRMT=union mes:MESSAGE+Key:SEC_KEY;
```

Πίνακας 3*. Περιγραφή των declarations του μοντέλου*

Ο τύπος MESSAGE είναι μία λίστα από MSG_PART και όπως λέει το όνομά του αναπαριστά ένα μήνυμα. Όπως είδαμε στην περιγραφή του πρωτοκόλλου τα περισσότερα μηνύματα είναι κρυπτογραφημένα με κάποιο κλειδί. Για τον λόγω αυτό χρησιμοποιήσαμε το colset encrMSG το οποίο συντίθεται από ένα μήνυμα και ένα κλειδί. Για λόγους ευχρηστίας και επέκτασης του μοντέλου, με την δυνατότητα να ενσωματωθεί εισβολέας και μπορεί να γίνει αποθήκευση πολλών μηνυμάτων σε ένα place, χρησιμοποιήσαμε το colset dispatchedMSG το οποίο είναι λίστα από encrMSG. Πολλά μηνύματα που ανταλλάσσονται κατά την διάρκεια της εκτέλεσης του πρωτοκόλλου είναι διπλά κρυπτογραφημένα. Έτσι χρησιμοποιήσαμε το colset encrDispatchedMSG που αποτελείται από ένα dispatchedMSG και ένα κλειδί. Επίσης για τους ίδιους λόγους που χρησιμοποιήσαμε τον τύπο dispatchedMSG ως λίστα, δημιουργήσαμε και ένα τύπο encrDispMSG που είναι μία λίστα από

encrDispatchedMSG. Τέλος το colset HASH_PRMT είναι ένα union από ένα μήνυμα και ένα κλειδί διότι από όσο είδαμε τα ορίσματα της συνάρτησης Hash μπορεί να είναι δύο τύπων μήνυμα και κλειδί.

Στη συνέχεια θα αναλύσουμε τις συναρτήσεις που χρησιμοποιήσαμε κατά την μοντελοποίηση του πρωτοκόλλου και παρουσιάζονται στον πίνακα 4.

Η πρώτη συνάρτηση η Decrypt έχει δύο ορίσματα ως είσοδο και επιτρέφει ένα μήνυμα. Ο έλεγχος που πραγματοποιεί είναι αν το δεύτερο μέρος του πρώτου ορίσματος της (που είναι τύπου encrMSG) είναι ίσο με το κλειδί που κλίθηκε. Σε αυτήν την περίπτωση επιστέφεται το αποκρυπτογραφημένο μήνυμα ενώ σε αντίθετη περίπτωση επιστρέφει την κενή λίστα.
Η δεύτερη συνάρτηση η Dec κάνει ακριβώς την ίδια δουλειά με την Decrypt αλλά για μηνύματα που είναι διπλοκρυπτογραφημένα και γι' αυτό είναι τύπου encrDispatchedMSG και το επιστρεφόμενο μήνυμα είναι ένα κρυπτογραφημένο μήνυμα τύπου dispatchedMSG σε περίπτωση σωστής αποκρυπτογράφησης ή μια κενή λίστα σε περίπτωση λανθασμένης αποκρυπτογράφησης.

```
fun Decrypt(m:encrMSG,sKey:sharedKey):MESSAGE =
    if (#2 m) = sKey
      then (#1 m)
    else [];
fun
Dec(m:encrDispatchedMSG,sKey:sharedKey):dispatchedMSG
=
    if (#2 m) = sKey
      then (#1 m)
    else [];
fun Hash(prmt:HASH_PRMT):STRING=
    if HASH_PRMT.of_Key(prmt)
    then HASH_PRMT.mkstr(prmt)
         else if HASH_PRMT.of_mes(prmt)
              then HASH_PRMT.mkstr(prmt)
    else "";
```

Πίνακας 4. *Περιγραφή των συναρτήσεων του μοντέλου*

Τέλος η συνάρτηση Hash έχει ως όρισμα εισόδου μία μεταβλητή τύπου HASH_PRMT και κάνει τον εξής έλεγχο: αν το όρισμα είναι τύπου Key τότε δημιουργεί το αντίστοιχο string του κλειδιού, αν το όρισμα είναι τύπου mes

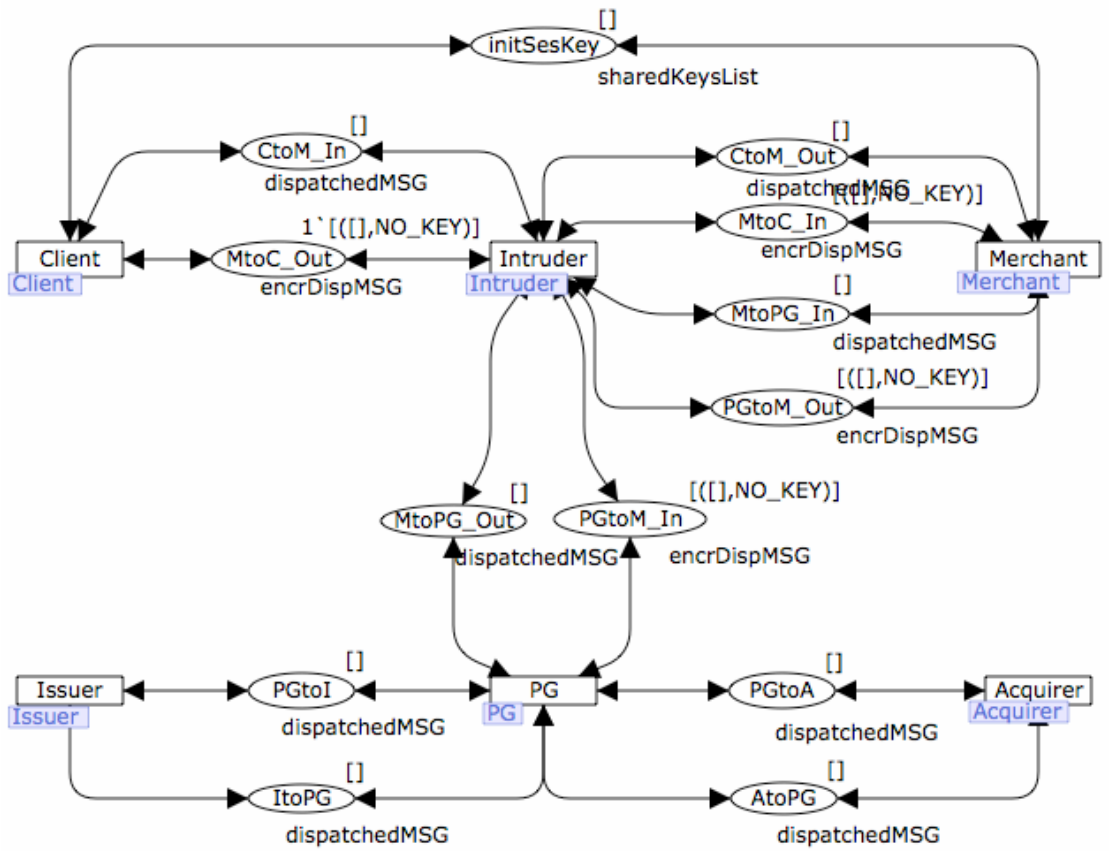
τότε δημιουργεί το αντίστοιχο string του μηνύματος. Σε κάθε άλλη περίπτωση επιστρέφει το κενό string.
Στο τελευταίο τμήμα ακολουθεί το τμήμα δηλώσεων των μεταβλητών και παρατίθεται στον πίνακα 5.

```
var anOD:OD;
var anInt,anInt2:INT;
var aPrice:PRICE;
var anApp,anApp2:APPROVED;
var anID,anID2,anID3:ID;
var aKey,aKey2:KEY;
var shKey:sharedKeysList;
var aMSGpart:MSG_PART;
var aMSG,aMSG1,aMSG2,
    aMSG3,aMSG4,aMSG5,
    aMSG6,aMSG7:MESSAGE;
var anEncrMSG,anEncrMSG2:encrMSG;
var aDispMSG,aDispMSG2,aDispMSG3:dispatchedMSG;
var anEncrDispMSG:encrDispMSG;
```

Πίνακας 5. *Περιγραφή των μεταβλητών του μοντέλου*

4.1.2 Περιγραφή του μοντέλου του Top Level

Στο σχήμα 3 παρουσιάζεται το υψηλότερο επίπεδο του μοντέλου το οποίο είναι δομημένο ιεραρχικά από πάνω προς τα κάτω. Όπως παρατηρούμαι υπάρχουν πέντε transitions που αναπαριστούν τα πέντε μέρη της συναλλαγής (Client, Merchant, PG, Issuer και Acquirer) και ένα transition που βρίσκεται ανάμεσα από τους Client, Merchant και PG και παίζει τον ρόλο του εισβολέα (Intruder). Με τον εισβολέα θα ασχοληθούμε εκτενέστερα στην παράγραφο 4.1.8. Και τα έξι αυτά transitions περιγράφονται από ένα μοντέλο το καθένα ξεχωριστά διότι είναι ιεραρχικά δομημένο το μοντέλο μας.



Σχήμα 3. *Το υψηλότερο επίπεδο του ιεραρχημένου μοντέλου μας*

Το δεύτερο που διακρίνουμε είναι ένα σύνολο από places που αναπαριστούν τα κανάλια επικοινωνίας του μοντέλου μας (CtoM_In, CtoM_Out, MtoC_In, MtoC_Out, MtoPG_In, MtoPG_Out, PGtoM_In, PGtoM_out, PGtoI, ItoPG, PGtoA, AtoPG), βρίσκονται ανάμεσα από τα transitions που αναπαριστούν τις οντότητες της συναλλαγής, έχουν συμβολικά ονόματα με βάση την κατεύθυνση μεταφοράς των μηνυμάτων, τα μηνύματα που μεταφέρουν είναι τύπου dispatchedMSG ή encrDispMSG ανάλογα με το αν το μήνυμα που μεταφέρουν έχει υποστεί μία ή δύο κρυπτογραφήσεις και στην αρχή το περιεχόμενό τους είναι κενό.

Τέλος, υπάρχει και ένα place (initSesKey) που είναι μία λίστα από sharedKey και θα αποθηκεύει το κλειδί του κάθε session της συναλλαγής. Στην αρχή είναι κενό.

4.1.3 Περιγραφή του μοντέλου του Client

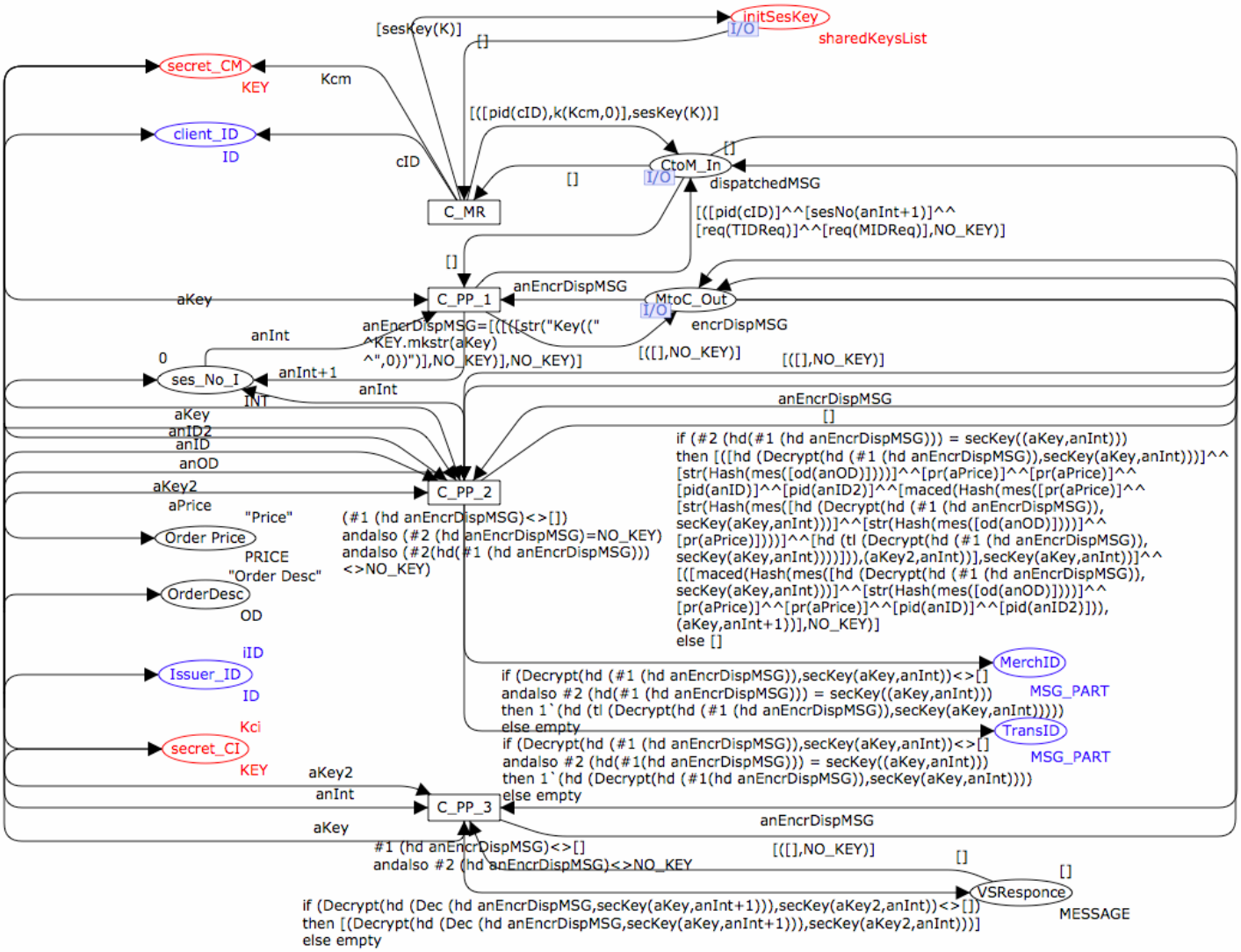
Οι περιγραφή των places του client έχει ως εξής:

- Ο Client έχει τέσσερα places (client_ID, MerchID, TransID και Issuer_ID) τα οποία έχουν μπλε χρώμα και αναπαριστούν τα IDs των παραπάνω οντοτήτων. Στην αρχή έχουν τιμή μόνο τα client_ID και issuer_ID διότι μόνο αυτά γνωρίζει ο client.
- Επίσης υπάρχουν τρία places (initSesKey, secret_CM και secret_CI) κόκκινου χρώματος που αναπαριστούν τα κλειδιά στα οποία έχει πρόσβαση ο client και αντιστοιχούν στο κλειδί του session της συναλλαγής, το κοινόχρηστο κλειδί του client με τον merchant και το κοινόχρηστο κλειδί του client με τον issuer.
- Δύο ακόμη places (Order Price και OrderDesc) αναπαριστούν την τιμή και την περιγραφή της συναλλαγής και έχουν αρχικές τιμές (τυπικά έχουμε ορίσει string).
- Το place ses_No_I αποθηκεύει το session number i της συγκεκριμένης συναλλαγής έχοντας αρχική τιμή 0 και αυξάνεται κατά μία μονάδα όταν εκτελείται το transition C_PP_1.
- Το place VSResponce το οποίο θα αποθηκεύσει την απάντηση για το αν η συναλλαγή ολοκληρώθηκε με επιτυχία.

Τα τέσσερα transition του client περιγράφονται παρακάτω:

- Το transition C_MR αναπαριστά το πρώτο βήμα (1)[1] εκτέλεσης του Merchant Registration Protocol. Με την εκτέλεση του transition παίρνουν τιμές τα places initSesKey, secret_CM, client_ID και ταυτόχρονα δημιουργείται το μήνυμα (1) του Merchant Registration Protocol και αποστέλλεται στο κανάλι επικοινωνίας CtoM_In ώστε να φτάσει στον merchant.

[1] Μέσα σε παρενθέσεις αναφέρουμε την αρίθμηση των μηνυμάτων όπως αναπαραστήσαμε το πρωτόκολλο στην παράγραφο 4 στο σχήμα 2



Σχήμα 4. *Το μοντέλο του client*

- Το transition C_PP_1 αναπαριστά το πρώτο βήμα (3) εκτέλεσης του Payment Protocol. Το transition έχει ως είσοδο από το κανάλι επικοινωνίας MtoC_Out την απάντηση του merchant στο αρχικό του μήνυμά του. Το transition δημιουργεί με τα δικά του δεδομένα το hash του δικού του Kcm το οποίο παίρνει από το place secret_CM ώστε να το συγκρίνει με το hash του Kcm που του έστειλε ο merchant. Μόνο αν αυτά τα δύο είναι απόλυτα ίδια ο client δημιουργεί το πρώτο μήνυμα (3) του Payment Protocol και το στέλνει στο κανάλι επικοινωνίας CtoM_In ώστε να φτάσει στον merchant.
- Το transition C_PP_2 αναπαριστά το δεύτερο βήμα (5) εκτέλεσης του Payment Protocol. Το transition έχει ως είσοδο από το κανάλι επικοινωνίας MtoC_Out την απάντηση του merchant στο προηγούμενο

του μήνυμά του. Κατά την εκτέλεσή του αποκρυπτογραφούνται τα MerchID και TransID και αποθηκεύονται στα αντίστοιχα places. Επίσης σε περίπτωση που το μήνυμα που παρέλαβε ο client είναι κρυπτογραφημένο με το σωστό κλειδί, τότε και μόνο τότε συντάσσει το νέο μήνυμα (παίρνοντας όλες τις απαραίτητες πληροφορίες από τα places Order Price, OrderDesc, ses_No_I, secret_CI, client_ID, secret_CM και κάνοντας χρήση της συνάρτησης Hash και Decrypt) και το στέλνει στο κανάλι επικοινωνίας CtoM_In ώστε να φτάσει στον merchant.

- Τέλος, το transition C_PP_3 αναπαριστά την παραλαβή του τελευταίου μηνύματος του Payment Protocol από τον merchant. Το transition έχει ως είσοδο από το κανάλι επικοινωνίας MtoC_Out την απάντηση του merchant για το αν έγινε επιτυχώς η συναλλαγή, ανάλογα με τις απαντήσεις που πήρε ο ίδιος από τον PG . Κατά την εκτέλεσή του αποκρυπτογραφούνται τα δεδομένα του μηνύματος με την βοήθεια των συναρτήσεων Decrypt, Dec και αποθηκεύεται η απάντηση στο place VSResponce. Από το place αυτό μπορούμε να εξάγουμε το συμπέρασμα για το αν έγινε σωστά η συναλλαγή. Κάτι που θα το εξετάσουμε διεξοδικά στην παράγραφο 4.2 όπου θα αναλύσουμε τον χώρο των καταστάσεων του μοντέλου.

4.1.4 Περιγραφή του μοντέλου του Merchant

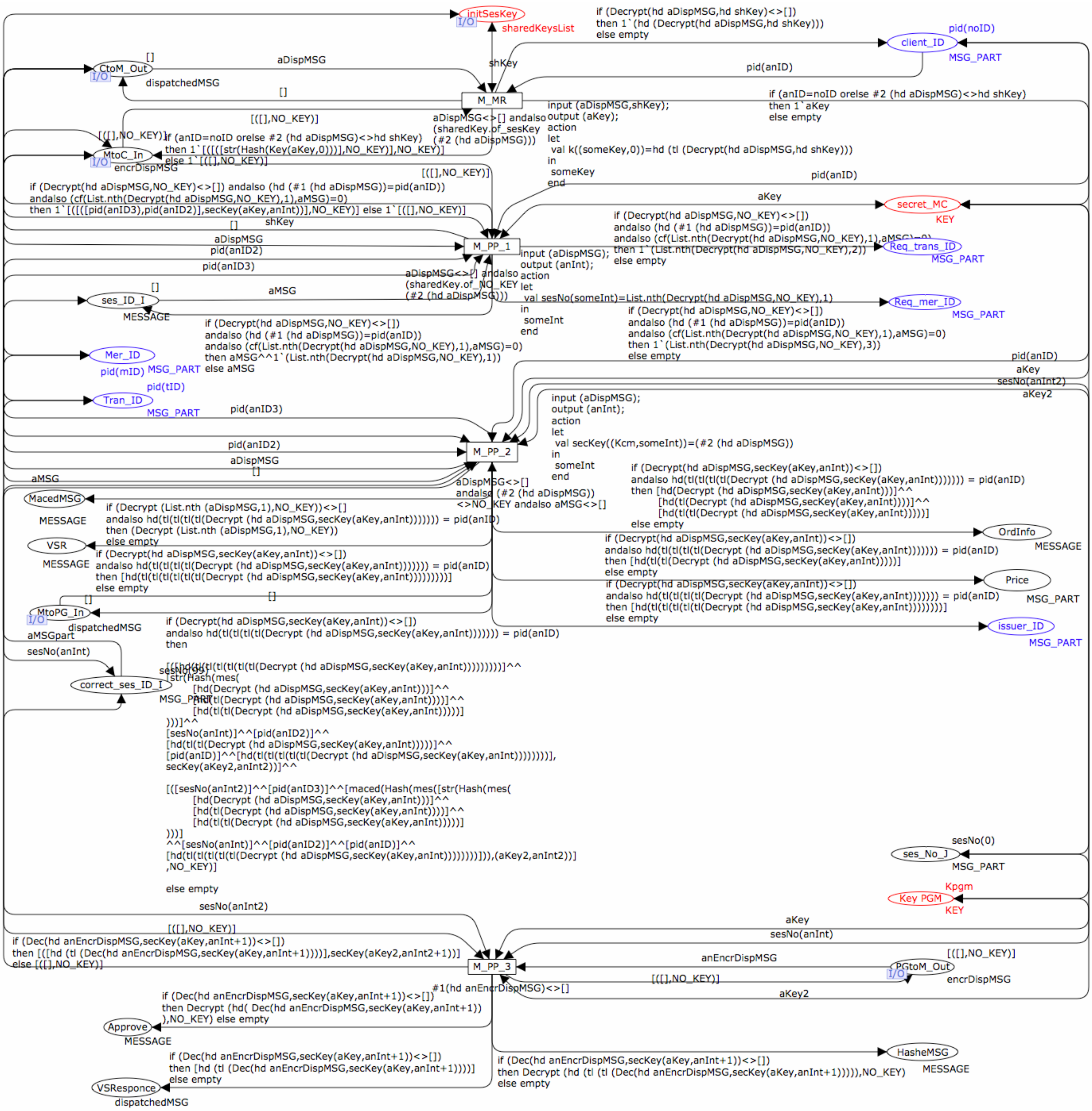
Οι περιγραφή των places του client έχει ως εξής:

- Ο Merchant έχει τέσσερα places (client_ID, MerchID, TransID και Issuer_ID) τα οποία έχουν μπλε χρώμα και αναπαριστούν τα IDs των παραπάνω οντοτήτων. Στην αρχή έχουν τιμή μόνο τα MerchID και TransID διότι μόνο αυτά γνωρίζει ο merchant.
- Επίσης υπάρχουν τρία places (initSesKey, secret_MC και Key_PGM) κόκκινου χρώματος που αναπαριστούν τα κλειδιά στα οποία έχει πρόσβαση ο merchant και αντιστοιχούν στο κλειδί του session της συναλλαγής, το κοινόχρηστο κλειδί του merchant με τον client και το κοινόχρηστο κλειδί του payment gateway με τον merchant.
- Τα places (OrdInfo και Price) θα αποθηκεύσουν την τιμή και την περιγραφή της συναλλαγής

- Δύο ακόμη places (Req_Trans_ID και Req_mer_ID) θα αποθηκεύσουν τις δύο αιτήσεις για το ID της συναλλαγής και του merchant.
- Το place ses_No_I αποθηκεύει το session number i της συγκεκριμένης συναλλαγής έχοντας αρχική τιμή την κενή λίστα και θα προσθέτει στη λίστα το νέο ses_No_I όταν εκτελείται το transition M_PP_1.
- Το place ses_No_J αποθηκεύει το session number j της συγκεκριμένης συναλλαγής έχοντας αρχική τιμή 0.
- Το place VSResponce το οποίο θα αποθηκεύσει την απάντηση για το αν η συναλλαγή ολοκληρώθηκε με επιτυχία.
- Τα places MacedMSG και VSR αποθηκεύουν τις αντίστοιχες πληροφορίες μετά την εκτέλεση του transition M_PP_2.
- Τα places HashedMSG, Approve και VSResponce αποθηκεύουν τις αντίστοιχες πληροφορίες μετά την εκτέλεση του transition M_PP_3.
- Τέλος το place correct_ses_ID_I αποθηκεύει το σωστό session ID I παίρνοντάς το από το μήνυμα με το οποίο του απαντά ο client και όχι από την λίστα ses_No_I, όπου απλά καταγράφονται όλα τα session I για τα οποία έγινε κάποια αίτηση συναλλαγής. Το συγκεκριμένο place μας εξασφαλίζει σε περίπτωση που ο intruder παραποιήσει κάποιο μήνυμα και ξαναστείλει πάλι το μήνυμα πίσω στον merchant.

Τα τέσσερα transition του merchant περιγράφονται παρακάτω:

- Το transition M_MR αναπαριστά το δεύτερο βήμα (2) εκτέλεσης του Merchant Registration Protocol. Το transition έχει ως είσοδο το κανάλι επικοινωνίας CtoM_Out. Με την εκτέλεση του transition αποθηκεύεται όποια πληροφορία θα χρειαστεί στη συνέχεια ο merchant, δημιουργείται και αποστέλλεται το Hashed μήνυμα στο κανάλι επικοινωνίας MtoC_In ώστε να φτάσει στον client.
- Το transition M_PP_1 αναπαριστά το δεύτερο βήμα (4) εκτέλεσης του Payment Protocol. Το transition έχει ως είσοδο το κανάλι επικοινωνίας CtoM_Out. Με την εκτέλεση του transition αποθηκεύεται όποια πληροφορία θα χρειαστεί στη συνέχεια ο merchant, δημιουργείται και αποστέλλεται το κρυπτογραφημένο (4) μήνυμα στο κανάλι επικοινωνίας MtoC_In ώστε να φτάσει στον client.



Σχήμα 5. *Το μοντέλο του merchant*

- Το transition M_PP_2 αναπαριστά το τέταρτο βήμα (6) εκτέλεσης του Payment Protocol. Το transition έχει ως είσοδο το κανάλι επικοινωνίας CtoM_out. Με την εκτέλεση του transition αποθηκεύεται όποια πληροφορία θα χρειαστεί στη συνέχεια ο merchant, δημιουργείται και

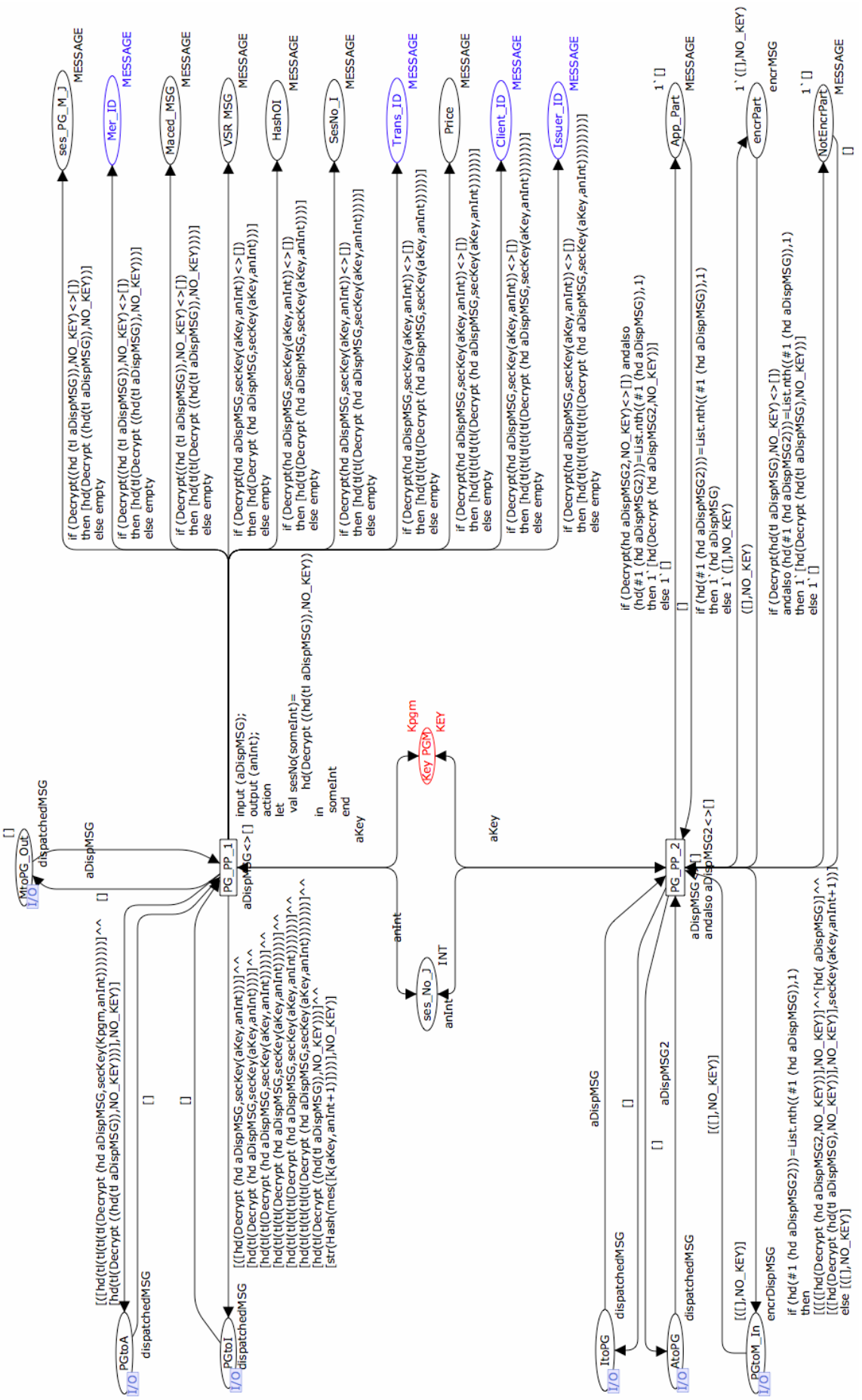
αποστέλλεται το κατάλληλο μήνυμα (6) στο κανάλι επικοινωνίας MtoPG_In ώστε να φτάσει στον payment gateway.

- Το transition M_PP_3 αναπαριστά το έννατο βήμα (11) εκτέλεσης του Payment Protocol. Το transition έχει ως είσοδο το κανάλι επικοινωνίας PGtoM_Out. Με την εκτέλεση του transition αποθηκεύεται όποια πληροφορία θα χρειαστεί ο merchant, δημιουργείται και αποστέλλεται το κατάλληλο διπλο-κρυπτογραφημένο μήνυμα (6) στο κανάλι επικοινωνίας MtoC_In ώστε να φτάσει στον client.

4.1.5 Περιγραφή του μοντέλου του Payment Gateway

Οι περιγραφή των places του PG έχει ως εξής:

- Ο PG έχει έξι places που αναπαριστούν τα κανάλια επικοινωνίας (MtoPG_Out, PGtoM_In, PGtoI, ItoPG, PGtoA, AtoPG) διότι είναι ο ενδιάμεσος ανάμεσα σε Client, Merchant, Issuer, Acquirer
- Επίσης υπάρχει ένα place (Key_PGM) κόκκινου χρώματος που αναπαριστά το κλειδί στο οποία έχει πρόσβαση ο payment gateway και δηλαδή το κλειδί το κοινόχρηστο κλειδί του payment gateway με τον merchant.
- Το place ses_No_J αποθηκεύει το session number j της συγκεκριμένης συναλλαγής και δεν έχει αρχική τιμή.
- Επίσης έχει δέκα places (ses_PG_M_J, Mer_ID, Maced_MSG, VSR MSG, HashOI, SesNo_I, Trans_ID, Price, Client_ID, Issuer_ID) τα οποία θα χρησιμεύσουν για την αποθήκευση όλης της απαιτούμενης γνώσης του payment gateway που θα προκύψει μετά την εκτέελεση του transition PG_PP_1
- Τέλος τα places (App_Part, encrPart, NotEncrPart) θα χρησιμεύσουν για την αποθήκευση όλης της απαιτούμενης γνώσης του payment gateway που θα προκύψει μετά την εκτέελεση του transition PG_PP_2



Σχήμα 6*. Το μοντέλο του payment gateway*

Τα τέσσερα transition του merchant περιγράφονται παρακάτω:

- Το transition PG_PP_1 αναπαριστά το πέμπτο και έκτο βήμα (7,8) του Payment Protocol. Το transition έχει ως είσοδο το κανάλι επικοινωνίας MtoPG_out. Με την εκτέλεση του transition αποθηκεύεται όποια πληροφορία θα χρειαστεί να κατέχει ο payment gateway , δημιουργεί και αποστέλλει το κατάλληλο μήνυμα (7,8) στο κανάλι επικοινωνίας PGtoI και PGtoA ώστε να φτάσουν τα μηνύματα στον Issuer και τον Acquirer αντίστοιχα.

- Το transition PG_PP_2 αναπαριστά το όγδοο βήμα (10) του Payment Protocol. Το transition έχει ως είσοδο τα κανάλια επικοινωνίας ItoPG και AtoPG. Με την εκτέλεση του transition αποθηκεύεται όποια πληροφορία θα χρειαστεί να κατέχει ο payment gateway , δημιουργεί και αποστέλλει το κατάλληλο μήνυμα (10) στο κανάλι επικοινωνίας PGtoM_In ώστε να φτάσει το μήνυμα στον Merchant.

4.1.6 Περιγραφή του μοντέλου του Acquirer

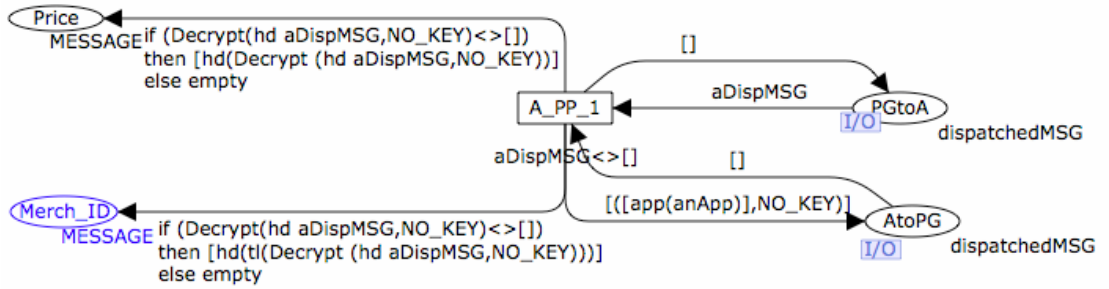
Οι περιγραφή των places του Acquirer έχει ως εξής:

- Ο Acquirer έχει δύο places (PGtoA, AtoPG) που αναπαριστούν τα κανάλια επικοινωνίας από τον Acquirer προς τον Payment Gateway και αντίστροφα.
- Ένα place (Price) το οποίο αποθηκεύει το ποσό της παραγγελίας
- Ένα place (MerchID) στο οποίο αποθηκεύεται το ID του merchant στον οποίο έγινε η πληρωμή

Ο Acquirer έχει ένα μοναδικό transition με το οποίο παίρνει στην είσοδο (PGtoA) το μήνυμα που του έστειλε ο Payment Gateway, κρατάει ότι πληροφορία χρειάζεται (Price και MerchID) και αποστέλλει το μήνυμα (9) πίσω στον Payment Gateway ώστε να τον ενημερώσει αν ολοκληρώθηκε με τον σωστό τρόπο η συναλλαγή.

Να σημειώσουμε σ'αυτό το σημείο ότι κατά την απάντηση που στέλνεται στο payment gateway υπάρχει μία τιμή app(anApp) όπου η μεταβλητή anApp

είναι του τύπου APPROVED που είναι enumerated και η τιμή της είναι είτε Yes είτε No. Κατά την εκτέλεση του transition η τιμή της μεταβλητής είναι τυχαία μία από τις δύο πιθανές. Ωστόσο κατά την δημιουργία του state space του μοντέλου μας θα παραχθεί ο γράφος που θα περιλαμβάνει και τα δύο σενάρια εκτέλεσης.



Σχήμα 7*. Το μοντέλο του Acquirer*

4.1.7 Περιγραφή του μοντέλου του Issuer

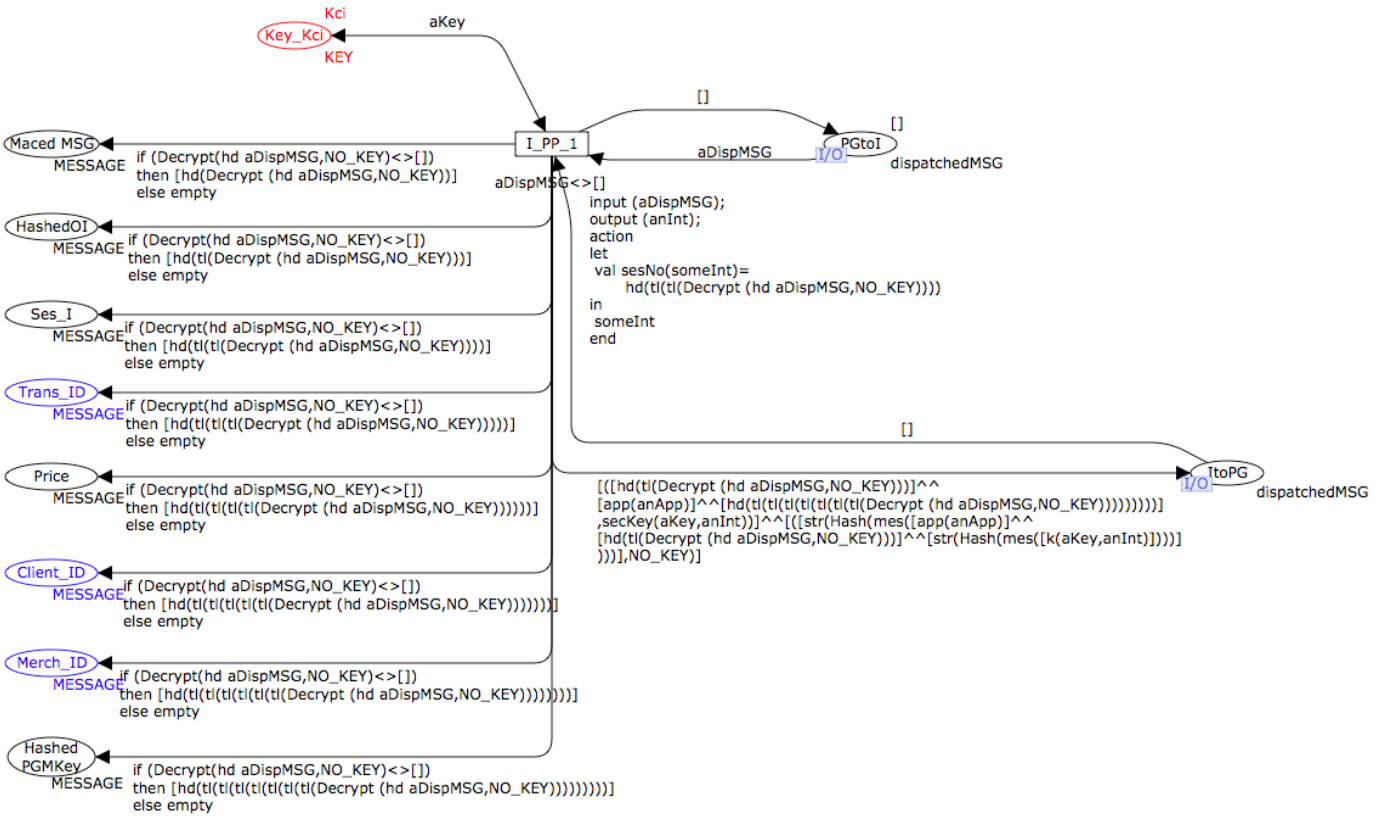
Οι περιγραφή των places του Issuer έχει ως εξής:

- Ο Issuer έχει δύο places (PGtoI, ItoPG) που αναπαριστούν τα κανάλια επικοινωνίας από τον Issuer προς τον Payment Gateway και αντίστροφα.
- Οκτώ places (Maced MSG, HashedOI, Ses_J, TransID, Price, Cliend_ID, Merch_ID, Hashed PGMKey) στα οποία αποθηκεύεται όλη η απαραίτητη πληροφορία που πρέπει να γνωρίζει ο issuer.
- Ένα place (Key_Kci) στο οποίο αποθηκεύεται το κοινόχρηστο κλειδί του issuer με τον client

Ο Issuer έχει ένα μοναδικό transition (I_PP_1) με το οποίο παίρνει στην είσοδο (PGtoI) το μήνυμα που του έστειλε ο Payment Gateway, κρατάει ότι πληροφορία χρειάζεται (Maced MSG, HashedOI, Ses_J, TransID, Price, Cliend_ID, Merch_ID, Hashed PGMKey) και αποστέλλει το μήνυμα (9) πίσω στον Payment Gateway ώστε να τον ενημερώσει αν ολοκληρώθηκε με τον σωστό τρόπο η συναλλαγή.

Να σημειώσουμε σ'αυτό το σημείο (όπως και στον Acquirer) ότι κατά την απάντηση που στέλνεται στο payment gateway υπάρχει μία τιμή app(anApp) όπου η μεταβλητή anApp είναι του τύπου APPROVED που είναι enumerated

και η τιμή της είναι είτε Yes είτε No. Κατά την εκτέλεση του transition η τιμή της μεταβλητής είναι τυχαία μία από τις δύο πιθανές. Ωστόσο κατά την δημιουργία του state space του μοντέλου μας θα παραχθεί ο γράφος που θα περιλαμβάνει και τα δύο σενάρια εκτέλεσης.



Σχήμα 8. *Το μοντέλο του Issuer*

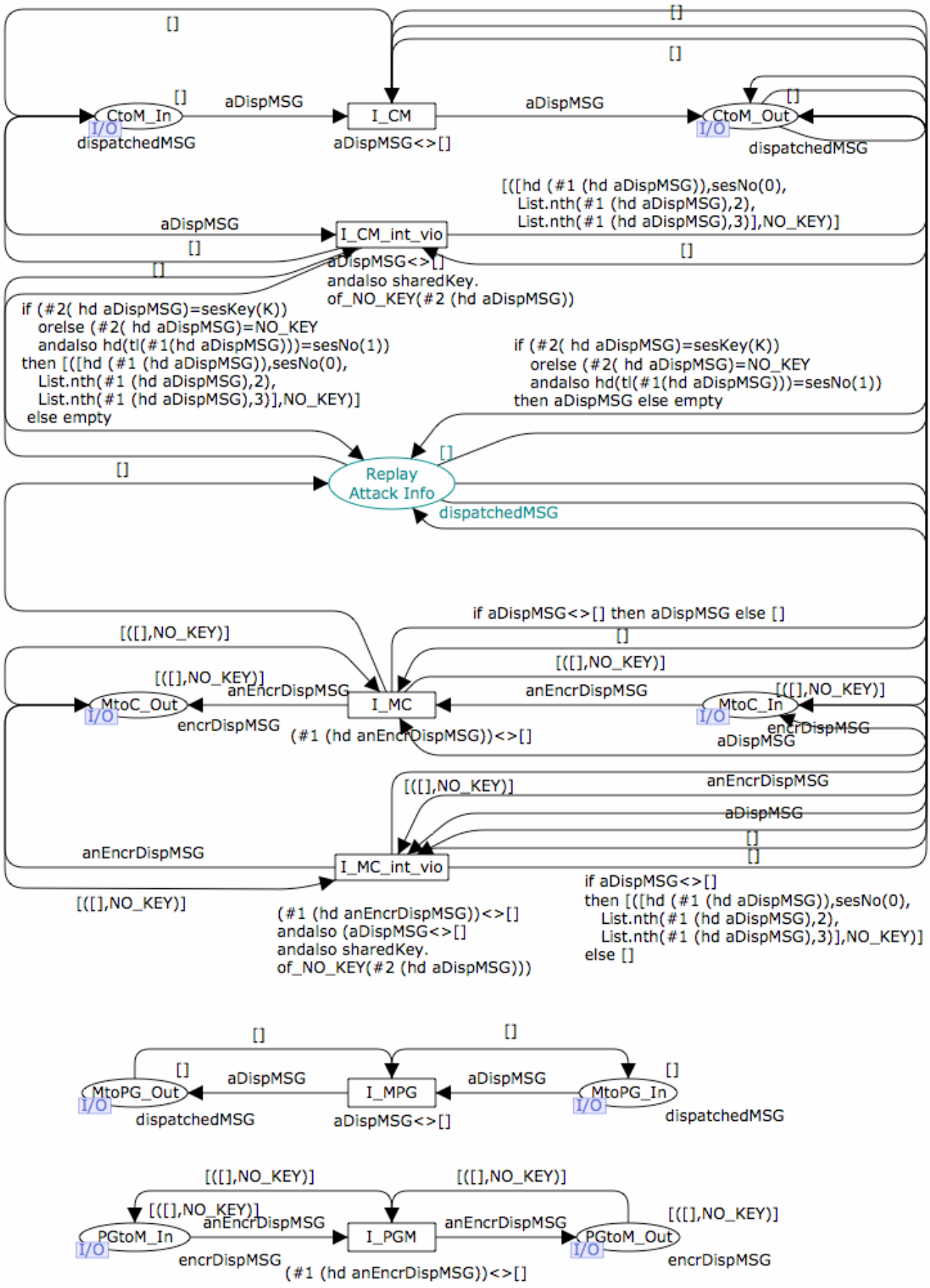
4.1.8 Περιγραφή του μοντέλου του εισβολέα

Οι περιγραφή των places του Intruder έχει ως εξής:

- Έχει 8 places (CtoM_In, CtoM_Out, MtoC_In, MtoC_Out, MtoPG_In, MtoPG_Out, PGtoM_In, PGtoM_out) που αναπαριστούν τα κανάλια επικοινωνίας του Intruder. Η βασική έννοια του intruder είναι να είναι ο ενδιάμεσος των client, merchant και payment gateway ώστε να μπορεί να υποκλέπτει όλα τα μηνύματα που παιρνούν από αυτόν.
- Υπάρχει ένα πολύ σημαντικό place (Replay Attack Info) με γαλαζοπράσινο χρώμα, το οποίο μπορεί και αποθηκεύει κάποια μηνύματα με σκοπό να τα χρησιμοποιήσει σε επόμενες μεταδόσεις (Replay Attack – Type Flaw Attack).

Οι περιγραφή των transition του Intruder έχει ως εξής:

- Έχει δύο transitions (I_CM και I_CM_vio_int) τα οποία παίρνουν τα μηνύματα από το κανάλι CtoM_In, δηλαδή από τον έξοδο του client και τα προωθούν στο κανάλι CtoM_Out. Παράλληλα με την προώθηση του μηνύματος, γίνεται και μία αντιγραφή του μηνύματος στο place Replay Attack Info με σκοπό στο επόμενο βήμα, με την εκτέλεση του transition I_MC ή I_MC_vio_int να σταλεί και πάλι το μήνυμα (Replay Attack – Type Flaw Attack). Η διαφορά των δύο είναι ότι, στην περίπτωση που το μήνυμα είναι μη κρυπτογραφημένο, το transition I_CM_vio_int κάνει μία επιπλέον λειτουργία, παραποιεί και αλλάζει το περιεχόμενο του μηνύματος. Συγκεκριμένα στέλνει το ίδιο μήνυμα με αλλαγμένο το session number (violation integrity). Να σημειώσουμε σ'αυτό το σημείο ότι και τα δύο transition είναι ταυτόχρονα ενεργά προς εκτέλεση, αλλά μόνο ένα θα εκτελεστεί. Ωστόσο κατά την δημιουργία του state space του μοντέλου μας θα παραχθεί ο γράφος που θα περιλαμβάνει και τα δύο σενάρια εκτέλεσης.
- Τα δύο transitions (I_MC και I_MC_vio_int) τα οποία παίρνουν τα μηνύματα από το κανάλι MtoC_In, δηλαδή από τον έξοδο του merchant και τα προωθούν στο κανάλι MtoC_Out. Παράλληλα με την προώθηση του μηνύματος, γίνεται η εξής λειτουργία: στέλνεται το μήνυμα που είναι αποθηκευμένο στο place (Replay Attack Info) προς τον merchant ως μία δεύτερη αποστολή στο κανάλι επικοινωνίας CtoM_Out (Replay Attack – Type Flaw Attack). Η διαφορά των δύο είναι ότι, στην περίπτωση που το μήνυμα είναι μη κρυπτογραφημένο, το transition I_CM_vio_int κάνει μία επιπλέον λειτουργία, παραποιεί και αλλάζει το περιεχόμενο του μηνύματος. Συγκεκριμένα στέλνει το ίδιο μήνυμα με αλλαγμένο το session number (violation integrity). Να σημειώσουμε σ'αυτό το σημείο ότι και τα δύο transition είναι ταυτόχρονα ενεργά προς εκτέλεση, αλλά μόνο ένα θα εκτελεστεί. Ωστόσο κατά την δημιουργία του state space του μοντέλου μας θα παραχθεί ο γράφος που θα περιλαμβάνει και τα δύο σενάρια εκτέλεσης.



Σχήμα 9. *Το μοντέλο του Intruder*

- Τα δύο τελευταία transitions (I_MPG και I_PGM) απλά, το πρώτο transition παίρνει τα μηνύματα από το κανάλι MtoPG_In, δηλαδή από τον έξοδο του merchant και τα προωθούν στο κανάλι MtoPG_Out και το δεύτερο transition παίρνει τα μηνύματα από το κανάλι PGtoM_In, δηλαδή από τον έξοδο του payment gateway και τα προωθούν στο κανάλι PGtoM_Out

4.2 Ανάλυση χώρου καταστάσεων

Στην παράγραφο αυτή θα πραγματοποιήσουμε μία εκτενής ανάλυση του χώρου των καταστάσεων που παράγει το μοντέλο μας. Θα εξετάσουμε ένα σύνολο ιδιοτήτων που θα πρέπει να πληρεί το μοντέλο μας, όπως:

a) Την απουσία self-loop terminal markings
b) Τον σωστό τερματισμό του πρωτοκόλλου και την απουσία deadlocks
c) Την απουσία livelocks
d) Την ασφαλή μετάδοση των μηνυμάτων

Στον πίνακα 6 παρουσιάζεται το αποτέλεσμα της αναφοράς του χώρου καταστάσεων του μοντέλου του CP-Net του πρωτοκόλλου.

Το πρώτο μέρος της αναφοράς, παρέχει κάποια στατιστικά στοιχεία που αφορούν τον γράφο του χώρου καταστάσεων του μοντέλου ή όπως αλλιώς είναι γνωστός αυτός οι γράφος, state space graph ή occurrence graph. Το χρωματισμένο δίκτυο Petri που προέκυψε, αποτελείται από 107 κόμβους και 165 τόξα, τα οποία αναπαριστούν όλες τις διαφορετικές καταστάσεις στις οποίες μπορεί να βρεθεί το μοντέλο μας. Μία απεικόνιση του γράφου αυτού είναι ο γράφος του σχήματος 10 Ο αντίστοιχος γράφος των ισχυρά συνδεδεμένων κόμβων (Scc graph) αποτελείται και αυτός από 107 κόμβους και 165 τόξα.

Το δεύτερο μέρος της αναφοράς, μας δίνει πληροφορίες σχετικά με την ιδιότητα “Home properties” του μοντέλου. Η ιδιότητα αυτή ελέγχει εάν το μοντέλο μας έχει transition τα οποία είναι συνεχώς εκτελέσιμα (ενεργά). Ωστόσο, όπως θα περιμέναμε το μοντέλο μας δεν έχει κανένα home marking,

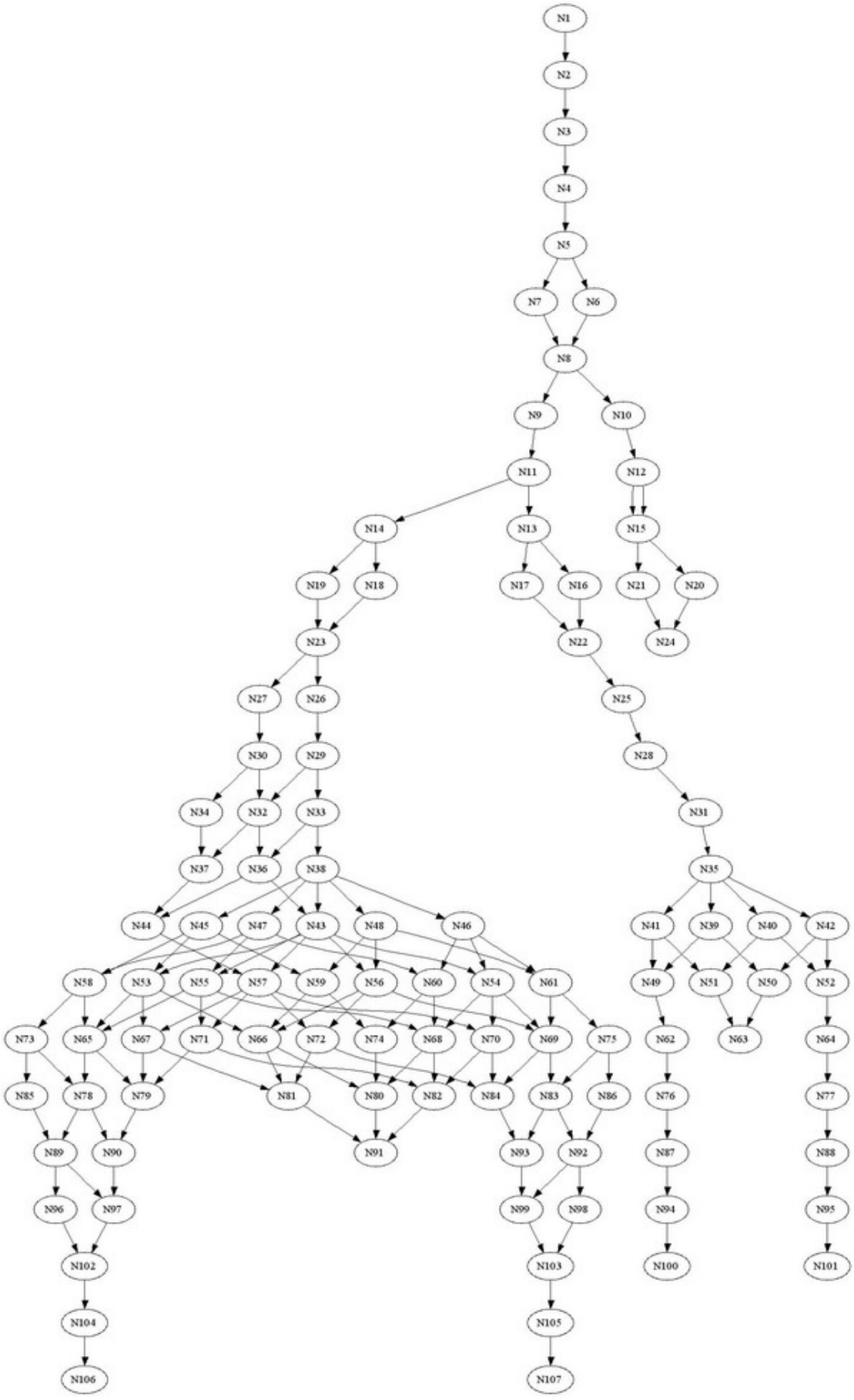
κάτι που μας δίνει την δυνατότητα να πούμε ότι το μοντέλο μας είναι σωστά σχεδισμένο.

```
Statistics
------------------------------------------------------------
  State Space
    Nodes:  107
    Arcs:   165
    Secs:   1
    Status: Full

  Scc Graph
    Nodes:  107
    Arcs:   165
    Secs:   0

 Home Properties
------------------------------------------------------------
  Home Markings:  None


 Liveness Properties
------------------------------------------------------------
  Dead Markings:  7 [91,63,24,107,106,...]
  Dead Transitions Instances: None
  Live Transitions Instances: None


 Fairness Properties
------------------------------------------------------------
  No infinite occurrence sequences.
```

Πίνακας 6. *Αναφορά της ανάλυσης χώρου καταστάσεων του μοντέλου*

Στο τρίτο μέρος της αναφοράς, εξετάζεται η ιδιότητα Liveness properties η οποία παρέχει πληροφορίες σχετικά με:

a) Τον αριθμό των dead markings, δηλαδή markings χωρίς εκτελέσιμο transition. Τα dead markings μπορεί να είναι είτε τελικές καταστάσεις του πρωτοκόλλου είτε deadlocks, οπότε χρειάζεται περαιτέρω ανάλυση.
b) Το πλήθος των dead transition, όπου είναι τα transition που δεν είναι εκτελέσιμα τουλάχιστον σε μία από τις καταστάσεις του μοντέλου.
c) Τον αριθμό των live transition, δηλαδή transition τα οποία είναι συνεχώς ενεργά.



Σχήμα 10*. Ο γράφος των καταστάσεων του μοντέλου*

Όπως θα περιμέναμε το χρωματισμένο δίκτυο Petri που δημιουργήσαμε δεν περιλαμβάνει dead και live transition, αλλά απαιτείται να ελέγξουμε το μοντέλο μας με περισσότερες λεπτομέρειες για πιθανά deadlocks και σωστό τερματισμό του πρωτοκόλλου. Στον παρακάτω πίνακα απαριθμούνται όλα τα dead markings του μοντέλου μας με την βοήθεια του κατάλληλου ερωτήματος.

let
  val fid = TextIO.openOut "ListOfDeadMarkings.txt"
  val _ = TextIO.output(fid, "List of dead markings: \n")
  val _ = EvalNodes(ListDeadMarkings(),
  fn n => INT.output(fid,n))
  val _ = TextIO.output(fid, "\nNumber of dead markings: ")
  val _ = INT.output(fid,length (ListDeadMarkings()))
in
  TextIO.closeOut(fid)
end
	List of dead markings:
	91 63 24 107 106 101 100
	Number of dead markings: 7

Πίνακας 7. *Τα Dead Markings του μοντέλου μας*

Το τελευταίο μέρος της βασικής αναφοράς (Πίνακας 6) αναφέρεται στην ιδιότητα fairness properties και από ότι παρατηρούμε δεν υπάρχουν infinite occurrence sequences στον γράφο μας.

Όπως αναφέραμε και παραπάνω θα πρέπει να ελέγξουμε στον σωστό τερματισμό του πρωτοκόλλου. Στον πίνακα 8 φαίνεται το ερώτημα με το οποίο ελέγχετε ο γράφος μας για πιθανά self-loops. Αυτό σημαίνει ότι όλα τα όλες οι τελικές καταστάσεις του πρωτοκόλλου μας πρέπει να συμπεριλαμβάνονται μέσα στην λίστα των dead markings του πίνακα 7.

fun SelfLoopTerminal n=(OutNodes(n)=[n])
fun InValidTerminal()=PredNodes(EntireGraph,
  fn n => (SelfLoopTerminal n),
  NoLimit);
let
  val fid = TextIO.openOut "ListOfSelfLoops.txt"
  val _ = TextIO.output(fid, "List of self loop terminals: \n")
  val _ = EvalNodes(InValidTerminal(),
  fn n => INT.output(fid,n))
in
  TextIO.closeOut(fid)
end
	List of self loop terminals:

Πίνακας 8. *Απουσία self-loop από το μοντέλο μας*

Στον πίνακα 9 παραθέτουμε το ερώτημα με το οποίο ελέγχεται το μοντέλο μας για deadlock markings. Με το ερώτημα αυτό αποδεικνύουμε ότι όλες οι τελικές καταστάσεις του πρωτοκόλλου (dead markings) είναι σωστές, δηλαδή ότι όλα τα κανάλια επικοινωνίας είναι άδεια και παράλληλα όλοι οι συμβαλλόμενοι της συναλλαγής βρίσκονται σε νόμιμη κατάσταση.

```
fun ValidTerminal n=(
      (Mark.TopLevel'CtoM_In 1 n)=1`[] andalso
      (Mark.TopLevel'CtoM_Out 1 n)=1`[] andalso
      (Mark.TopLevel'MtoC_Out 1 n)=1`[([],NO_KEY)] andalso
      (Mark.TopLevel'MtoC_In 1 n)=1`[([],NO_KEY)] andalso
      (Mark.TopLevel'MtoPG_In 1 n)=1`[] andalso
      (Mark.TopLevel'PGtoM_Out 1 n)=1`[([],NO_KEY)] andalso
      (Mark.TopLevel'MtoPG_Out 1 n)=1`[] andalso
      (Mark.TopLevel'PGtoM_In 1 n)=1`[([],NO_KEY)] andalso
      (Mark.TopLevel'PGtoI 1 n)=1`[] andalso
      (Mark.TopLevel'ItoPG 1 n)=1`[] andalso
      (Mark.TopLevel'PGtoA 1 n)=1`[] andalso
      (Mark.TopLevel'AtoPG 1 n)=1`[] andalso

      (((Mark.Client'VSResponce 1 n)=1`[str("mes([pid(tID),str(\"mes([od(\"Order Desc\")])\"),pr(\"Price\")])"),
app(Yes),str("mes([k((Kpgm,1))])")] andalso
      ((Mark.Merchant'Approve 1 n)=1`[app(Yes)] ) andalso
      ((Mark.PG'App_Part 1 n)=1`[app(Yes)]))

      orelse

      ((Mark.Client'VSResponce 1 n)=1`[str("mes([pid(tID),str(\"mes([od(\"Order Desc\")])\"),pr(\"Price\")])"),
app(No),str("mes([k((Kpgm,1))])")] andalso
      ((Mark.Merchant'Approve 1 n)=1`[app(No)]) andalso
      ((Mark.PG'App_Part 1 n)=1`[app(No)])

      orelse

      ((Mark.Client'VSResponce 1 n)=1`[] andalso
      ((Mark.Merchant'ses_ID_I 1 n)<> [])))




fun InValidTerminal()=PredNodes(ListDeadMarkings(),
                                fn n => not (ValidTerminal n),
                                NoLimit);
let
  val fid = TextIO.openOut "DeadlockMarkings.txt"
  val _ = TextIO.output(fid, "List of deadlock markings: \n")
  val _ = EvalNodes(InValidTerminal(),
          fn n => INT.output(fid,n) )
in
  TextIO.closeOut(fid)
end
```

Πίνακας 9. *Απουσία deadlock από το μοντέλο μας*

Πιο αναλυτικά, υπάρχουν οι εξής πιθανές αποδεκτές τελικές καταστάσεις του μοντέλου μας:

- Να γίνει επιτυχής συναλλαγή, οπότε ο Issuer και ο Acquirer θα απαντήσουν καταφατικά για την εξέλιξη της συναλλαγής (Yes). Αυτή η κατάσταση αναπαριστάται με τους κόμβους N101 και N107 του σχήματος 10.
- Να μην γίνει επιτυχής συναλλαγή, οπότε ο Issuer και ο Acquirer θα απαντήσουν αρνητικά για την εξέλιξη της συναλλαγής (No). Αυτή η κατάσταση αναπαριστάται με τους κόμβους N100 και N106 του σχήματος 10.
- Να διακοπεί η συναλλαγή λόγω ανίχνευσης επίθεσης οπότε τα περιεχόμενα των αντίστοιχων places των Client, Issuer και Acquirer θα είναι κενά (κόμβοι 24) είτε η απάντηση των Issuer και Acquirer θα είναι διαφορετική οπότε δεν θα προχώρησε η συναλλαγή (κόμβοι 63 και 91).

Τέλος στον πίνακα 10 ελέγχουμε το μοντέλο μας για livelocks. Όταν στον χώρο καταστάσεων υπάρχουν κύκλοι που δεν συνδέονται με άλλους κόμβους εκτός του κύκλου, τότε υπάρχει livelock. Σε αυτή την περίπτωση εάν η εκτέλεση οδηγηθεί μέσα στον κύκλο αυτό, θα παραμείνει συνεχώς εγκλωβισμένη μέσα στον κύκλο αυτό. Από ότι διαπιστώσαμε το μοντέλο μας δεν περιέχει livelocks.

fun ListTerminalSCCs()=PredAllSccs(SccTerminal); fun InValidTermSCC()=PredSccs(ListTerminalSCCs(), fn n => not (SccTrivial n), NoLimit); let val fid = TextIO.openOut "ListOfLivelocks.txt" val _ = if InValidTermSCC()=[] then TextIO.output(fid, "No Livelocks!") else TextIO.output(fid, "Livelocks detected!") in TextIO.closeOut(fid) end	No Livelocks!

Πίνακας 10. *Απουσία livelock από το μοντέλο μας*

Στη συνέχεια μπορούμε να προχωρήσουμε σε περαιτέρω ανάλυση, για να ελέγξουμε πιθανά προβλήματα ασφαλείας του πρωτοκόλλου. Με την βοήθεια του παρακάτω non-standard ερωτήματος, ελέγχουμε εάν πραγματοποιήθηκε μία πληρωμή ενώ παράλληλα τα session number που χρησιμοποιήθηκαν από τον Client και τον Merchant αντίστοιχα, ήταν διαφορετικά. Από ότι παρατηρούμε δεν προέκυψε τέτοιου είδους πρόβλημα.

```
fun SessionNumberChecking n=
(sesNo(hd(Mark.Client'ses_No_I 1 n)) <>(hd(Mark.Merchant'correct_ses_ID_I 1 n)));

fun PaymentCompleted n=
(Mark.Client'VSResponce 1 n)= 1`[str("mes([pid(tID),str(\"mes([od(\"Order Desc\")] )\"),
pr(\"Price\")])"),app(Yes),str("mes([k((Kpgm,1))])")]

orelse

(Mark.Client'VSResponce 1 n)=1`[str("mes([pid(tID),str(\"mes([od(\"Order Desc\")] )\"),
pr(\"Price\")])"),app(No),str("mes([k((Kpgm,1))])")];

val trustSecurityofSessionNumbers = PredNodes(EntireGraph,
                        fn n => ((SessionNumberChecking n) andalso (PaymentCompleted n)),
                        NoLimit);

let
  val fid = TextIO.openOut "SecurityofSessionNumbers.txt"
  val _ = if (trustSecurityofSessionNumbers = [])
            then TextIO.output(fid,"No Session Number Violation!")
            else TextIO.output(fid,"Session Number Violation detected!")
in
  TextIO.closeOut(fid)
end
```

```
No Session Number
Violation!
```

Πίνακας 11. *Πρώτος έλεγχος ασφαλείας του μοντέλου μας*

Ο επόμενος έλεγχος που κάνουμε είναι να ελέγχουμε αν υπάρχει περίπτωση να πληρωθεί μία συναλλαγή από τον Client μέσω του Issuer και να μην παραλάβει ο Merchant τα χρήματα αυτά από τον Acquirer. Επίσης ελέγχουμε και το αντίστροφο, δηλαδή να μην πληρωθεί μία συναλλαγή από τον Client μέσω του Issuer και να παραλάβει ο Merchant τα χρήματα από τον Acquirer. Η παραπάνω ανάλυση μεταφράζεται στο μοντέλο μας ως εξής:

- Πρέπει ο Issuer να απαντήσει καταφατικά στον Payment Gateway (Yes) και ο Acquirer αρνητικά (No)
- Και αντίστροφα, ο Issuer να απαντήσει αρνητικά στον Payment Gateway (No) και ο Acquirer καταφατικά (Yes)

fun PaymentTrast n= (Mark.PG'App_Part 1 n)=1`[app(Yes)] andalso (Mark.PG'encrPart 1 n)=1`([str("mes([pid(tID),str(\"mes([od(\"Order Desc\")])\"),pr(\"Price\")])"),app(No),str("mes([k((Kpgm,1))])")],secKey((Kci,1))) orelse (Mark.PG'App_Part 1 n)=1`[app(No)] andalso (Mark.PG'encrPart 1 n)=1`([str("mes([pid(tID),str(\"mes([od(\"Order Desc\")])\"),pr(\"Price\")])"),app(Yes),str("mes([k((Kpgm,1))])")],secKey((Kci,1))); val trustofPayment = PredNodes(EntireGraph, fn n => (PaymentTrast n), NoLimit); let val fid = TextIO.openOut " PaymentTrast.txt" val _ = if (trustofPayment = []) then TextIO.output(fid,"There is trust on Payment!") else TextIO.output(fid,"There is no trust on Payment!") in TextIO.closeOut(fid) end	There is trust on Payment!

Πίνακας 12. *Δεύτερος έλεγχος ασφαλείας του μοντέλου μας*

Στον πίνακα 12 παρουσιάζεται το ερώτημα αυτό και η έξοδός του. Πράγματι το πρωτόκολλο δεν έχει πρόβλημα ασφαλείας στην περίπτωση αυτή.

4.3 Κριτική θεώρηση των αποτελεσμάτων της ανάλυσης (σχολιασμός αποτελεσμάτων, προβλήματα, τεχνικές αντιμετώπισης των προβλημάτων της ανάλυσης)

Από την παραπάνω ανάλυση της παραγράφου 4.2 προκύπτει ότι το μοντέλο του πρωτοκόλλου μας δεν περιέχει:

e) self-loop terminal markings
f) deadlocks
g) livelocks

Το γεγονός αυτό μας δίνει την δυνατότητα να ισχυριστούμε την σωστή λειτουργία του πρωτοκόλλου. Επίσης στην συνέχεια αποδείξαμε ότι το πρωτόκολλό μας πληρεί δύο ελέγχους σε ιδιότητες ασφαλείας. Ωστόσο

υπάρχουν πολλών ειδών έλεγχοι που θα μπορούσαμε να κάνουμε με σκοπό να ελέγξουμε την ασφάλεια του συγκεκριμένου πρωτοκόλλου. Κάτι τέτοιο όμως είναι εκτός του σκοπού της εργασίας αυτής. Στο [21] οι συγγραφείς του πρωτοκόλλου μας αποδεικνύουν με μαθηματικό τρόπο τις απαραίτητες ιδιότητες ασφαλείας που πρέπει να τηρεί ένα πρωτόκολλο ασφαλών ηλεκτρονικών πληρωμών μέσω κινητών συσκευών. Ωστόσο εμείς προτείναμε έναν άλλο τρόπο μοντελοποίησης – ανάλυσης πρωτοκόλλων αυτού του είδους.

Στο χρωματισμένο δίκτυο Petri που προτείναμε ακολουθήσαμε την εξής τακτική: χρήση όσο το δυνατόν λιγότερων transition. Αυτό διότι αν το μοντέλο μας αποτελείται από πολλά transitions, γίνεται όλο και δυσκολότερη η ανάλυση του πρωτοκόλλου και πολλές φορές μάλιστα είναι αδύνατη. Έτσι αντί να χρησιμοποιούμε ένα transition για να δέχεται τα tokens από τα διάφορα places, ένα για να αποθηκεύει την απαραίτητη πληροφορία, κάποια άλλα για να δημιουργήσουν τα προς αποστολή μηνύματα και ένα για να αποστείλει το μήνυμα στο κατάλληλο κανάλι επικοινωνίας, κάναμε χρήση ενός transition για κάθε τέτοια περίπτωση, όπου ταυτόχρονα δέχεται, αποθηκεύει, συντάσσει και αποστέλλει μηνύματα. Το αρνητικό της μεθόδου αυτής είναι ότι τα τόξα του μοντέλου μας συνοδεύονται από μεγάλες προτάσεις για να επιτευχθούν όλοι οι απαραίτητοι έλεγχοι και να συνταχθούν τα μηνύματα του πρωτοκόλλου. Να σημειωθεί ότι το συγκεκριμένο πρωτόκολλο συμπεριλαμβάνει μεγάλα σε μέγεθος μηνύματα, κάτι που επιβαρύνει την γραφική αναπαράσταση του μοντέλου μας. Ωστόσο, αυτή η τεχνική μας βοήθησε και καταλήξαμε σε ένας γράφο με 107 κόμβους (σχετικά μικρός) ενώ σε κάθε άλλη περίπτωση ο γράφος μας θα ήταν πολύ μεγαλύτερος και ίσως δεν θα μπορούσαμε να τον αναπαραστήσουμε γραφικά όπως στο σχήμα 10.

Επίσης σε αυτό το σημείο θα πρέπει να αναφέρουμε ότι χρησιμοποιήσαμε τις βιβλιοθήκες που είναι απαραίτητες ώστε να παραχθεί ο γράφος του σχήματος 10 και με την βοήθεια του εργαλείου Graphviz καταφέραμε να αναπαραστήσουμε τον γράφο του μοντέλου μας. Οι εντολές ενσωμάτωσης λειτουργιών γράφων στα χρωματισμένα δίκτυα Petri είναι αυτές του πίνακα 13.

```
use (ogpath^"export/OGtoGraphviz.sml");
open OGtoGraphviz;
```

Πίνακας 13. *Εντολές παραγωγής γράφου*

Κεφάλαιο 5

Επίλογος

5 Επίλογος

5.1 Αναφορά σε άλλες τεχνικές τυπικής ανάλυσης πρωτοκόλλων ασφάλειας

Στην εργασία αυτή χρησιμοποιήσαμε το εργαλείο CPN Tool για να αναπαραστήσουμε και να αναλύσουμε ένα ασφαλές πρωτόκολλο πληρωμών μέσω κινητών συσκευών με χρωματισμένα δίκτυα Petri. Ωστόσο υπάρχει μία πληθώρα εργαλείων για τυπική ανάλυση συστημάτων – πρωτοκόλλων και το καθένα έχει τα πλεονεκτήματα και τα μειονεκτήματα του. Μία σύντομη λίστα τέτοιων εργαλείων παρατίθεται παρακάτω:

- Το εργαλείο SPIN το οποίο χρησιμοποιείται στην ανάλυση λογικών αποτελεσμάτων από σύγχρονα συστήματα και ιδιαίτερα πρωτοκόλλων επικοινωνίας [22]. Χρησιμοποιείται η γλώσσα προγραμματισμού PROMELA (Process MEta LAnguage) για να περιγράψει την συμπεριφορά και την σύνδεση μεταξύ διάφορετικών λειτουργιών.
- CADP (CÆSAR – ALDÉBARAN Development Package) [23] το οποίο είναι ένα εργαλείο επαλήθευσης σχεδιασμού πρωτοκόλλων επικοινωνίας και κατανεμημένων συστημάτων, χρησιμοποιόντας την γλώσσα προγραμματισμού LOTOS (Language of Temporal Ordering Specifications).
- Το Concurrency Workbench [24] είναι ένα εργαλείο που συνδιάζει πολλές στρατηγικές επαλήθευσης συστημάτων.
- Cospan [25] το οποίο χρησιμοποιεί την θεωρία αυτομάτων για να κάνει επαλήθευση συστημάτων. Η γλώσσα που χρησιμοποιείται είναι κατά βάση η S/R (Selection/Resolution) αλλά και η Vérilog και η VHDL (Very high speed integrated circuit Hardware Description Language)
- Το FC2TOOLS (the next-generation AUTO/GRAPH) [26] είναι ένα εργαλείο επαλήθευσης που υποστηρίζει γραφική αναπαράσταση συστημάτων.

Παρόλη την πληθώρα των εργαλείων τυπικής ανάλυσης, εμείς επικεντρωθήκαμε στο CPN Tool διότι όπως αναφέραμε στην παράγραφο 3.3 τα χρωματισμένα δίκτυα Petri παρουσιάζουν πολλά πλεονεκτήματα στην τυπική ανάλυση πρωτοκόλλων.

Κεφάλαιο 6

Βιβλιογραφία - Αναφορές

6 Βιβλιογραφία – Αναφορές

[1] Amir Herzberg, Payments and banking with mobile personal devices. CACM 46(5): 53-58 (2003)

[2] Dr. Menezes, Submitted By Christopher C. Lamb Jack Pullikottil Jim Tanzola M-Commerce Payment Systems : Architectures and Protocols

[3] Konrad Wrona, Marko Schuba, Guido Zavagli, Mobile Payments - State of the Art and Open Problems

[4] Wrona, K. Zavagli, G. (1999). Adaptation of the SET Protocol to Mobile Networks and to the Wireless Application Protocol. European Wireless 1999 Conference, Munich.

[5] "Υπηρεσίες Προστιθέμενης Αξίας στο Διαδίκτυο", Α. Πομπόρτσης, Α. Μήλιου, Εκδόσεις Τζιόλα 2004

[6] ISO. IS7498-2, [ISO88] "Basic Reference Model for Open Systems Interconnection, Part 2: Security Architecture". International Organisation for Standardisation, 1988.

[7] P. Ryan and S. Schneider. The modeling and Analysis of Security Protocols: the CSP Approach. Addison – Wesley, 2001

[8] Issam Al-Azzoni, B.Eng. The Verification Of Cryptographic Protocols Using Coloured Petri Nets. McMaster University 2004

[9] Katsaros, P., A roadmap to transaction guarantees in electronic payments and a Colored Petri Net model checking approach. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

[10] Heintze, N., Tygar , J., Wing, J., Wong, H., Model checking electronic commerce protocols, Proceedings of the 2nd USENIX Workshop in

Electronic Commerce, Oakland, CA, USENIX Association, California, 1996; 146-164

[11] O' Mahony, D., Peirce, M., Tewari, H. *Electronic payment systems for e-commerce* (Second Edition). Artech House, 2001.

[12] Lyon, J., Evans, K., Klein, J. Transaction Internet Protocol, Version 3.0, Network Working Group, Request for Comments (RFC 2371), July 1998. http://www.ietf.org/rfc/rfc2371.txt

[13] Asokan, N. Fairness in Electronic Commerce, PhD Thesis, University of Waterloo, Ontario, Canada, 1998 (fairness)

[14] Franklin, M., Reiter, M., Fair exchange with a semi-trusted third party, Proceedings of the 4th ACM Conference on Computer and Communication Security, 1997; 1-6.

[15] Schuldt H., Popovici, A., Schek, H.-J. Automatic generation of reliable e-commerce payment processes. Proceedings of the First International Conference on Web Information Systems Engineering (WISE 00), Vol. 1, IEEE Computer Society, 2000; 434-441 (distributed payment atomicity)

[16] Wong, H. L. Protecting individuals' interests in Electronic Commerce Protocols, PhD Thesis, CMU-CS-00-160, School of Computer Science. Carnegie Mellon University, Pittsburgh, 2000 (theorem proving approach – NetBill – protecting participants' interests)

[17] Chun Ouyang. Formal Specification and Verification of the Internet Openn Trading Protocol using Coloured Petri Nets. University of South Australia

[18] CPN Tools, http://wiki.daimi.au.dk/cpntools/cpntools.wiki

[19] S.Gritzalis, D.Spinellis. Cryptographic Protocols over Open Dostributed Systems: A Taxonomy of Flaws and related Protocol Analysis Tools

[20] S. Kungpisdan, B. Srinivasan, and Phu Dung Le, A Secure Account-Based Mobile Payment Protocol, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), University of Monash, Australia

[21] S. Kungpisdan, B. Srinivasan, and Phu Dung Le, Accountability Logic for Mobile Payment Protocol, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), University of Monash, Australia

[22] Holzmann G.J., "The SPIN Model Checker: Primer and Reference Manual" Addison-Wesley, ISBN 0-321-22862-6, 2003.

[23] Fernandez J.C., Garavel H., Kerbrat A., Mateesku R., Mounier L., Sighireanu M., "CADP: A Protocol Validation and Verification Toolbox". Proceedings of the 8th International Conference on Computer-Aided Verification (CAV'96), New Brunswick, NJ, USA, 1996. From Lecture Notes in Computer Science, pages 437-440, R. Alur and T. A. Henzinger, Eds., 2005.

[24] Cleaveland R., Parrow J., Steffen B., "The Concurrency Workbench: A Semantics-Based Tool for the Verification of Concurrent Systems". ACM Transactions on Programming Languages and Systems, vol. 15(1), pages 36-72, 1993.

[25] Hardin R. H., Har'El Z., Kurshan R. P., "COSPAN,". Proceedings of the 8th International Conference on Computer Aided Verification (CAV'96), USA, 1996. Lecture Notes in Computer Science 1102, pages 423-427, R. Alur , T. A. Henzinger, Eds, 2005.

[26] Bouali A., Ressouche A., Roy V., Simone R. d., “The FC2TOOLS Set”. Proceedings of the 8th International Conference on Computer-Aided Verification (CAV'96), USA, 1996. From Lecture Notes in Computer Science, pages 441-445, R. Alur and T. A. Henzinger, Eds, 2005.

[27] Katsaros, P,. On the design of access control to prevent sensitive information leakage in distributed object systems: a Colored Petri Net based model, In proccedings of the International Symposium on Distributed Objects and Applications (DOA 2005), Agia Napa, Cyprus, Lecture Notes in Computer Science 3761, Springer Verlag, 941-959, 2005.

[28] Katsaros, P., Odontidis, V. and Gousidou-Koutita, M., Simulation and verification of atomicity properties for an electronic cash system, In Proceedings of the 2005 European Simulation and Modelling Conference (ESM 2005), EUROSIS, Porto, Portugal, 558-563, 2005.

[29] Katsaros, P., Odontidis, V. and Gousidou-Koutita, M., Colored Petri Nets based model checking and failure analysis for E-commerce protocols, In Proceedings of the Sixth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools (CPN’05), DAIMI PB-576, Dept of Computer Science, Univeristy of Aarhus, Denmark, 267-283, 2005.