ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΤΥΠΙΚΩΝ ΜΕΘΟΔΩΝ ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΠΡΩΤΟΚΟΛΛΩΝ ΑΣΦΑΛΕΙΑΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΣΤΥΛΙΑΝΟΣ Δ. ΜΠΑΣΑΓΙΑΝΝΗΣ

ΕΠΙΒΛΕΠΩΝ: ΚΑΘ. ΑΝΔΡΕΑΣ ΠΟΜΠΟΡΤΣΗΣ

ΘΕΣΣΑΛΟΝΙΚΗ 2010

Στους γονείς μου Δημήτρη και Μαρία,

στα αδέρφια μου Χρήστο, Θανάση και Αγγελική

«ΑΝΑΠΤΥΞΗ ΤΥΠΙΚΩΝ ΜΕΘΟΔΩΝ ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΠΡΩΤΟΚΟΛΛΩΝ ΑΣΦΑΛΕΙΑΣ»

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΣΤΥΛΙΑΝΟΣ Δ. ΜΠΑΣΑΓΙΑΝΝΗΣ

Υποβλήθηκε στο Τμήμα Πληροφορικής
του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Ανδρέας Πομπόρτσης,
Καθ. Πληροφορικής ΑΠΘ
(Επιβλέπων)

Γεώργιος Παπαδημητρίου,
Αν. Καθ. Πληροφορικής ΑΠΘ
(Μέλος Συμβ. Επιτροπής)

Παναγιώτης Κατσαρός,
Λέκτορας Πληροφορικής, ΑΠΘ
(Μέλος Συμβ. Επιτροπής)

Κωνσταντίνος Καρανίκας,
Καθ. Πληροφορικής ΑΠΘ

Ελένη Καρατζά,
Καθ. Πληροφορικής ΑΠΘ

Γεώργιος Πάγκαλος,
Καθ. Ηλεκτρολόγων Μηχανικών
& Μηχανικών Η/Υ, ΑΠΘ

Βασίλειος Κάτος,
Επίκουρος Καθ. Ηλεκτρολόγων
Μηχανικών Η/Υ ΔΠΘ

«ΑΝΑΠΤΥΞΗ ΤΥΠΙΚΩΝ ΜΕΘΟΔΩΝ ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΠΡΩΤΟΚΟΛΛΩΝ ΑΣΦΑΛΕΙΑΣ»

Η έγκριση της παρούσης διδακτορικής διατριβής από το Τμήμα Πληροφορικής δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέως (Ν. 5343/1932,άρθρο 202,παρ. 2)

Ευχαριστίες

Η παρούσα διδακτορική διατριβή είναι το αποτέλεσμα ερευνητικής εργασίας που πραγματοποιήθηκε τα τελευταία χρόνια στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, στο τμήμα Πληροφορικής. Σε αυτή την ενότητα θα ήθελα να εκφράσω τις ευχαριστίες μου σε όσους συνέβαλλαν, με διαφορετικό τρόπο ο καθένας στην προσπάθεια αυτή.

Θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου στον επιβλέποντα της διδακτορικής μου διατριβής, καθηγητή κ. Ανδρέα Πομπόρτση για την πολύτιμη επιστημονική καθοδήγηση και εμπιστοσύνη που μου παρείχε καθ' όλη την διάρκεια της διδακτορικής μου έρευνας. Θα ήθελα να ευχαριστήσω τον αναπληρωτή καθηγητή κ. Γεώργιο Παπαδημητρίου για τις πολύτιμες συμβουλές του, καθοδηγώντας την όλη ερευνητική μου προσπάθεια στον καλύτερο δυνατό δρόμο. Θερμές ευχαριστίες οφείλονται αν μη τι άλλο, στον λέκτορα κ. Παναγιώτη Κατσαρό. Ξεκινώντας μαζί από το προπτυχιακό μου επίπεδο με το έναυσμα που μου έδωσε, βρέθηκα σε ένα προκλητικό και άκρως ενδιαφέρον ακαδημαϊκό περιβάλλον, διατηρώντας το συνεχές ενδιαφέρον του στην εξέλιξη της ερευνητικής μου εργασίας. Πραγματικά του οφείλω κάτι παραπάνω από την συμβολή του στην ερευνητική μου διαδρομή, μιας και μου παρείχε την ειλικρινή και ανιδιοτελή υποστήριξή του σε όλα τα θέματα που με απασχόλησαν τα τελευταία 4 χρόνια.

Ευχαριστίες οφείλονται και στους φίλους και συνεργάτες του εργαστηρίου αρχιτεκτονικής και δικτύων υπολογιστών του τμήματος πληροφορικής. Ειδικότερα στους υποψήφιους διδάκτορες Φώτη Λούκο για την ανιδιοτελή και πάντα παρούσα βοήθειά του, στον Γεώργιο Κάλφα για τις πολύτιμες συμβουλές του, στον Ανακρέοντα Μέντη για την αμέριστη συμπαράστασή του. Πολλές ευχαριστίες στους διδάκτορες Κωνσταντίνο Βυρσοκινό για όλες τις συζητήσεις που είχαμε μαζί και στη Σοφία Πετρίδου που συνεχίζει πάντα να μου προσφέρει την βοήθειά της, αποδεικνύοντας τον μοναδικό χαρακτήρας της. Ευχαριστώ επίσης τον Λέκτορα κ. Νίκο Πλέρο που οι συζητήσεις και συμβουλές του ήταν πάντα παρούσες οποιαδήποτε στιγμή και αν τις ζήτησα.

Πάνω απ' όλα όμως ευχαριστώ όλους εκείνους που βρέθηκαν κοντά μου στη διάρκεια της ζωής μου, για την αμέριστη και ουσιαστική συμπαράστασή τους, ειδικά σε αυτή τη διαδρομή. Τον Αλέξανδρο Καριπίδη και την Εύη , τον Λάζαρο Ευγενίδη και την Χλόη, τον Σταύρο Μπάλτο, τον Σέργιο Μουσιώλη, αλλά και τους

συγκατοίκους της Manor Place, Γιώργο Ρίζο και Βασίλη Ριζικιανό, και την Elena Prieto Rodriguez. Όλοι μαζί βρέθηκαν κάθε φορά, κάθε στιγμή στο πλευρό μου, προσφέροντας απλόχερα την συμπαράσταση και την βοήθειά τους, όποτε την χρειάστηκα. Επίσης αξίζει να αναφερθώ και στον Σωκράτη Μ., που οι εμπνεύσεις του μου έδιναν και συνεχίζουν να μου δίνουν κουράγιο σε όποιο εμπόδιο και αν βρεθεί μπροστά μου.

Ένα μεγάλο ευχαριστώ αξίζουν στα αδέρφια μου Χρήστο, Θανάση και Αγγελική που όχι μόνο στήριξαν αυτή μου την προσπάθεια, αλλά υπέμεναν και υπομένουν κάθε δύσκολη στιγμή μου. Τέλος, ευχαριστώ τους γονείς μου Δημήτρη και Μαρία για την ηθική και πρακτική υποστήριξή τους, αλλά και για τις κατάλληλες συνθήκες που δημιούργησαν έως τώρα για να βρίσκομαι σε αυτή τη θέση. Χωρίς αυτούς απλά δεν θα τα είχα καταφέρει!!!

Ιανουάριος 2010
Θεσσαλονίκη

Στυλιανός Δ. Μπασαγιάννης

Περίληψη

Στις μέρες μας το δίκτυα υπολογιστών και οι διαδικτυακές επικοινωνίες αναπτύσσονται ταχύτατα, με αποτέλεσμα να αυξάνεται ολοένα και περισσότερο η ανάγκη για ποιότητα των προσφερόμενων υπηρεσιών, αλλά πρωτίστως για την ασφάλειά τους. Η παρούσα διδακτορική διατριβή ασχολείται με την μελέτη και τον έλεγχο πρωτοκόλλων ασφαλείας με χρήση των τυπικών μεθόδων ανάλυσης συστημάτων. Ειδικότερα η ερευνά που παρουσιάζεται, αφορά τις ιδιότητες ασφαλείας που εγγυώνται σύγχρονα πρωτόκολλα ασφαλείας προς τις συμμετέχουσες οντότητες. Με τη χρήση αυτοματοποιημένων τυπικών μεθόδων επαλήθευσης, διεξήχθη μια ενδελεχής έρευνα προς την ανάπτυξη εξειδικευμένων θεωριών δημιουργίας εισβολέων. Με την βοήθεια αυτών ελέχθησαν εξαντλητικά για λάθη, σύγχρονα πρωτόκολλα ασφαλείας, προσπαθώντας με αυτό τον τρόπο να συμπεριληφθούν στον έλεγχο και εχθρικές ενέργειες που μπορεί να εξαπολύσει ένας κακόβουλος χρήστης. Οι εισβολείς αυτοί σχεδιάστηκαν και υλοποιήθηκαν μέσα σε περιβάλλοντα μοντελοποίησης σε αυτοματοποιημένα εργαλεία ελέγχου μοντέλων, που ανήκουν στην οικογένεια των Τυπικών Μεθόδων (Formal Methods) Ανάλυσης Συστημάτων.

Ένα από τα κύρια προβλήματα που παρουσιάζεται στην ερευνητική περιοχή του ελέγχου ασφαλείας με χρήση των τυπικών μεθόδων, είναι και αυτό της Έκρηξης του Χώρου Καταστάσεων, ΕΧΚ (state explosion problem). Πρόκειται για τον παραγόμενο χώρο των καταστάσεων της αναπαράστασης του συστήματος, ο οποίος εξαιτίας της πολυπλοκότητας των μηχανισμών που έγκειται σε όλα σχεδόν τα σύγχρονα πρωτόκολλα ασφαλείας (λ.χ. κρυπτογραφικών μηχανισμών, μηχανών παραγωγής τυχαίων αριθμών) μπορεί να οδηγήσει σε ΕΧΚ. Στην ίδια κατεύθυνση οδηγούμαστε και με την χρήση των υπαρκτών μεθοδολογιών κατασκευής εισβολέων, οι οποίες και στην πλειοψηφία τους βασίζονται στο διάσημο μοντέλο εισβολέα των Dolev και Yao. Πιο συγκεκριμένα, στην διατριβή αυτή, περιγράφεται η δημιουργία τριών καινούργιων θεωριών κατασκευής εισβολέων, οι οποίοι προσπαθούν με βάση τον σχεδιασμό τους, να προσπεράσουν το πρόβλημα της ΕΚΧ, αλλά παράλληλα να καταφέρουν να αναπαριστούν πιστά επιθέσεις που μπορούν να ανιχνευθούν σήμερα σε πρωτόκολλα ασφαλείας. Παρουσιάζεται η μαθηματική ορολογία στην οποία βασίστηκε ο κάθε εισβολέας, η αποτελεσματικότητά αυτών απέναντι σε μια σειρά από πρωτόκολλα ασφαλείας που εξετάσθηκαν και

παράλληλα, η ικανότητα που προσφέρουν, για την αποφυγή της ΕΧΚ. Ως αποτέλεσμα της χρήσης των εισβολέων αυτών, ανακαλύφθηκαν παραβιάσεις ασφαλείας σε υπό εξέταση πρωτόκολλα, οι οποίες ήταν άγνωστες μέχρι σήμερα.

Συνοψίζοντας, η συνεισφορά της παρούσας διατριβής εντοπίζεται κατά κύριο λόγο στον έλεγχο εγγυήσεων των πρωτοκόλλων ασφαλείας, ανεξάρτητα του μέσου επικοινωνίας που υλοποιείται το υπό εξέταση πρωτόκολλο. Με τη βοήθεια εξειδικευμένων εισβολέων επιτυγχάνεται, όχι μόνο ο εξαντλητικός έλεγχος για λάθη ασφαλείας των πρωτοκόλλων αυτών, αλλά επιπρόσθετα, αποφεύγεται η έκρηξη του χώρου των καταστάσεων, επιτρέποντας τη γρήγορη και αποτελεσματική ανάλυση του εκάστοτε συστήματος ασφαλείας.

Extended Abstract

Nowadays, the spread of computer networks and internet communications increases rapidly, causing an even strong need for quality of services, but mostly, the security of them. The present doctorate thesis involves with the verification of security protocols with the use of formal methods. The research been conducted, studies the security guaranties that most of the modern security protocols, tend to offer to the participant entities. Related work has shown that in order to successfully verify a security protocol, it is a necessity to check the protocol correctness while operating with a strong intruder entity. Using automatic model checking tools, we have developed discrete formal method techniques that combined with the proposed intruder theories can be useful for the correctness and the exhaustive verification of a series of security properties. Using these intruder theories, we verify security protocols' tolerance against common attack policies that can be applied today by dishonest protocol users.

One of the major problems found today in the research area of formal methods is the known problem of the State Space explosion. When modeling a system using model checking, the produced state space may increase dramatically due to the complexity of the processed involved in the model or for example the mechanisms that are embedded into (e.g. random generator number machine, cryptographic functions). Furthermore, the combination of a powerful intruder model into the security system may lead to an enormous state space, resulting in a difficult and time-consuming security analysis. In the present thesis, three distinct intruder theories and models have been developed in order to guide the analysts into a much easier analysis of their security protocols.

While the analyst today may find a wide area of security properties that target towards verification of them, the described intruder theories provide a selection for the appropriate intruder model –depending on the properties- to be applied, providing a successful verification mean, into revealing potential security flaws. We formally describe these three intruder theories and verify their success over a series of security protocols. As a result, the proposed intruders have not only dramatically decreased the produces state space during model checking but also they have been proved capable of revealing unknown to us security flaws in the tested protocol systems.

To conclude, the contribution of this thesis lies in the successful verification of the security guaranties of security protocols, with the help of specific intruder models, which are independent from the used mean of communication by the protocols' participants. Using the developed powerful intruder creation methodologies, the analyst may exhaustively check its protocol for security flaws, using formal methods techniques, without producing a large state space that could lead its analysis impossible to be conducted.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1ο

Εισαγωγή

1.1 Πρόλογος

Στην σημερινή εποχή, ένα από τα θέματα τα οποία απασχολούν τους επιστήμονες της πληροφορικής είναι και αυτό της ποιότητας του λογισμικού. Η ποιότητα και η αξιοπιστία του λογισμικού αποτελούν καίρια, αν μη τι άλλο ζητήματα, ειδικότερα σε περιπτώσεις, όπου η ασφάλεια παίζει πρωταρχικό ρόλο. Με την ραγδαία ανάπτυξη των διαδικτυακών εφαρμογών και επικοινωνιών, εμφανίζονται ολοένα και περισσότερο προβλήματα ασφαλείας, τα οποία προκαλούνται εξαιτίας των κακόβουλων ενεργειών από χρήστες που βρίσκονται σήμερα στο διαδίκτυο. Κάτι τέτοιο δημιουργεί μια αυξημένη ανάγκη για απαιτήσεις ασφαλείας και πιστοποίησης των διαδικτυακών υπηρεσιών που παρέχονται, ειδικότερα στις περιπτώσεις του ηλεκτρονικού εμπορίου και των διαδικτυακών συναλλαγών.

Παρόλο όμως της αύξησης των απαιτήσεών, μαζί αυξάνεται και η πολυπλοκότητα των εν λόγω συστημάτων λογισμικού, θέτοντας ένα σοβαρό ερώτημα του κατά πόσο αυτό πληρεί τις προϋποθέσεις ορθότητας και σωστής λειτουργίας του. Τη λύση στο συγκεκριμένο πρόβλημα έρχεται να δώσει ο συστηματικός έλεγχος του συστήματος παίζοντας ένα σημαντικό ρόλο στη βελτίωση της ποιότητάς και της ασφάλειάς του. Ο έλεγχος του συστήματος εμπεριέχει έναν εξαντλητικό συνεχή έλεγχο της ορθότητας, της δομής και της λειτουργίας του, μέσω αυτοματοποιημένων τεχνικών, καθοδηγούμενες από τον

αναλυτή. Στα διαδικτυακά πρωτόκολλα ασφαλείας, όπου ο έλεγχος για την ποιότητα και την ασφάλεια αυτών, είναι απαραίτητος, οι σημερινές τεχνικές των τυπικών μεθόδων ανάλυσης συστημάτων, έχουν αποδείξει την αποτελεσματικότητά τους, επαληθεύοντας την σωστή τους λειτουργία ή εντοπίζοντας λάθη, προειδοποιώντας τον αναλυτή για τη διόρθωσή τους.

Ειδικότερα για τον έλεγχο πρωτοκόλλων ασφαλείας, επιλέχθηκε η τεχνική του αυτόματου έλεγχου μοντέλων [29], μέρος των προσεγγίσεων των τυπικών μεθόδων ανάλυσης συστημάτων. Με αυτή την τεχνική, ο αναλυτής μοντελοποιεί το σύστημα (πρωτόκολλο) που θέλει να ελέγξει στη γλώσσα μοντελοποίησης που ορίζει το εργαλείο του ελέγχου μοντέλων. Το μοντέλο αυτό θα πρέπει να είναι πιστό στις βασικές λειτουργίες του πρωτοκόλλου, αλλά με προσέγγιση αφαιρετικότητας για εσωτερικές του διεργασίες, οι οποίες θεωρούνται ασφαλείς, και συνεπώς ανώφελο να εξεταστούν. Στα επόμενα κεφάλαια θα παρουσιαστεί εκτενής αναφορά για τις τυπικές μεθόδους και τον έλεγχο μοντέλων παρουσιάζοντας τα εργαλεία τα οποία χρησιμοποιήθηκαν για την εκάστοτε ανάλυση. Ιδιαίτερη έμφαση δίνεται στον αυτόματο ελεγκτή μοντέλων SPIN [46][93] και στο εργαλείο πιθανολογικού ελέγχου μοντέλων PRISM [91]. Σε αυτά τα εργαλεία υλοποιήθηκαν πρωτόκολλα ασφαλείας τα οποία και αλληλεπίδρασαν με έξυπνους εισβολείς, όπως θα περιγραφεί σε επόμενα κεφάλαια αυτής της διατριβής.

1.2 Τυπικές Μέθοδοι Ανάλυσης Συστημάτων

Οι τυπικές μέθοδοι ανάλυσης συστημάτων έκαναν την εμφάνισή τους στις αρχές τις δεκαετίας του 1980. Πρόκειται για μεθόδους ανάλυσης συστημάτων λογισμικού με την βοήθεια μαθηματικών τεχνικών, όπου δοθέντος ενός χώρου καταστάσεων που παράγει ολοκληρωτικά ένα υπό εξέταση σύστημα, μπορούν να επαληθεύσουν ποικίλες ομάδες ιδιοτήτων [83]. Οι τεχνικές αυτές στην πλειοψηφία τους, χειραγωγούν με έξυπνο τρόπο τον παραγόμενο χώρο καταστάσεων, έτσι ώστε να μπορέσουν να εξετάσουν όλες τις πιθανές καταστάσεις για τυχόν παραβάσεις συνθηκών ή γενικών λαθών του λογισμικού.

Τα τελευταία χρόνια παρατηρείται μια τάση ανάπτυξης τυπικών μεθόδων για το σχεδιασμό και την ανάλυση συστημάτων τόσο μέσω θεωρητικών

μεθόδων περιγραφής αυτών όσο και πρακτικών, μέσω αυτοματοποιημένων ή ημι-αυτοματοποιημένων εργαλείων [16][46][89]. Ενώ τα σημερινά συστήματα κινούνται ολοένα και περισσότερο στην παροχή ολοκληρωμένων υπηρεσιών, η πολυπλοκότητα που εισάγουν, οδηγεί τον εξαντλητικό έλεγχο του συστήματος σε πολύ δύσκολη, και αρκετές φορές αδύνατη διεργασία να εφαρμοστεί. Με βάση το γεγονός αυτό, η ερευνητική κοινότητα έτεινε προς την δημιουργία εξειδικευμένων τεχνικών και εργαλείων, όπου το κάθε ένα θα χρησιμοποιούνταν για διαφορετικές περιπτώσεις επαλήθευσης ιδιοτήτων. Έτσι, για κατανεμημένα συστήματα λογισμικού, όπως είναι και τα πρωτόκολλα ασφαλείας, επιλέγονται τεχνικές τυπικών μεθόδων που κυρίως συγκαταλέγονται στους αυτόματους ελεγκτές μοντέλων όπως το SPIN [93], το PRISM[91] ή το OFMC [16] της εργαλειοθήκης AVISPA [5]. Επιπρόσθετα ως πλεονέκτημα των τυπικών μεθόδων θεωρούνται και οι αλγεβρικές γλώσσες προδιαγραφών με τις οποίες περιγράφεται το υπό εξέταση σύστημα. Οι γλώσσες αυτές, αναπτύχθηκαν ειδικά για το σκοπό της εύκολης αλλά πιστής προδιαγραφής των μοντέλων, με σκοπό τον ουσιαστικό εξαντλητικό έλεγχο των καταστάσεων, μόνο στις απαραίτητες διεργασίες του συστήματος. Ειδικότερα στην περίπτωση περιγραφής των πρωτοκόλλων ασφαλείας, ανάλογα με τις ιδιότητες ορθότητας και τα λάθη τα οποία στοχεύει ο αναλυτή να αποκαλύψει, οι εκτελέσιμες αλγεβρικές γλώσσες προδιαγραφών, μπορούν με τις κατάλληλες τεχνικές αφαίρεσης να συμπεριλάβουν επικοινωνίες απομακρυσμένων οντοτήτων (λ.χ. συμμετέχουσες σε πρωτόκολλα) ή ταυτόχρονες εκτελέσεις διεργασιών.

Κύριο πλεονέκτημα των τυπικών μεθόδων, είναι ο έλεγχος και η ανάλυση ολόκληρου του γράφου των καταστάσεων που παράγεται από ένα μοντέλο-σύστημα. Με άλλα λόγια καταγράφονται όλες οι δυνατές και καταστάσεις που μπορεί να παραχθούν από όλες τις πιθανές εκτελέσεις του μοντέλου. Στη συνέχεια με ειδικές γλώσσες περιγραφής των ιδιοτήτων (φόρμουλες) που θέλουμε να ελέγξουμε, επισκέπτονται και επαληθεύονται οι καταστάσεις αυτές έναντι της φόρμουλας, παράγοντας είτε μια επιβεβαίωση ότι η φόρμουλα ισχύει, είτε παρουσιάζοντας το λάθος, δίνοντας επιπλέον και μια πιθανή εκτέλεση του συστήματος που μπορεί να οδηγήσει σε αυτό. Κάτι τέτοιο δίνει το πλεονέκτημα στις τυπικές μεθόδους ανάλυσης των συστημάτων σήμερα έναντι αυτών που βασίζονται στις απλές προσομοιώσεις (simulations) του συστήματος, καθώς

αυτές παρέχουν αποτελέσματα για μια μόνο πιθανή εκτέλεση του συστήματος, και όχι για όλες του τις εκτελέσεις (οι οποίες συχνά μπορεί να οδηγήσουν σε λάθη). Το βασικό θεωρητικό υπόβαθρο των τυπικών μεθόδων ανάλυσης αλλά και ειδικότερα του ελέγχου μοντέλων, θα παρουσιαστεί στο δεύτερο κεφάλαιο αυτής της διατριβής.

Ένα από τα κύρια προβλήματα που συναντάται στις τυπικές μεθόδους περιγραφής των συστημάτων είναι και αυτό της Έκρηξης του Χώρου των Καταστάσεων, ΕΧΚ (State Space Explosion) [97][95][78][45]. Το κύριο πρόβλημα της τεχνικής του έλεγχου μοντέλων είναι ότι ο χώρος καταστάσεων ενός συντρέχοντος (*concurrent)* συστήματος μπορεί να είναι αρκετά μεγάλος (και θεωρητικά άπειρος) σε μέγεθος. Για παράδειγμα, ένα σύστημα αποτελούμενο από *n* διεργασίες όπου η κάθε μία μπορεί να έχει *m* καταστάσεις, θα έχει ένα χώρο καταστάσεων *n*m*. Ο αριθμός αυτός των καταστάσεων περιορίζει την αποτελεσματικότητα της τεχνικής του ελέγχου μοντέλων, αφού όλες οι μεταβάσεις του συστήματος γίνονται μεγάλες -σε μερικές περιπτώσεις άπειρες- κάνοντας αρκετά δύσκολη (ή αδύνατη) την κατασκευή των γράφων τους. Αρκετοί ερευνητές έχουν μελετήσει το πρόβλημα οδηγούμενοι στο συμπέρασμα όμως ότι δεν υπάρχει κάποια γενική λύση για την αντιμετώπισή του. Στο σημείο αυτό θα παρουσιαστούν κάποιες στρατηγικές που αναφέρονται στην βιβλιογραφία και κρίνονται ως ικανές για την αντιμετώπιση της έκρηξης του χώρου καταστάσεων. Συγκεκριμένα οι τεχνικές αυτές μπορούν να διακριθούν σε δύο κατηγορίες:

- τις τεχνικές, οι οποίες προσπαθούν να μοντελοποιήσουν μόνο τις διεργασίες εκείνες του συστήματος που θεωρούνται αναγκαίες για την επαλήθευση της συγκεκριμένης προδιαγραφής που εξετάζεται [29],
- τις συμβολικές τεχνικές, οι οποίες αναπαριστούν συμβολικά της διεργασίες ενός συστήματος αντί να τις απαριθμούν με ακρίβεια (symbolic model checking techniques)[65]

Μερικές τεχνικές οι οποίες ανήκουν στην πρώτη κατηγορία είναι οι ακόλουθες: α) Τεχνικές μερικής ταξινομημένης μείωσης (Partial Order Reduction), β) On – The – Fly τεχνικές, γ) Συμμετρικές τεχνικές , δ)Τεχνική της Αφαίρεσης (Abstraction). Η δεύτερη κατηγορία των τεχνικών που προσπαθούν να αντιμετωπίσουν το φαινόμενο της έκρηξης καταστάσεων είναι και η

προσέγγιση του συμβολικού έλεγχου μοντέλων. Η ιδέα αυτής της τεχνικής είναι η αναπαράσταση όλων των καταστάσεων και μετατροπών του συστήματος που μοντελοποιεί το πρόγραμμα. Συχνά η αναπαράσταση αυτή πραγματοποιείται με τα δυαδικά γραφήματα απόφασης (Binary Decision Diagrams, BDDs). Αξίζει να σημειωθεί ότι η συμβολική τεχνική μας επιτρέπει να αντιμετωπίσουμε αρκετά πολύπλοκα συστήματα σε αντίθεση με τις άλλες τεχνικές που περιορίζονται σε αρκετά μικρό χώρο καταστάσεων. Ειδικά όμως στον έλεγχο των πρωτοκόλλων ασφαλείας, όπου καθεμία παραγόμενη κατάσταση μπορεί να ισοδυναμεί με παραβίαση ασφαλείας του πρωτοκόλλου, η συμβολική αναπαράσταση του χώρου των καταστάσεων μπορεί να οδηγήσεις σε σημαντικές παραλείψεις, παραβλέποντας καταστάσεις λάθους. Παρόλα αυτά, περισσότερες λεπτομέρειες για καθεμία από αυτές θα δοθούν στο επόμενο κεφάλαιο.

1.3 Έλεγχος Πρωτοκόλλων Ασφαλείας

Η ανάγκη για ασφάλεια στο λογισμικό που χρησιμοποιείται σήμερα, αυξάνεται ολοένα και περισσότερο ως απόρροια της ευρείας εξάπλωσης των εφαρμογών Η/Υ και του ηλεκτρονικού εμπορίου, συγκαταλέγοντας τις διαδικτυακές συναλλαγές που επιτελούνται μεταξύ διαφόρων χρηστών. Για τον λόγο αυτό, και εξαιτίας των διαφορετικών υπηρεσιών που προσφέρονται στο διαδίκτυο, έχουν δημιουργηθεί πληθώρα ηλεκτρονικών πρωτοκόλλων ασφαλείας με σκοπό την διασφάλιση της ποιότητα και πρωτίστως της ασφάλειας του συνόλου της επικοινωνίας. Τα περισσότερα από αυτά τα πρωτόκολλα βασίζουν την λειτουργία τους σε κρυπτογραφικούς μηχανισμούς, θωρακίζοντας έτσι το απόρρητο των επικοινωνιών από τρίτες, μη έμπιστες οντότητες. Τέτοιοι μηχανισμοί όπως η ασύμμετρη κρυπτογραφία (public key cryptography), η συμμετρική κρυπτογραφία (private key cryptography), ή οι συναρτήσεις κατακερματισμού (hash functions), προσφέρουν εγγυήσεις ασφαλείας όσον αφορά ένα σύνολο βασικών ιδιοτήτων, που πρέπει να επαληθεύονται σε κάθε σημείο εκτέλεσης του πρωτοκόλλου. Για την ενδυνάμωση των προαναφερθέντων μηχανισμών, υπεισέρχονται μηχανές παραγωγής τυχαίων αριθμών (random generator engines), καθώς και η συμμετοχή καθολικών

έμπιστων οντοτήτων, με σκοπό την διατήρηση της δικαιοσύνης (fairness) της όλης συνόδου του πρωτοκόλλου.

Το σύνολο των βασικών ιδιοτήτων που καλείται να προσφέρει ένα πρωτόκολλο ασφαλείας, ποικίλει ανάλογα με τις προϋποθέσεις για τις οποίες κατασκευάστηκε το πρωτόκολλο, που ορίζουν οι τελικοί του χρήστες. Σήμερα, η ασφάλεια της διακινούμενης πληροφορίας έγκειται στην προστασία της πληροφορίας στην ολότητά της [37], προσπαθώντας να αποκρύψει όλα της τα δεδομένα από μη εξουσιοδοτημένους χρήστες. Οι θεμελιώδεις ιδιότητες ασφαλείας συνοψίζονται σήμερα στην ακεραιότητα των πληροφοριών (integrity), στην εμπιστευτικότητα (confidentiality), στην αυθεντικοποίηση (authentication) των οντοτήτων που ανταλλάσσουν την πληροφορία και στην διαθεσιμότητα (availability) αυτής. Εκτός όμως από τις βασικές αυτές ιδιότητες, τα πρωτόκολλα ασφαλείας πρέπει σε αρκετές περιπτώσεις να παρέχουν επιπρόσθετες εγγυήσεις, οι οποίες και δε συγκαταλέγονται στην ομάδα των προαναφερθέντων ιδιοτήτων. Τέτοιες εγγυήσεις είναι η ιδιότητα της εγκυρότητας (validity) [37], η μοναδικότητα της πληροφορίας (uniqueness) και η μη αποποίηση της ευθύνης (non repudiation) από τις συμμετέχουσες οντότητες. Η έννοια της ασφάλειας πολλές φορές στον χώρο της πληροφορικής τείνει να έχει πολλές ερμηνείες, ανάλογα με τις απαιτήσεις του εκάστοτε συστήματος που υλοποιείται. Στις προαναφερθείσες ιδιότητες, πρωταρχικός στόχος τους είναι η διατήρηση συνθηκών ευρωστίας του συστήματος, όπου μέσα από την επαλήθευσή τους, διατηρείται η συνολική ασφάλεια της πληροφορίας. Καθώς όμως διαφορετικές περιπτώσεις και νέες τεχνικές παραβίασης της ασφάλειας, εμφανίζονται ολοένα και περισσότερο, δημιουργείται η ανάγκη για τον ακριβή καθορισμό των ιδιοτήτων ασφαλείας, κατά την διάρκεια ελέγχου με τις τυπικές μεθόδους. Οι παραπάνω ιδιότητες της ασφάλειας των πληροφοριών δεν μετρώνται σε απόλυτα μεγέθη αλλά είναι συγκρίσιμες και έτσι υπεισέρχεται σε ένα βαθμό η σχετικότητα [37]. Παρά τη σαφήνεια και απλότητα των ορισμών που δίδονται για τις τρεις βασικές ιδιότητες, στην πράξη δεν είναι πάντοτε εύκολο να προσδιορίσουμε πότε μία από αυτές έχει παραβιαστεί. Περισσότερες όμως πληροφορίες για όλες τις ιδιότητες των πρωτοκόλλων θα περιγραφθούν στο Κεφάλαιο 3 αυτής της διατριβής.

Παρόλα αυτά όμως, δεν είναι λίγες οι αναφορές εκείνες [10][23][53][84], όπου έχουν ανιχνεύσει και περιγράψει σοβαρά λάθη και παραβιάσεις της ασφάλειας, που επιτυγχάνουν κακόβουλοι χρήστες στα πρωτόκολλα ασφαλείας. Όλες οι παραβιάσεις αυτές βασίζονται στην έξυπνη τοποθέτηση ισχυρών μοντέλων εισβολέων, οι οποίοι τίθενται κυρίαρχοι του επικοινωνιακού μέσου μεταξύ των οντοτήτων του πρωτοκόλλου. Ένας από τους πιο γνωστούς εισβολείς είναι και ο εισβολέας Dolev – Yao [34], που πήρε το όνομά του από τους εφευρέτες του Dolev και Yao. Πρόκειται για τον εισβολέα, στις αρχές του οποίου βασίζεται η πλειοψηφία όσων μοντέλων επακολούθησαν προς τη συγκεκριμένη ερευνητική περιοχή.

Ο συγκεκριμένος εισβολέας βασίζεται πάνω στην βασική υπόθεση, ότι με την προϋπόθεση λειτουργίας ενός πρωτοκόλλου ασφαλείας, ο εισβολέας είναι σε θέση να κατέχει όλα τα προς ανταλλαγή μηνύματα των συμμετεχόντων. Επιπρόσθετα, με τη βοήθεια συγκεκριμένων διεργασιών του εισβολέα, ο ίδιος είναι ικανός να παράγει καινούργια μηνύματα, τα οποία και θα βασίζονται σε προηγούμενα υποκλεμμένα μηνύματα του πρωτοκόλλου. Παρόλο όμως τη μεγάλη απήχηση που είχε ο εισβολέας DY, στην περίπτωση του τυπικού ελέγχου μοντέλων, η ακριβής απεικόνισή του μοντέλου αυτού οδηγεί σε μεγάλο χώρο καταστάσεων. Ειδικότερα σε πρωτόκολλα ασφαλείας που περιλαμβάνουν πολύπλοκες διεργασίες, ο εισβολέας DY επιφέρει εύκολα το φαινόμενο της έκρηξης του χώρου των καταστάσεων, καθιστώντας την ανάλυση ασφαλείας του πρωτοκόλλου, χρονοβόρα και πολλές φορές αδύνατη να εκτελεσθεί. Αν και στην βιβλιογραφία, έχουν υπάρξει αναφορές για βελτιώσεις του μοντέλου του εισβολέα DY [23][71][84] [89], παρόλα αυτά η ανάλυση συγκεκριμένων ιδιοτήτων ασφαλείας για επικοινωνιακά συστήματα, επιφέρει σημαντικές δυσκολίες διεξαγωγής τους. Ο έλεγχος με τις τυπικές μεθόδους ανάλυσης συστημάτων αποτελεί μια από τις πιο διαδεδομένες μεθόδους εντοπισμού λαθών και επαλήθευσης των ιδιοτήτων των πρωτοκόλλων ασφαλείας [29]. Ειδικότερα, εάν ο αναλυτής μοντελοποιήσει κατάλληλα τον εισβολέα (κακόβουλο χρήστη) χωρίς να προκαλέσει την ΕΧΚ, θα μπορέσει να παρέχει ακριβείς εγγυήσεις για το πρωτόκολλο ασφαλείας και τις ιδιότητες που εξετάζει. Με τη χρήση εξειδικευμένων εισβολέων, τα υπό εξέταση πρωτόκολλα υπόκεινται σε διαρκείς επιθέσεις προσπαθώντας με αυτόν τον τρόπο, να

πραγματοποιηθεί ένας εξαντλητικός έλεγχος των παραγόμενων καταστάσεων, για τυχόν παραβιάσεις, σεβόμενοι πάντα το συνολικό χώρο των καταστάσεων.

1.4 Συνεισφορά της Διατριβής

Μετά από την απαραίτητη εισαγωγή στην ερευνητική περιοχή των τυπικών μεθόδων ανάλυσης συστημάτων και ειδικότερα του ελέγχου μοντέλων, καθώς και των προβλημάτων που εμφανίζονται κατά την διάρκεια ελέγχου των πρωτοκόλλων ασφαλείας, εντοπίζεται και περιγράφεται η κύρια συνεισφορά της παρούσας διατριβής. Η συνολική συνεισφορά συνοψίζεται στην παρακάτω λίστα:

- Μελετώνται οι τυπικές μέθοδοι ανάλυσης συστημάτων και ειδικότερα ο έλεγχος μοντέλων (model checking). Παρουσιάζονται οι βασικές αρχές που διέπουν τις φάσεις της συγκεκριμένης ανάλυσης, οι οποίες στοχεύουν, ως επί το πλείστον στον έλεγχο πρωτοκόλλων ασφαλείας. Εντοπίζονται τα εργαλεία εκείνα του ελέγχου μοντέλων που θα μπορούσαν να χρησιμεύσουν για την εξαντλητική επαλήθευση ιδιοτήτων των σημερινών πρωτοκόλλων ασφαλείας.
- Περιγράφεται η διαδικασία με την οποία ο εκάστοτε αναλυτής, μπορεί να εφαρμόσει τον έλεγχο μοντέλων σε πρωτόκολλα ασφαλείας που αυτός προτείνει, πιστοποιώντας την όλη διαδικασία με την εξαντλητική επαλήθευση της ορθότητας των ιδιοτήτων που επιθυμεί να αποδείξει, ότι το πρωτόκολλό του παρέχει. Επίσης μελετώνται τα πιο διαδεδομένα μοντέλα εισβολέων που παρουσιάζονται στην βιβλιογραφία, καθώς και οι επιπτώσεις που έχουν αυτά στον παραγόμενο χώρο των καταστάσεων, χρησιμοποιώντας τους στον έλεγχο μοντέλων πρωτοκόλλων ασφαλείας.
- Προτείνεται η θεωρία του μοντέλου εισβολέα πολλαπλών επιθέσεων (ΕΠΕ) [7]. Ο εισβολέας ορίζεται και υλοποιείται στο εργαλείο του ελεγκτή μοντέλων SPIN, προσπαθώντας από τη μία να αναπαραστήσει τεχνικές επιθέσεων ενάντια στις οντότητες του πρωτοκόλλου που ελέγχεται, χωρίς να οδηγεί στο φαινόμενο της έκρηξης του χώρου των καταστάσεων. Ο συγκεκριμένος εισβολέας θεωρείται ως ένα

παραμετρικό μοντέλο επιθέσεων, όπου ο αναλυτής ανάλογα με τις ανάγκες του, μπορεί να προσθέσει ή να αφαιρέσει (προς αποφυγή της ΕΧΚ) επιθέσεις από το σώμα του. Αποτέλεσμα του εισβολέα ΕΠΕ, είναι η ανακάλυψη μιας άγνωστης επίθεσης στο πρωτόκολλο ασφαλείας μικροπληρωμών PayWord[10] [9].

- Προτείνεται η θεωρία του μοντέλου εισβολέα διερεύνησης μηνύματος (ΕΔΜ)[6]. Ο εισβολέας αυτός βασίζεται στον γνωστό εισβολέα DY, παρέχοντας επιπλέον πληροφορίες για το πρωτόκολλο που αλληλεπιδρά, μέσω της διερεύνησης των μηνυμάτων που ανταλλάσσουν οι οντότητες του πρωτοκόλλου. Και αυτός ο εισβολέας ορίζεται και υλοποιείται στο εργαλείο του ελεγκτή μοντέλων SPIN, με κύριο χαρακτηριστικό του την επίδρασή του στον παραγόμενο χώρο των καταστάσεων. Ο συγκεκριμένος εισβολέας αποδεικνύει την αποτελεσματικότητά του, οδηγώντας επιτυχώς στην ανακάλυψη της γνωστής επίθεσης πλαστοπροσωπίας [63] στο πρωτόκολλο ασφαλείας ασύμμετρης κρυπτογράφησης των Needham και Schroeder [74] , εμφανίζοντας μεγάλο πλεονέκτημα σε σύγκριση με τον εισβολέα DY, στις παραγόμενες καταστάσεις [12].
- Προτείνεται η θεωρία του πιθανοκρατικού εισβολέα (ΠΕ) [8]. Ο συγκεκριμένος εισβολέας ορίζεται και υλοποιείται μέσα στο εργαλείο πιθανοκρατικού ελέγχου μοντέλων PRISM. Λόγω της φύσης της συγκεκριμένης τεχνικής, η οποία βασίζεται σε Μαρκοβιανές αλυσίδες διακριτού χρόνου (DTMC), ο εισβολέας αυτός ακολουθεί τις συγκεκριμένες αρχές, επικεντρώνοντας στην αποκάλυψη λαθών ασφαλείας που χαρακτηρίζονται ως επιθέσεις άρνησης της εξυπηρέτησης (Denial of Service). Πρόκειται για τον πρώτο εισβολέα που δημιουργήθηκε με τον τρόπο αυτό. Ως αποτέλεσμα του εισβολέα ΠΕ, είναι η ανακάλυψη μιας άγνωστης επίθεσης DoS [9] στο πρωτόκολλο ασφαλείας ασύρματου ή ενσύρματου περιβάλλοντος HIP[50].
- Σε όλες τις παραπάνω επιθέσεις, προτάθηκαν τρόποι ενδυνάμωσης τον συγκεκριμένων πρωτοκόλλων, προς αποφυγή των παραβιάσεων ασφαλείας που εντοπίστηκαν από τους προτεινόμενους εισβολείς.

Επιπλέον, θα αναφερθούν περιληπτικά προσεγγίσεις ασφαλείας για δίκτυα κλειδωμάτων (network interlockings), όπου αναπτύχθηκε ένας αλγόριθμος ο οποίος διασφαλίζει την ασφάλεια των κλειδωμάτων για τους πόρους ενός δικτύου, για τις συμμετέχουσες οντότητές του. Ο αλγόριθμος αυτός αναπτύχθηκε και εφαρμόστηκε αρχικά στο [11], και στη συνέχεια επεκτάθηκε ως μια ολοκληρωμένη πρόταση ασφαλούς διαμοιρασμού πόρων σε δίκτυα στο [14]. Επιπρόσθετα δημιουργήθηκε μια καινούργια έκδοση του αλγορίθμου, η οποία χαρακτηρίζεται ως αλγόριθμος ασφάλειας-λάθους, αποδεικνύοντας την αποτελεσματικότητά του σε δίκτυα ανοχής λαθών [15]. Οι τελευταίες ερευνητικές προσπάθειες, πραγματοποιήθηκαν κατά την διάρκεια της διατριβής, και αν και δεν συμπεριλαμβάνονται στα κύρια κεφάλαιά της, βοήθησαν επικουρικά την ερευνητική μελέτη του χώρου των τυπικών μεθόδων ανάλυσης συστημάτων.

1.5 Δομή της Διατριβής

Ο γενικός στόχος αυτής της διατριβής είναι η ανάπτυξη τυπικών μεθόδων για τον εξαντλητικό έλεγχο πρωτοκόλλων ασφαλείας. Προσπαθώντας να αποφευχθεί το φαινόμενο της ΕΧΚ που περιγράφθηκε παραπάνω, δημιουργήθηκαν με τη βοήθεια του ελέγχου μοντέλων, ξεχωριστές θεωρίες εισβολέων, οι οποίες καταφέρνουν από τη μια και προσπερνούν το φαινόμενο ΕΧΚ, αλλά παράλληλα διατηρούν την δύναμή τους, στο να εξαπολύουν από απλές μέχρι και πολύπλοκες επιθέσεις, στα υπό εξέταση πρωτόκολλο. Η δομή της παρούσας διατριβής βρίσκεται σε απόλυτη συνέπεια, προς τον παραπάνω στόχο, περιγράφοντας αρχικά βασικές αρχές που διέπουν τόσο τις τυπικές μεθόδους ανάλυσης συστημάτων και ειδικότερα τον έλεγχο μοντέλων, όσο και αυτές των σημερινών πρωτοκόλλων ασφαλείας.

Στο παρόν *πρώτο κεφάλαιο* περιέχεται η περιγραφή, η οριοθέτηση και η σημασία επίλυσης του συγκεκριμένου προβλήματος. Σε πρώτη φάση γίνεται η εισαγωγή στο πλαίσιο του προβλήματος, δίνοντας έμφαση στην σημασία των τυπικών μεθόδων ανάλυσης συστημάτων σήμερα. Συγκεκριμένα, παρουσιάζεται η επιτυχία και ο λόγος χρήσης των μεθόδων αυτών πάνω σε συστήματα λογισμικού που είναι κρίσιμα σε ασφάλεια. Παρουσιάζεται το γνωστό

φαινόμενο της ΕΧΚ που εμφανίζεται, αλλά και τεχνικές οι οποίες το αντιμετωπίζουν, ειδικότερα για τις περιπτώσεις χρήσεις εξειδικευμένων εισβολέων, για τον έλεγχο πρωτοκόλλων ασφαλείας.

Στο *δεύτερο κεφάλαιο*, παρουσιάζονται οι βασικές έννοιες που σχετίζονται με τις τυπικές μεθόδους ανάλυσης συστημάτων και ειδικότερα του ελέγχου μοντέλων. Θεμελιώνονται θεωρητικά οι έννοιες που συνθέτουν τον έλεγχο μοντέλων σε ένα σύστημα, παραθέτοντας περιγραφές για την κατηγορία των ιδιοτήτων, των γλωσσών προδιαγραφών και των εργαλείων που χρησιμοποιούνται σήμερα. Ειδικότερα, το ενδιαφέρον επικεντρώνεται στα εργαλεία και τις τεχνικές που χρησιμοποιήθηκαν για την ανάπτυξη μεθόδων που εφαρμόστηκαν σε αυτή τη διατριβή.

Στο *τρίτο κεφάλαιο* περιγράφονται οι βασικές αρχές για τα πρωτόκολλα ασφαλείας και τον έλεγχο αυτών με τις τυπικές μεθόδους. Γίνεται αναφορά σε προηγούμενες πετυχημένες ερευνητικές προσπάθειες ανάπτυξης τυπικών μεθόδων, καθώς και αναλύσεις ασφαλείας οι οποίες μετέπειτα αποδείχθηκαν λανθασμένες με την βοήθεια του ελέγχου μοντέλων. Παράλληλα περιγράφονται αναλυτικά οι ιδιότητες ασφάλειας που καλούνται να προσφέρουν τα σημερινά πρωτόκολλα ασφαλείας, καθώς και τα είδη των επιθέσεων που μπορούν αυτά να υποστούν από εισβολείς (κακόβουλους χρήστες). Τέλος αναφορά γίνεται και στους κρυπτογραφικούς μηχανισμούς που ως επί το πλείστον, εμπεριέχονται στις υλοποιήσεις των πρωτοκόλλων αυτών.

Στο *τέταρτο κεφάλαιο* παρουσιάζεται το πρώτο μοντέλο εισβολέα που αναπτύχθηκε, το επονομαζόμενο **μοντέλο εισβολέα πολλαπλών επιθέσεων, ΕΠΕ** [7]. Περιγράφονται φορμαλιστικά όλες οι λειτουργίες του, τακτικές επιθέσεων και επιθέσεις ως ολότητες που περιλαμβάνει η δομή του. Ο συγκεκριμένος εισβολέας υλοποιήθηκε μέσα στον αυτόματο ελεγκτή μοντέλων SPIN, αποδεικνύοντας την αποτελεσματικότητά του με την εφαρμογή του στον έλεγχο δύο πρωτοκόλλων ασφαλείας μικροπληρωμών. Με την βοήθεια του συγκεκριμένου εισβολέα εντοπίστηκε μια παράβαση ασφαλείας στο πρωτόκολλο PayWord.

Στο *πέμπτο κεφάλαιο* παρουσιάζεται το δεύτερο μοντέλο εισβολέα που αναπτύχθηκε, το επονομαζόμενο **μοντέλο εισβολέα διερεύνησης μηνύματος, ΕΔΜ** [6]. Και σε αυτή την περίπτωση περιγράφονται φορμαλιστικά ο εισβολέας

και οι λειτουργίες του, καθώς και ο αλγόριθμος που αναπτύχθηκε για τη διερεύνηση των μηνυμάτων του συγκεκριμένου εισβολέα. Ο ΕΔΜ υλοποιήθηκε και αυτός στο εργαλείο SPIN, δίνοντας όμως τις κατάλληλες περιγραφές για να υλοποιηθεί και σε περαιτέρω εργαλεία ελέγχου μοντέλων. Βασιζόμενος στη δομή του εισβολέα ΕΠΕ, ο ΕΔΜ παρουσιάζει την ικανότητα να προτείνει στον αναλυτή ποιες επιθέσεις πρόκειται να πετύχουν απέναντι στο πρωτόκολλο ασφαλείας, και ποιες όχι, οδηγώντας έτσι τον αναλυτή σε διορθώσεις του μοντέλου του, και κλαδεύοντας αποτελεσματικά τον παραγόμενο χώρο καταστάσεων. Ο εισβολέας εφαρμόστηκε επιτυχώς στο πρωτόκολλο NSPK, όπου και συγκριτικά με τον εισβολέα DY, απέδειξε ότι μπορεί να μειώσει τον χώρο των καταστάσεων σε μερικές χιλιάδες, αποφεύγοντας έτσι το φαινόμενο της ΕΧΚ.

Στο *έκτο κεφάλαιο* παρουσιάζεται το τρίτο κατά σειρά μοντέλο εισβολέα που αναπτύχθηκε, το επονομαζόμενο **πιθανοκρατικό μοντέλο εισβολέα, ΠΕ** [8]. Το μοντέλο ΠΕ διαφέρει αισθητά από τα προηγούμενα μοντέλα, μιας και βασίστηκε στις Μαρκοβιανές αλυσίδες διακριτού χρόνου. Υλοποιήθηκε στο περιβάλλον του πιθανοκρατικού ελεγκτή μοντέλων PRISM, με σκοπό την πιθανοκρατική ανάλυση του χώρου των καταστάσεων ενός πρωτοκόλλου ασφαλείας, με κύριο σκοπό τον έλεγχο για επιθέσεις άρνησης εξυπηρέτησης (Denial of Service Attack, DoS). Ως αποτέλεσμα αυτού, μοντελοποιήθηκε και πραγματοποιήθηκε ο έλεγχος του πρωτοκόλλου ασφαλείας HIP (Host Identity Protocol). Με την βοήθεια του εισβολέα ΠΕ και βάσει της ικανότητάς του να προσομοιώνει N οντότητες φαντάσματα (zombie machines), ο εισβολέας κατάφερε να εντοπίσει μια επίθεση DoS για το πρωτόκολλο HIP, με την πιθανότητα να υπολογίζεται στο 0,89.

Στο *κεφάλαιο επτά* περιγράφονται τα συνολικά συμπεράσματα αυτής της διατριβή καθώς και οι προοπτικές που ανοίγονται για περαιτέρω έρευνα. Τέλος, παρουσιάζεται ένας πίνακας με τις επεξηγήσεις των συντομεύσεων και των ακρωνυμίων που χρησιμοποιούνται, καθώς και οι αντίστοιχοι όροι στην αγγλική γλώσσα.

Κεφάλαιο 2ο

Τυπικές Μέθοδοι Ανάλυσης και Έλεγχος Μοντέλων

2.1 Εισαγωγή

Ο κατανεμημένος σχεδιασμός πολλών εφαρμογών είναι γνωστό ότι εισάγει καινούργια ζητήματα ασφαλείας, τα οποία μέχρι πρότινος δεν απασχολούσαν τη βιομηχανία των υπολογιστών. Διαφορετικά στοιχεία μιας κατανεμημένης εφαρμογής μπορούν να εκτελεστούν σε διαφορετικούς υπολογιστές, χωρίς να είναι αναγκαία η ύπαρξη ενός κοινού ρολογιού, το οποίο σημαίνει ότι η εκτέλεση είναι ασύγχρονη. Επειδή όμως στην επικοινωνία των υπολογιστών αυτών μπορεί να μεσολαβεί κάποιο δίκτυο, υπάρχει η πιθανότητα να παρουσιάζονται καθυστερήσεις, οι οποίες οφείλονται σε κάποιες άλλες εφαρμογές που εξυπηρετεί το δίκτυο. Αξίζει να σημειωθεί ότι οι καθυστερήσεις αυτές δε μπορούν να προσδιοριστούν και να προβλεφθούν. Λύση στο πρόβλημα των καθυστερήσεων που προκύπτουν, έρχεται να δώσει ο μη-ντετερμινισμός. Επιπρόσθετα, πολλές εφαρμογές περιέχουν γεγονότα που απαιτούν την εκτέλεσή τους σε μία συγκεκριμένη σειρά. Αυτό έχει ως αποτέλεσμα ότι πρέπει να συγχρονιστούν και συντονιστούν τα διαφορετικά ασύγχρονα στοιχεία μιας εφαρμογής με σκοπό να εκτελεστούν στη συγκεκριμένη σειρά που απαιτούν. Ο συγχρονισμός κάνει την κατασκευή κατανεμημένων εφαρμογών αρκετά

περίπλοκη και ένας τρόπος υλοποίησης τους είναι με χρήση αφηρημένων γλωσσών υψηλού επιπέδου.

Τα παράλληλα συστήματα πεπερασμένης κατάστασης διαδίδονται ολοένα και περισσότερο, ιδιαίτερα στο χώρο του σχεδιασμού ψηφιακών κυκλωμάτων και των πρωτοκόλλων επικοινωνίας. Κάποια λογικά λάθη που βρίσκονται σε ένα από τα τελευταία στάδια της σχεδίασης τέτοιων συστημάτων, αποτελούν σημαντικό πρόβλημα τόσο για τους σχεδιαστές, όσο και για τους προγραμματιστές. Τέτοιου είδους λάθη μπορούν είτε να καθυστερήσουν την παραγωγή ενός καινούργιου προϊόντος, είτε να προκαλέσουν σφάλμα σε κάποια συσκευή που είναι ήδη σε χρήση. Παράδειγμα τέτοιου λάθους είναι ο πύραυλος Ariane 5, ο οποίος εξερράγη σε λιγότερο από 40 δευτερόλεπτα από την εκτόξευσή του. Η υπεύθυνη επιτροπή έρευνας έβγαλε το πόρισμα ότι η πτώση οφειλόταν σε λάθος του λογισμικού του υπολογιστή, που ήταν υπεύθυνος για την κίνηση του πυραύλου. Πιο συγκεκριμένα, κατά τη διάρκεια της εκτόξευσης σε έναν 64-bit αριθμό κινητής υποδιαστολής προσπάθησε να γίνει η αποθήκευσή του σε θέση μνήμης για 16-bit ακέραιο. Αυτή η διεργασία δεν καλυπτόταν από το κώδικα του προγράμματος με αποτέλεσμα ο υπολογιστής να παρουσιάσει σφάλμα και κατά συνέπεια ο πύραυλος να εκραγεί. Η υπεύθυνη επιτροπή έρευνας πρότεινε τη λήψη μέτρων, κυρίως κατασκευή λογισμικού επαλήθευσης, για την αποφυγή παρόμοιων περιστατικών στο μέλλον.

Η πιο διαδεδομένη τεχνική επαλήθευσης βασίζεται σε εκτεταμένο έλεγχο ή προσομοίωση, αλλά μπορεί να μην εντοπίσει σημαντικά λάθη όταν ο αριθμός των καταστάσεων του αλγορίθμου ή του πρωτοκόλλου είναι πολύ μεγάλος. Σε αυτό το σημείο η τεχνολογική κοινότητα ανέπτυξε τεχνικές και μεθόδους οι οποίες θα μπορούσαν να επαληθεύσουν την λειτουργία και ορθότητα του εκάστοτε λογισμικού. Οι μέθοδοι αυτοί ονομάστηκαν Τυπικές Μέθοδοι Ανάλυσης Συστημάτων (Formal Methods). Παρά το γεγονός ότι είχαν διεξαχθεί πολλές έρευνες για την υλοποίηση της τεχνικής επαλήθευσης, τα περισσότερα αποτελέσματα δεν ήταν ικανοποιητικά, λόγω του ότι τα συστήματα που κατασκευάζονταν απαιτούσαν πολύ χρόνο για την προσομοίωση, αλλά και πολλές ενέργειες από το χρήστη. Τη δεκαετία 1980 επινοήθηκε μια εναλλακτική τεχνική επαλήθευσης από τους Clarke, Emerson, Quielle και Sifakis , που ονομάστηκε χρονική λογική έλεγχου μοντέλων (temporal logic model checking).

Σε αυτή την προσέγγιση οι αλγόριθμοι και τα πρωτόκολλα μοντελοποιούνται ως συστήματα μετάβασης καταστάσεων [31]. Ο έλεγχος μοντέλων (model checking) παρουσιάζει σημαντικά πλεονεκτήματα:

1. Το κυριότερο είναι ότι στο σύνολό της η διαδικασία είναι αυτόματη και δεν απαιτούνται πολλές ενέργειες από το χρήστη. Με απλά λόγια, ο χρήστης εισάγει μία αναπαράσταση του μοντέλου και τα χαρακτηριστικά που επιθυμεί να ελέγξει. Ο αλγόριθμος της επαλήθευσης ή θα τερματίσει το πρόγραμμα με απάντηση true, ή θα δώσει μία απάντηση ότι η φόρμουλα είναι οδηγεί σε λάθος.
2. Επίσης η διαδικασία είναι πολύ γρήγορη και συνήθως υπάρχει απόκριση μέσα σε διάστημα λίγων λεπτών, με εξαίρεση το φαινόμενο της Έκρηξης του χώρου των καταστάσεων (ΕΧΚ), που μπορεί να συμβεί αν το σύστημα που επαληθεύεται αποτελείται από πολλά στοιχεία που κάνουν παράλληλες μεταβάσεις. Σε αυτή την περίπτωση οι καταστάσεις του συστήματος μπορούν να αυξάνονται εκθετικά. Εξαιτίας αυτού του προβλήματος, πολλοί ερευνητές προέβλεψαν ότι ο έλεγχος μοντέλων δε θα μπορούσε να είναι πρακτικός σε μεγάλα προβλήματα.

Η έκρηξη του χώρου καταστάσεων θα περιγραφθεί σε επόμενη παράγραφο αναλυτικά. Παρά το γεγονός αυτό, τα συστήματα που ελέγχονται αυξάνονται συνεχώς και αυτό οφείλεται κυρίως στη χρησιμοποίηση των δυφιακών διαγραμμάτων αποφάσεων (Binary Decision Diagrams, BDDs) που αποτελούν μία δομή δεδομένων για αναπαράσταση Boolean συναρτήσεων. Αυτή η μέθοδος είναι εξαιρετικά χρήσιμη για σύγχρονα κυκλώματα. Σε ασύγχρονα πρωτόκολλα επικοινωνιών, είναι δυνατό να μειωθεί το μέγεθος του χώρου καταστάσεων με τη χρησιμοποίηση διαφόρων τεχνικών, όπως αυτή της τεχνικής μερικής διατεταγμένης μείωσης (Partial Order Reduction, POR). Ως αποτέλεσμα αυτών των τεχνικών ο «έλεγχος μοντέλων» έχει ευρεία διάδοση ως τεχνική επαλήθευσης. Διακρίνουμε τα παρακάτω επίπεδα ανάλογα με την εφαρμογή των τυπικών [29]:

- 1ο Επίπεδο: Εφαρμογή των τυπικών μεθόδων σε όλα τα μέρη του συστήματος
- 2ο Επίπεδο: Εφαρμογή των τυπικών μεθόδων με δύο ή περισσότερα επίπεδα αφαίρεσης του συνόλου καταστάσεων χρησιμοποιώντας

γραπτές αποδείξεις ότι στη συγκεκριμένη προδιαγραφή μπορούν να εφαρμοστούν τεχνικές μείωσης

- 3ο Επίπεδο: Εφαρμογή τυπικών μεθόδων στην περίπτωση προδιαγραφής η οποία ελέγχεται από ελεγκτή θεωρητικής μηχανικής

Υπάρχουν δύο τύποι ελεγκτών αποδεικτικής θεωρίας [29]:

- Αυτόματοι: Προσπαθούν να αποδείξουν τη συγκεκριμένη προδιαγραφή χωρίς ανθρώπινη παρέμβαση
- Διαλογικοί: Προσπαθούν να αποδείξουν τη συγκεκριμένη ιδιότητα με την καθοδήγηση του χρήστη

Μία τυπική ιδιότητα αποτελεί μια ακριβή, ολοκληρωμένη, επαρκή και μη διφορούμενη βάση, τόσο για το σχεδιασμό και την κωδικοποίηση ενός συστήματος όσο και για τον έλεγχό του. Πρόκειται για ένα μεγάλο πλεονέκτημα σε σχέση με τις παραδοσιακές διεργασίες ελέγχου. Ένα δεύτερο πλεονέκτημα της χρήσης των τυπικών μεθόδων για έλεγχο είναι η ικανότητά τους να τον πραγματοποιούν με την βοήθεια εργαλείων. Οι αλγόριθμοι που χρησιμοποιούν τα εργαλεία αυτά, έχουν να κάνουν με τον αυτόματο έλεγχο συστημάτων που είναι αρκετά γρήγορος και με λιγότερα λάθη. Κάτι τέτοιο ανοίγει τον δρόμο για την περίπτωση ελέγχου που το μόνο που χρειάζεται είναι η ιδιότητα προδιαγραφής του συστήματος που εξετάζεται. Όλοι αυτοί οι έλεγχοι μπορούν αυτόματα να αποδειχθούν εάν είναι σωστοί ή όχι.

2.1.1 Τυπικός έλεγχος

Ο τυπικός έλεγχος και η τυπική επαλήθευσή αποτελούν συμπληρωματικές τεχνικές για ανάλυση και έλεγχο της ορθότητας συστημάτων. Ενώ η επαλήθευση έχει ως στόχο την απόδειξη ιδιοτήτων συστημάτων που έχουν μοντελοποιηθεί με βάση μαθηματικούς όρους, ο έλεγχος εφαρμόζεται με το να πραγματοποιεί την πραγματική εκτέλεση της εφαρμογής ή την εκτέλεση μιας προσομοίωσης του μοντέλου. Επίσης η επαλήθευση μπορεί να δώσει απάντηση εάν μια ιδιότητα ικανοποιείται αρκεί το σύστημα να είναι σωστά μοντελοποιημένο αν και μόνο αν *ισχύει ότι μια επαλήθευση είναι σωστή πάντα όταν είναι σωστό και το μοντέλο που αναπαριστά το σύστημα*. Ο έλεγχος που βασίζεται μόνο στην παρατήρηση ενός μικρού συνόλου καταστάσεων ενός συστήματος ποτέ δεν είναι επαρκής. Ο έλεγχος μπορεί να υποδείξει μόνο την

παρουσία λαθών και όχι την διαβεβαίωση ότι δεν πρόκειται να υπάρξουν λάθη στο μέλλον. Αλλά από την στιγμή που ο έλεγχος μπορεί να εφαρμοστεί σε πραγματικές εφαρμογές, είναι χρήσιμο σε αυτές τις περιπτώσεις να υπάρξει ένα αξιόπιστο μοντέλο αναπαράστασης της πραγματικής εφαρμογής που ελέγχεται.

2.1.2 Τυπική Επαλήθευση

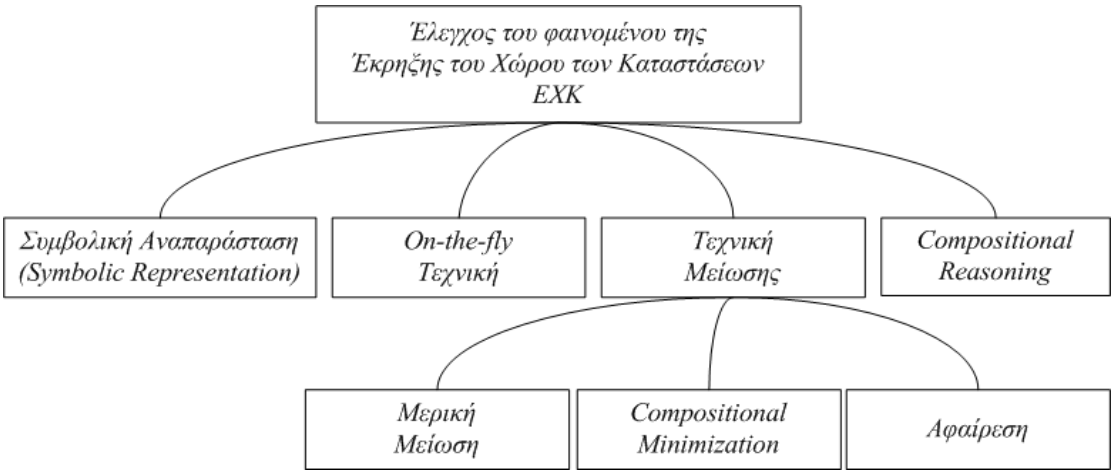
Η τυπική επαλήθευση αποτελεί μια εναλλακτική οικογένεια μεθόδων που χρησιμοποιούν τεχνικές βασισμένες πάνω σε μια μαθηματική λογική με σκοπό να διασφαλίσουν την ποιότητα συστημάτων υλικού ή λογισμικού. Ο σχεδιασμός της επαλήθευσης που χρησιμοποιεί μια τυπική λογική κατά την καταγραφή των προδιαγραφών ενός συστήματος, παρέχει ένα δομημένο πλαίσιο για την μετάφραση υψηλού επιπέδου προδιαγραφών σε μια συλλογή διακριτών στοιχείων που περιγράφονται από αλγόριθμους, εύκολα κατανοητά από τους προγραμματιστές.

Η επαλήθευση προγραμμάτων εμπλέκει ουσιαστικά αποδείξεις ορθότητας αυτών. Πολλές εταιρείες λογισμικού εισάγουν την επαλήθευση στα συστήματά τους ως ένα ξεχωριστό και πολύ σημαντικό τμήμα του προϊόντος τους. Η αφοσίωση που δίνεται πάνω σε όλες τις μεθόδους επαλήθευσης έχει προσθέσει αρκετά επίπεδα όσο αφορά στην αξιόπιστη παραγωγή του λογισμικού. Βοηθούν τον σχεδιαστή να επικεντρώσει πάνω στο θέμα του τι κάνει το λογισμικό και όχι τον τρόπο με τον οποίο το κάνει. Γενικά οι μέθοδοι αυτοί μπορούν να θεωρηθούν σαν ένα σύνολο διαφορετικών τεστ από όπου πρέπει να περάσει το προϊόν, όπως τον έλεγχο μοντέλων, έλεγχο κώδικα και αρκετές άλλες.

2.2 Το πρόβλημα της έκρηξης καταστάσεων

Το κύριο πρόβλημα της τεχνικής του έλεγχου μοντέλων είναι ότι ο χώρος καταστάσεων ενός συντρέχοντος (*concurrent)* συστήματος μπορεί να είναι αρκετά μεγάλος σε μέγεθος [95]. Για παράδειγμα, ένα σύστημα αποτελούμενο από n διεργασίες όπου η κάθε μία μπορεί να έχει m καταστάσεις, θα έχει έναν χώρο καταστάσεων *n*m*. Ο αριθμός αυτός των καταστάσεων περιορίζει την αποτελεσματικότητα της τεχνικής ελέγχου, αφού όλες οι μεταβάσεις του συστήματος γίνονται μεγάλες (σε μερικές περιπτώσεις άπειρες) κάνοντας

αρκετά δύσκολο (ή αδύνατο) την κατασκευή των μοντέλων τους. Το φαινόμενο αυτό είναι γνωστό ως το φαινόμενο της έκρηξης του χώρου καταστάσεων (State explosion problem) [97].



Εικόνα 2.2.1 Έλεγχος της έκρηξης του χώρου των καταστάσεων με σημερινές τεχνικές

Αρκετοί ερευνητές έχουν προσπαθήσει να μετριάσουν το πρόβλημα βγάζοντας ως συμπέρασμα όμως ότι δεν υπάρχει κάποια γενική λύση για την αντιμετώπισή του (Εικόνα 2.2.1). Στο σημείο αυτό θα παρουσιαστούν μερικές διαφορετικές στρατηγικές που αναφέρονται στην βιβλιογραφία και κρίνονται ως ικανές για την αντιμετώπιση της έκρηξης του χώρου καταστάσεων. Συγκεκριμένα οι στρατηγικές αυτές μπορούν να διακριθούν σε δύο κατηγορίες [29]:

- Τις τεχνικές αυτές οι οποίες προσπαθούν να μοντελοποιήσουν μόνο τις καταστάσεις εκείνες του συστήματος που θεωρούνται αναγκαίες για την επαλήθευση της συγκεκριμένης προδιαγραφής που εξετάζεται.
- Τις συμβολικές τεχνικές οι οποίες αναπαριστούν συμβολικά της καταστάσεις ενός συστήματος αντί να τις απαριθμούν με ακρίβεια

Μερικές τεχνικές οι οποίες ανήκουν στην πρώτη κατηγορία είναι οι ακόλουθες[29] [45]:

- Τεχνικές μερικής ταξινομημένης μείωσης (Partial Order Reduction)
- On – The – Fly τεχνικές
- Συμμετρικές τεχνικές
- Τεχνική της Αφαίρεσης (Abstraction)

Η δεύτερη κατηγορία των τεχνικών που προσπαθούν να αντιμετωπίσουν το φαινόμενο της έκρηξης καταστάσεων είναι και η προσέγγιση του συμβολικού έλεγχου μοντέλων. Η ιδέα αυτής της τεχνικής είναι η αναπαράσταση όλων των καταστάσεων και μετατροπών του συστήματος που μοντελοποιεί το πρόγραμμα. Συχνά η αναπαράσταση αυτή πραγματοποιείται με την βοήθεια των BDDs. Πρώτος ο McMillan το 1993 στο [65] ήταν αυτός ο οποίος εισήγαγε την έννοια συμβολικός έλεγχος μοντέλων. Τέλος αξίζει να σημειωθεί ότι η συμβολική αυτή τεχνική μας επιτρέπει να αντιμετωπίσουμε αρκετά πολύπλοκα συστήματα σε αντίθεση με τις άλλες τεχνικές που περιορίζονται σε αρκετά μικρό χώρο καταστάσεων.

Παρακάτω παρουσιάζεται ένας ενδεικτικός πίνακας 2.2.1 ο οποίος δείχνει την σχέση διαφόρων μεθόδων ελέγχου με κάποια σημαντικά κριτήρια που αφορούν την ανάπτυξη συστημάτων υλικού και λογισμικού.

Πίνακας 2.2.1 Σχέση μεθόδων τυπικού ελέγχου και επαλήθευσης

Κριτήρια Μεθόδου	***Έλεγχος***	***Επαλήθευση***	***Έλεγχος Μοντέλων***
Μέγεθος Συστήματος	Μικρό ως πολύ μεγάλο	Παραδείγματα Παιχνιδιών	100 – 1000 γραμμές κώδικα
Χρόνος	Ανάλογα διάρκεια ανάπτυξης	Μέρες – Βδομάδες	Λεπτά – Ώρες
Ειδικότητα χρηστών	Έμπειροι προγραμματιστές	Μαθηματικοί, Επιστήμονες Λογικής	Επιστήμονες Λογικής
Χρήση σήμερα	Υλικό και λογισμικό	Έρευνα	Υλικό και λογισμικό
Προδιαγραφές	Ανάλογα επιθυμίας χρήστη	Λογικές ή βασισμένες σε αυτόματα	Λογικές ή βασισμένες σε αυτόματα
Μοντελοποίηση και αλλαγές	Εφαρμόζεται απευθείας	Επιβάλλεται κατά την πραγματοποίηση	Επιβάλλεται κατά την πραγματοποίηση

Παράδειγμα 1 *Έκρηξης του χώρου καταστάσεων, ΕΧΚ. Έστω το παρακάτω κομμάτι κώδικα το οποίο χρησιμοποιεί μια εντολή επανάληψης :*

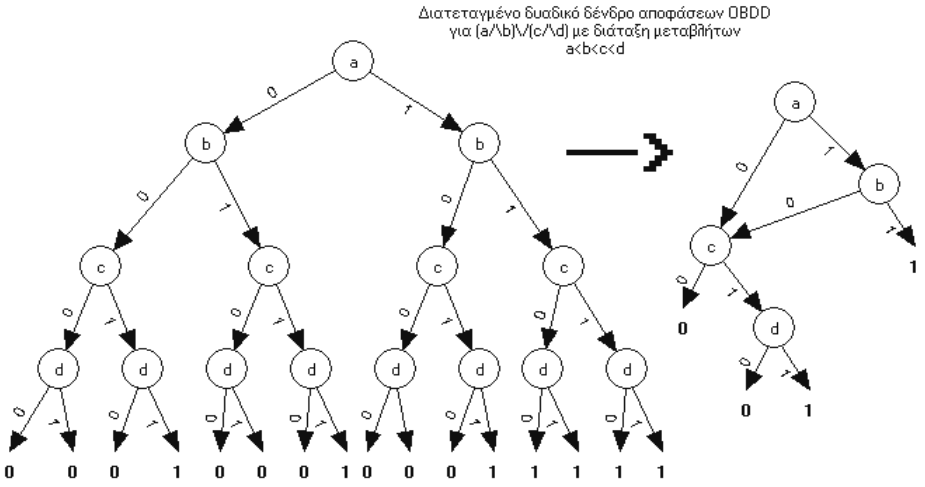
```
void main (void)
{
int i = 0;
while (1) i++;
}
```

Κατά την εκτέλεση αυτού, η `main` θα παρουσιάσει 232 καταστάσεις σε ένα 32-bit μηχάνημα. Μια προσομοίωση του κομματιού αυτού απλώς θα δημιουργούσε

έναν ατέρμονα βρόχο, αλλά ένας ελεγκτής μοντέλων θα έψαχνε να βρει όλες τις πιθανές καταστάσεις του συστήματος.

Οι αλγόριθμοι ελέγχου μοντέλων, όπως αναφέρθηκε και παραπάνω, συναντούν δυσκολίες από το φαινόμενο ΕΧΚ. Παρακάτω θα συζητηθούν μερικές περιπτώσεις λύσεις οι οποίες μπορούν να μετριάσουν το φαινόμενο αυτό. Η συμβολική αναπαράσταση βασίζεται στην αναπαράσταση ενός πεπερασμένου μοντέλου που εκφράζει ένα σύστημα. Η πιο συνηθισμένη αναπαράσταση πραγματοποιείται με μία τεχνική η οποία αναφέρεται ως μια επαρκής κωδικοποίηση συναρτήσεων τύπου Boolean γνωστή ως OBDD, (Ordered Binary Decision Diagrams) [29]. Οι αναπαραστάσεις OBDD έχουν τρία κύρια πλεονεκτήματα:

- Είναι αρκετά μικρές για την αναπαράσταση μεγάλων κλάσεων
- Είναι κανονικοποιημένες για μια δοσμένη διάταξη μεταβλητών εισόδου
- Μπορούν απευθείας να επεξεργαστούν κάτω από όλες τις βασικές Boolean συναρτήσεις



Εικόνα 2.2.2 Διατεταγμένο δυαδικό δένδρο αποφάσεων

Ένα OBDD είναι παρόμοιο με ένα δυαδικό δένδρο αποφάσεων ,με την διαφορά ότι η δομή του αποτελεί έναν απευθείας μη-κυκλικό γράφο παρουσιάζοντας μία αυστηρή συμπεριφορά διάταξης πάνω στις μεταβλητές ,καθώς γίνεται προσπέλαση του γράφου από την ρίζα προς τα φύλλα. Ειδικότερα ,η αναπαράσταση OBDD μιας Boolean συνάρτησης f, γίνεται με την μείωση μιας σχετικής δομής που ονομάζεται δυαδικό δένδρο αποφάσεων. Η

απόκτηση πραγματικής τιμής μπορεί να γίνει με την προσπέλαση του δένδρου από την ρίζα ως τα φύλλα. Σε κάθε κόμβο ,η τιμή της κάθε μεταβλητής είναι αυτή η οποία θα αποφασίσει για το ποιο μονοπάτι θα ακολουθηθεί στην συνέχεια : μπορείς να κατέβεις στο δένδρο σε αριστερό/δεξιό παιδί εάν η τιμή της μεταβλητής που φέρει ο κόμβος είναι αντίστοιχα λάθος/σωστό ,(τιμή 0/1). Η μεταβλητές τοποθετούνται στο δένδρο σε αύξουσα σειρά από την ρίζα προς τα φύλλα. Το δυαδικό δένδρο αποφάσεων μπορεί να μειωθεί σε ένα OBDD με τον συνδυασμό οποιονδήποτε ισομορφικών υποδένδρων σε ένα απλούστερο, εξαλείφοντας με αυτόν τον τρόπο κόμβους όπου τα αριστερά και τα δεξιά τους παιδιά είναι όμοια, εικόνα 2.2.2.

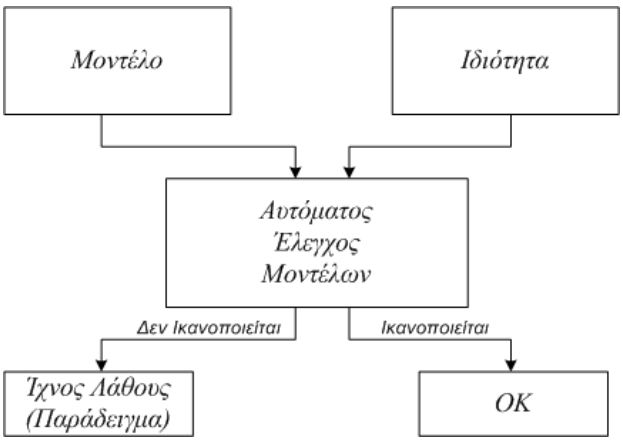
Το πεπερασμένο μοντέλο καταστάσεων ενός συστήματος μπορεί να εκφραστεί με την μορφή OBDD ως εξής : Κάθε κατάσταση κωδικοποιείται με μία ανάθεση τιμών τύπου Boolean σε μια ομάδα μεταβλητών καταστάσεων του συστήματος. Εάν οι μεταβλητές δεν παίρνουν τιμές 0 και 1 αλλά ορίζονται μέσα σε ένα πεδίο D τότε πρέπει να γίνει χρήση μιας συγκεκριμένης δυαδικής κωδικοποίησης πάνω στο πεδίο τιμών D. Αυτή η διεργασία μπορεί να γίνει χωρίς επενέργεια του χρήστη αλλά με εργαλεία τα οποία υποστηρίζουν συμβολική αναπαράσταση (για παράδειγμα το εργαλείο nuSMV [27]). Στη συνέχεια με τον ορισμό της σχέσης μετάβασης, μπορεί η διεργασία να εκφραστεί σαν μια Boolean συνάρτηση δύο μεταβλητών. Η σχέση αυτή θα αναπαριστά την κωδικοποίηση της τρέχουσας κατάστασης αλλά και μια δεύτερη η οποία θα κωδικοποιεί την καινούργια κατάσταση. Με την παραπάνω διεργασία σχηματίζεται μια OBDD αναπαράσταση.

2.3 Αυτόματος Έλεγχος Μοντέλων

Ο αυτόματος έλεγχος μοντέλων αποτελεί ένα μεγάλο μέρος της οικογένειας των τυπικών μεθόδων ανάλυσης συστημάτων. Πρόκειται για αυτοματοποιημένη τεχνική η οποία προτάθηκε αρχικά στα [29][31][44], στις αρχές της δεκαετίας του 80, και στη συνέχεια αναπτύχθηκε ραγδαία μέχρι σήμερα. Αποκορύφωμα ήταν η βράβευση με το βραβείο Turing των Clarke, Queille και Sifakis το 2008, οι οποίοι θεμελίωσαν τις αρχές του ελέγχου μοντέλων στις ερευνητικές τους

εργασίες. Η βασική λειτουργία του ελέγχου μοντέλων απεικονίζεται στην εικόνα 2.3.1.

Δεδομένου του ελέγχου ενός συστήματος (λογισμικού), για πιθανά λάθη ή για επαλήθευση της ορθής του λειτουργίας, ο αναλυτής αποφασίζει να χρησιμοποιήσει έναν ελεγκτή μοντέλων. Στην αρχή θα πρέπει να καταγραφούν οι προδιαγραφές του συστήματος οι οποίες περιλαμβάνουν τις βασικές λειτουργίες του προγράμματος που θα πρέπει να μοντελοποιηθούν. Για την υλοποίηση του μοντέλου, το οποίο θα πρέπει με ακρίβεια να αναπαριστά τις βασικές λειτουργίες του συστήματος, απαιτείται η καταγραφή των όποιων υποθέσεων λαμβάνει ο αναλυτής αλλά και η αφαιρετικότητα [31] με την οποία περιγράφονται πολύπλοκες διεργασίες του συστήματος. Στη συνέχεια, με την βοήθεια ειδικής γλώσσας προδιαγραφών που ορίζει ο ελεγκτής μοντέλων ορίζονται οι ιδιότητες του συστήματος που θέλουμε να επαληθεύσουμε έναντι στο μοντέλο που αναπαριστά το σύστημα.



Εικόνα 2.3.1 Αυτόματος Έλεγχος Μοντέλων

Μετά από αυτή τη διαδικασία το πρόγραμμα επαληθεύεται ξανά. Ένα ‘ίχνος λάθους’ (error-trace) μπορεί να προκύψει από λανθασμένη μοντελοποίηση ή από παραβίαση μιας εκ των ιδιοτήτων που επαληθεύουμε. Σε αυτές τις δύο περιπτώσεις η επαλήθευση δε θα τερματίσει κανονικά και θα χρειαστεί να επαναληφθεί αφού αλλαχθεί και διορθωθεί το μοντέλο, απαλείφοντας με αυτό τον τρόπο το εντοπιζόμενο το πρόβλημα.

2.3.1 Βασικά Στάδια στον έλεγχο μοντέλων

Ο έλεγχος μοντέλων περιλαμβάνει τρία στάδια:

1. Μοντελοποίηση: Σε αυτό το στάδιο πρέπει να μετατραπεί ένα σχέδιο σε ένα φορμαλισμό, ο οποίος να γίνεται δεκτός από έναν ελεγκτή μοντέλων. Στις περισσότερες περιπτώσεις αρκεί μία μεταγλώττιση προδιαγραφών. Σε κάποιες άλλες όμως, εξαιτίας του περιορισμού σε χρόνο και μνήμη, η μετατροπή του σχεδίου απαιτεί τη χρησιμοποίηση αφηρημένων εννοιών για την απομάκρυνση λιγότερο σημαντικών λεπτομερειών. Με την τεχνική αυτή αποφεύγονται οι μοντελοποιήσεις μηχανισμών ή διεργασιών όπου η πολυπλοκότητά τους μπορεί να οδηγήσει στο φαινόμενο ΕΧΚ.
2. Ορισμός Ιδιοτήτων: Πριν το στάδιο της επαλήθευσης, είναι απαραίτητο να καθορίσουμε τις προδιαγραφές που το μοντέλο πρέπει να ικανοποιεί. Ο καθορισμός αυτός δίνεται συνήθως σε κάποια μορφή λογικού φορμαλισμού ή γλώσσας προδιαγραφών. Για συστήματα λογισμικού χρησιμοποιούμε χρονικές λογικές (temporal logic), με την οποία υποθέτοντας τη συμπεριφορά του συστήματος σε σχέση με το χρόνο, ορίζουμε τις ιδιότητες που το μοντέλο μας θέλουμε να ικανοποιεί. Ένα σημαντικό στοιχείο στον ορισμό των ιδιοτήτων (specification definement) είναι η ολοκληρωσιμότητα. Ο έλεγχος μοντέλων παρέχει τη δυνατότητα ελέγχου, εάν ένα σχέδιο ικανοποιεί ένα συγκεκριμένο καθορισμό, αλλά είναι αδύνατο να επιστρέχει με σιγουριά, μια έξοδο ότι ο δοθέν καθορισμός καλύπτει όλες τις ιδιότητες που πρέπει να ικανοποιεί το σύστημα.
3. Επαλήθευση: Θεωρητικά το στάδιο της επαλήθευσης είναι αυτόματο, αλλά στην πράξη απαιτούνται και ενέργειες από το χρήστη. Μία τέτοια ενέργεια είναι η ανάλυση των αποτελεσμάτων. Σε περίπτωση αρνητικού αποτελέσματος ο χρήστης ειδοποιείται από το εργαλείο ελέγχου με ένα ίχνος λάθους (error trace). Το τελευταίο μπορεί να αποβεί πολύ χρήσιμο στο σχεδιαστή του προγράμματος που ελέγχθηκε, ο οποίος μπορεί να δει που ακριβώς είναι το λάθος και να το διορθώσει.

2.3.2 Επαλήθευση με την μέθοδο On-the-fly

Είναι γνωστό ότι η ανάλυση προσεγγισιμότητα (reachability analysis) αποτελεί μια τεχνικής επαλήθευσης η οποία εφαρμόζει μια εξαντλητική εξερεύνηση όλων

εκείνων των καταστάσεων και μεταβάσεων που φτάνει ένα σύστημα κατά την εκτέλεσή του. Οι On-the-fly τεχνικές [46][93], ναι μεν βασίζονται στην εφαρμογή της ανάλυσης αυτής, αλλά δεν υποχρεώνονται να αποθηκεύσουν ολόκληρο τον γράφο καταστάσεων του συστήματος. Στην περίπτωση ειδικά του φαινομένου της έκρηξης του χώρου των καταστάσεων, θα ήταν αδύνατο να αποθηκευθεί ένας τέτοιος μεγάλος γράφος (λ.χ. με δισεκατομμύρια καταστάσεις). Την λύση στο πρόβλημα θα έδινε μια προσομοίωση όλων των δυνατόν μεταβάσεων που είναι ικανό το σύστημα να παρουσιάσει. Έτσι, μια απλή αναζήτηση πρώτα σε βάθος (depth-first search, DFS) μπορεί να χρησιμοποιηθεί για να εξερευνηθεί το σύστημα με την μέθοδο On-the-fly, δηλαδή χωρίς την αποθήκευση των μεταβάσεων που λαμβάνουν χώρα κατά την διάρκεια της αναζήτησης. Όταν εφαρμόζεται η αναζήτηση DFS σε έναν γράφο, ο ελάχιστος χώρος αποθήκευσης που απαιτείται ,είναι εκείνος ο χώρος του συγκεκριμένου μονοπατιού που εξετάζεται. Κάτι τέτοιο είναι δυνατόν να εξασφαλίσει μείωση της απαιτούμενης μνήμης ενώ παράλληλα εγγυάται για μια εξαντλητική εξερεύνηση στον χώρο καταστάσεων. Παρόλο αυτά όμως, ο χρόνος που χρειάζεται για την επαλήθευση του συστήματος, μπορεί να μεγαλώσει δραματικά εξαιτίας της εξερεύνησης του αλγορίθμου σε καταστάσεις που ήδη έχει επισκεφθεί. Κάτι τέτοιο θα λύνονταν μόνο στην περίπτωση της αποθήκευσης όλων των επισκεφθέντων καταστάσεων μόλις αυτές προσπελάζονται.

Αλλά στην περίπτωση μεγάλων γράφων προσεγγισιμότητας, θα ήταν αδύνατον να αποθηκευθούν όλες οι καταστάσεις του συστήματος. Αρκετές είναι οι τεχνικές που έχουν προταθεί και οι οποίες προσπαθούν να συνεργαστούν με τις παραπάνω στρατηγικές. Έτσι, για την αποθήκευση του τρέχοντος μονοπατιού, μία μνήμη (cache) δημιουργείται για την αποθήκευση επιλεγμένων επισκεφθέντων καταστάσεων. Αρχικά όλες οι καταστάσεις που έχουν επισκεφθεί θα αποθηκεύονται σε αυτήν την μνήμη ,μέχρι αυτή να γεμίσει. Όταν συμβεί αυτό, οι πιο παλιές καταστάσεις αντικαθίστανται από καινούργιες. Η αποτελεσματικότητα της μνήμης αυτής (state-space caching), εξαρτάται από το μέγεθός της αλλά επίσης και από την δομή του χώρου καταστάσεων. Το τελευταίο αποτελεί αρκετά δύσκολο εγχείρημα, αφού είναι αρκετά δύσκολη η πρόβλεψη της δομής αυτής. Επίσης μια τέτοια πρόβλεψη θα μπορούσε να οδηγήσει σε αδιέξοδο όσον αφορά την επιλογή της cache μνήμης, αφού μια

λανθασμένη επιλογή θα μεγάλωνε παρά πολύ τον χρόνο της εκτέλεσης της αναζήτησης.

Μια τεχνική η οποία μπορεί να εφαρμοστεί στην περίπτωση όπου το μέγεθος του προβλήματος θεωρείται πολύπλοκο, απαγορεύοντας την εξαντλητική επαλήθευση ,είναι και αυτή με το όνομα bit-state hashing ή αλλιώς supertrace και η οποία εφαρμόζει μια μερική αναζήτηση του χώρου καταστάσεων. Στην τεχνική αυτή, όλες οι καταστάσεις που έχουν επισκεφθεί ,αποθηκεύονται σε έναν πίνακα κατακερματισμού (hash) *H*, του οποίου το μέγεθος εξαρτάται από την μνήμη που είναι ελεύθερη. Για κάθε κατάσταση *s*, χρησιμοποιείται ένα απλό *bit* με διεύθυνση *h(s*), όπου το *h* είναι μια συνάρτηση κατακερματισμού ,η οποία επιστρέφει τις *bit*-διευθύνσεις στο *H*. Εάν το bit στην διεύθυνση *h(s)* έχει τιμή 1, τότε ο αλγόριθμος αναζήτησης υποθέτει ότι η κατάσταση *s* έχει ήδη επισκεφθεί. Έτσι από την στιγμή που δεν ανιχνεύεται κάποια σύγκρουση των καταστάσεων, η αναζήτηση γίνεται μερική. Επίσης αξίζει να σημειωθεί ότι τεχνική μπορεί να επεκταθεί με την χρησιμοποίηση αυτής για αρκετές φορές, όπου σε κάθε μια από αυτές θα εφαρμόζονται διαφορετικές συναρτήσεις κατακερματισμού (μέχρι το σημείο εκείνο που το επίπεδο κάλυψης του αλγορίθμου θα είναι ικανοποιητικό). Κάτι τέτοιο δεν αποτελεί πρόβλημα, αφού ο περιοριστικός παράγοντας στην ανάλυση προσεγγισιμότητας είναι συνήθως ο χώρος και όχι ο χρόνος.

Κατά παράδοση, η ανάλυση αυτή εφαρμόζεται με επιτυχία στην ανίχνευση λαθών, όπως για παράδειγμα σε περιπτώσεις αδιέξοδων (deadlocks) η κώδικα που δεν εκτελείται (unreachable code). Επιπλέον, η εφαρμογή των αλγόριθμων της ανάλυσης προσεγγισιμότητας, επεκτείνεται με την ανάπτυξη του έλεγχου μοντέλων που στηρίζονται στην θεωρία αυτομάτων. Για παράδειγμα, ο έλεγχος μοντέλων με την μέθοδο της γραμμικής χρονικής λογικής (Linear Temporal Logic, LTL), μπορεί να μειωθεί σε αυτήν της ανάλυσης προσεγγισιμότητας. Επομένως είναι δυνατόν να παρέχονται αλγόριθμοι οι οποίοι να πραγματοποιούν έλεγχο μοντέλων με την τεχνική On-the-fly. Αυτοί οι αλγόριθμοι χρησιμοποιούνται για την εφαρμογή της On-the-fly επαλήθευσης και είναι συμβατοί με τεχνικές διαχείρισης πολυπλοκότητας, όπως οι bit-state hashing και state-space caching.

Ένα πλεονέκτημα της On-the-fly επαλήθευσης είναι ότι αυτή προχωράει μέχρι το σημείο όπου βρεθεί λάθος, και που στην περίπτωση αυτή παράγει ένα αντιπαράδειγμα (counter-example) το οποίο βοηθά τον σχεδιαστή να κατανοήσει καλύτερα το λάθος και να το διορθώσει. Συχνά τα λάθη βρίσκονται πολύ νωρίς από το σημείο εκκίνησης της αναζήτησης, κάτι που σημαίνει ότι μπορεί να αποφευχθεί η αναζήτηση του υπολοίπου χώρου καταστάσεων. Αντίθετα, στην περίπτωση που το σύστημα παρουσιάζεται σωστό, η αναζήτηση καλύπτει όλο το φάσμα του χώρου καταστάσεων. Μπορούμε έτσι να συμπεράνουμε ότι η προσέγγιση αυτή ταιριάζει στις περιπτώσεις που βρισκόμαστε στα αρχικά σταδία σχεδίασης ενός συστήματος, στα οποία είναι λογικό να παρουσιάζονται αρκετά λάθη.

2.3.3 Η τεχνική της Μείωσης (Reduction technique)

Η τεχνικές της μείωσης επικεντρώνονται στο να αποθηκεύουν ένα μέρος ή ένα στιγμιότυπο του χώρου καταστάσεων ενός προγράμματος, διατηρώντας όλες τις ιδιότητες του χώρου αυτού ,για να μπορεί στην συνέχεια να επαληθεύσει τις ιδιότητες που επιθυμεί [29][31]. Στην επόμενη παράγραφο περιγράφονται οι κύριες προσεγγίσεις της τεχνικής μείωσης του χώρου καταστάσεων (state-space reduction).

2.3.4 Μερική Διατεταγμένη Μείωση (Partial-order reduction)

Στις περισσότερες τεχνικές ελέγχου μοντέλων, το φαινόμενο διάφορων συντρέχουσων διεργασιών μοντελοποιείται με την παρεμβολή (interleaving), κάτι που αποτελεί μεγάλο παράγοντα όσον αφορά το πρόβλημα της έκρηξης του χώρου των καταστάσεων. Η τεχνική της μερικής μείωσης βασίζεται στην συνεχή παρατήρηση των συντρέχουσων (concurrent) συστημάτων όπου η συνολική επίδραση συνόλου ενεργειών συχνά είναι ανεξάρτητη της σειράς που εκτελούνται [30]. Σαν αποτέλεσμα, η δημιουργία όλων των δυνατών παρεμβολών μεταξύ ενεργειών του συστήματος, καθιστά κάποιες από αυτές άχρηστες, γι αυτό και μπορούν να παραλειφθούν. Αρκετές μέθοδοι έχουν προταθεί βασιζόμενες στην παραπάνω ιδέα ,οι οποίες εξερευνούν έναν μειωμένο γράφο ενός συστήματος διατηρώντας παράλληλα τις βασικές ιδιότητες αυτού.

Οι μέθοδοι της μερικής μείωσης εφαρμόζουν μια επιλεκτική αναζήτηση στον χώρο καταστάσεων του συστήματος [59]. Για κάθε κατάσταση *s* που προσπελάσετε κατά την αναζήτηση ,υπολογίζεται ένα υποσύνολο *T* του συνόλου μεταβάσεων του *s* και εξετάζονται μόνο εκείνες οι μεταβάσεις που αφορούν το *T*. Αυτή είναι και η κύρια διαφορά τους με τις κλασικές αναζητήσεις όπου για κάθε κατάσταση *s* να εξετάζονταν όλες οι μεταβάσεις που θα ήταν δυνατόν να επέλθουν, μέσω της *s*.

Δύο βασικές τεχνικές έχουν προταθεί για την αναγνώριση αυτού του υποσυνόλου οι οποίες βασίζονται στον υπολογισμό των επίμονων συνόλων (persistent sets) και των αδρανή συνόλων (sleep sets). Ένα persistent set *T* για κάποιες καταστάσεις *s*, περιέχει μεταβάσεις ενεργές στο *s* με το παρακάτω χαρακτηριστικό: κάθε μετάβαση που είναι δυνατή να συμβεί από το *s* εφαρμόζοντας μεταβάσεις έξω από το υποσύνολο *T*, είναι ανεξάρτητο (δεν υπάρχει αλληλεπίδραση) από τις μεταβάσεις του *T*. Μία από τις βασικές τεχνικές τύπου persistent set, προτάθηκαν από τον Valmari (1993) [96] και βασίζονται στον προσδιορισμό των stubborn sets.

Κατά την διάρκεια της μειωμένης πλέον εξερεύνησης του χώρου καταστάσεων του συστήματος, επιλέγονται μόνο οι καταστάσεις εκείνες που ανήκουν στο stubborn set. Έχει αποδειχθεί ότι η εκτέλεση όλων των εναπομείναντα μεταβάσεων (που δεν έχουν επιλεχθεί), μπορεί να αναβληθεί χωρίς αυτό να επιδράσει τα αποτελέσματα της επαλήθευσης. Ο σκοπός έτσι του stubborn set είναι να παραμείνει αυτό όσο το δυνατόν πιο μικρό σε μέγεθος έτσι ώστε να επιτευχθεί μία μεγάλη μείωση του χώρου καταστάσεων. Ο αλγόριθμος που περιγράφει ο Valmari υπολογίζει τα stubborn sets κατά την διάρκεια της εξερεύνησης του χώρου καταστάσεων και μπορεί να εφαρμοστεί με την μέθοδο On-the-fly.

Η τεχνική των sleep sets [94], εκμεταλλεύεται το ιστορικό της αναζήτησης. Με την χρήση της, μειώνει τον αριθμό των μεταβάσεων που εξερευνούνται αλλά όχι και τον αριθμό των καταστάσεων. Όπως προαναφέρθηκε, κάτι τέτοιο είναι χρήσιμο όταν τα sleep sets συνδέονται με τεχνικές του χώρου καταστάσεων με μνήμη (state-space caching). Κατά την διάρκεια της αναζήτησης depth-first στον γράφο του συστήματος, κάθε κατάσταση *s* σχετίζεται με ένα sleep set, το οποίο αποτελεί ένα σύνολο μεταβάσεων που μπορούν να γίνουν από το *s* αλλά δεν θα

εκτελούνται από αυτό. Τα sleep sets μπορούν να συνδυαστούν ακόμη και με τα persistent sets για να μειώσουν ακόμη περισσότερο ο χώρος των καταστάσεων. Επιπρόσθετα, αφού η τεχνική των persistent sets δεν μπορεί να αποφύγει την επιλογή ανεξάρτητων μεταβάσεων μιας κατάστασης, τα sleep sets αποφεύγουν την εξερεύνηση πολλαπλών παρεμβολών αυτών των μεταβάσεων. Ο Godefroid [39] προτείνει τεχνικές μερικής μείωσης για την ανίχνευση και επαλήθευση της απουσίας αδιεξόδου και ιδιοτήτων ασφάλειας (safety properties). Σύμφωνα με την περιγραφή των παραπάνω, ο έλεγχος των ιδιοτήτων ασφάλειας γίνεται (ή μειώνεται) στην ανίχνευση αδιεξόδων του συστήματος. Πιο πρόσφατες τεχνικές έχουν προταθεί και οι οποίες επεκτείνουν τις αρχικές τεχνικές μερικής μείωσης, φέρνοντας μπροστά όλες τις δυνατότητες και των οφελών που απορρέουν από τον έλεγχο μοντέλων. Κάποιες από αυτές εφαρμόζουν έλεγχο μοντέλων σε τύπους LTL οι οποίοι δεν περιέχουν τον τελεστή "next time", *X*. Η συγκεκριμένη τεχνική, μπορεί να χειριστεί κάθε είδους λογική LTL, καθώς και επίσης επεκτάσεις αυτής. Αυτή η προσέγγιση χρησιμοποιεί τεχνικές που βασίζονται στην θεωρία αυτομάτων περιλαμβάνοντας ακόμη επεκτάσεις αυτών όπως τα ω-αυτόματα. Με το συνδυασμό του ελέγχου μοντέλων με την τεχνική της μείωσης, η δεύτερη ακολουθεί πάντα τους κανόνες που θέτει η υπό-επαλήθευση ιδιότητα. Πρόκειται για την περίπτωση όπου η τεχνική της μερικής μείωσης προσπαθεί να υπολογίσει, κατά την διάρκεια της αναζήτησης, τα μέρη εκείνα του γράφου καταστάσεων που πλεονάζουν και μπορούν να παραλειφθούν. Περισσότερες πληροφορίες για την ορισμό ιδιοτήτων με την χρονική λογική LTL, ο αναγνώστης μπορεί να ανατρέξει στο Παράρτημα Α της παρούσας διατριβής.

2.4 Ο αυτόματος ελεγκτής μοντέλων SPIN

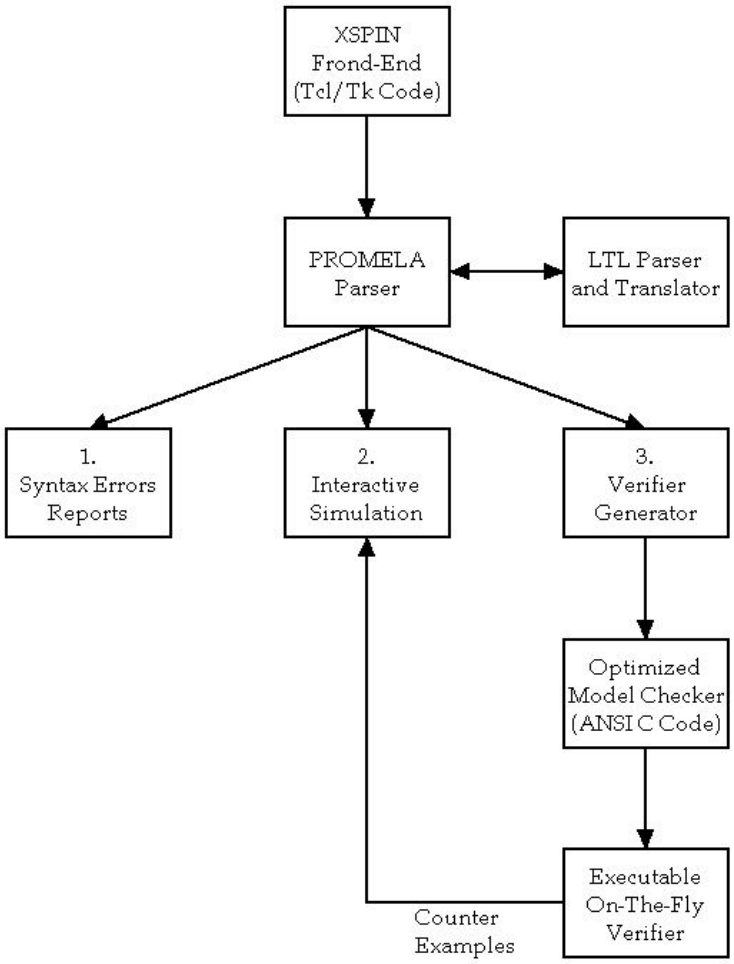
Το SPIN (Simple PROMELA Interpretation) είναι ένα ευρέως διαδεδομένο λογισμικό για την επαλήθευση κατανεμημένων συστημάτων. Αναπτύχθηκε στα Bell Labs περίπου στις αρχές της δεκαετίας 1980 [46][93]. Τα μοντέλα επαλήθευσης του SPIN επικεντρώνονται στην απόδειξη της ορθότητας των αλληλεπιδράσεων επεξεργασίας και προσπαθούν να απεικονίσουν, όσο το δυνατό περισσότερο, τους εσωτερικούς ακολουθιακούς υπολογισμούς. Οι αλληλεπιδράσεις επεξεργασίας μπορούν να οριστούν στο SPIN με ασύγχρονο

πέρασμα μηνυμάτων μέσω καναλιών buffer, ή με κοινές μεταβλητές, ή τέλος με οποιοδήποτε συνδυασμό από τις δύο προηγούμενες μεθόδους. Για να επαληθεύσουμε έναν αλγόριθμο ή πρωτόκολλο με το SPIN, πρέπει να το γράψουμε στην PROMELA, τη γλώσσα μοντελοποίησης που χρησιμοποιεί το SPIN. Η PROMELA (PROcess MEta LAnguage) είναι μια μη ντετερμινιστική γλώσσα, που χρησιμοποιεί τρεις τύπους αντικειμένων:

- ενέργειες: είναι καθολικά αντικείμενα
- μεταβλητές: τοπικές ή καθολικές μέσα σε μία ενέργεια
- κανάλια μηνυμάτων (message channels): τοπικά ή καθολικά σε μία ενέργεια

Ο ευκολότερος τρόπος να ξεκινήσει κανείς να δουλεύει με το SPIN είναι να χρησιμοποιήσει το γραφικό περιβάλλον του, το XSPIN. Το τελευταίο, «τρέχει» ανεξάρτητα από το SPIN και βοηθάει στη δημιουργία των εντολών του SPIN με απλές επιλογές από το μενού. Επίσης, το μεγάλο του πλεονέκτημα είναι ότι το αποτέλεσμα της προσομοίωσης ή της επαλήθευσης το παριστάνει γραφικά. Χρησιμοποιώντας το XSPIN, ο χρήστης δε χρειάζεται να απομνημονεύει σειρές εντολών, αλλά επικεντρώνεται περισσότερο στην ουσία που του προσφέρει το SPIN μέσω του XSPIN. Το SPIN μπορεί να χρησιμοποιηθεί με τους ακόλουθους τρεις τρόπους: ως προσομοιωτής, επιτρέποντας τρία είδη προσομοίωσης, ανάλογα με τις ανάγκες του χρήστη και του αλγορίθμου ή πρωτοκόλλου που χρησιμοποιείται, την τυχαία (random), καθοδηγημένη (guided) και την διαλογική (interactive). ως εξαντλητικός (exhaustive) αναλυτής χώρου καταστάσεων, ικανός να αποδείξει την εγκυρότητα των χαρακτηριστικών που εισήγαγε ο χρήστης κάνοντας χρήση της θεωρίας partial order reduction,που αναλύθηκε παραπάνω. ως αναλυτής bit-state χώρου, ικανός να επαληθεύσει και τα μεγαλύτερα πρωτόκολλα με τη μέγιστη έκταση του χώρου καταστάσεων. Η βασική δομή του SPIN απεικονίζεται στο παρακάτω σχήμα (εικόνα 2.4.1).

Ο τυπικός τρόπος για να αρχίσει κανείς να δουλεύει με το SPIN ή με το γραφικό του περιβάλλον XSPIN είναι να μοντελοποιήσει καταρχήν, τα βασικά χαρακτηριστικά-διεργασίες ενός παράλληλου συστήματος ή ενός κατανεμημένου αλγορίθμου. Μετά τη διόρθωση όποιων συντακτικών λαθών, αν βέβαια υπάρχουν, το σύστημα εισάγεται σε προσομοίωση, μέχρι να βεβαιωθούμε ότι συμπεριφέρεται όπως αναμένεται.



Εικόνα 2.4.1 Η δομή του αυτόματου ελεγκτή μοντέλων SPIN

Έπειτα, το SPIN χρησιμοποιείται για να παράγει μία on-the-fly επαλήθευση του προγράμματος. Η επαλήθευση εκτελείται με διαφορετικές επιλογές κάθε φόρα και αν οδηγήσει σε σφάλμα, τότε ο αναλυτής επιστρέφει ξανά στο στάδιο της προσομοίωσης για να εντοπίσει το σημείο εκείνο που προκάλεσε το σφάλμα. Αν είναι εφικτό το διορθώνει και εκτελέι εκ νέου την επαλήθευση.

2.4.1 Προτεραιότητες του εργαλείου

Στον έλεγχο μοντέλων, σε αντίθεση με το συμβατικό προγραμματισμό, οι απαιτήσεις σε χώρο είναι πιο σημαντικές από τις απαιτήσεις σε χρόνο. Για το λόγο αυτό θα πρέπει να λαμβάνονται σοβαρά υπό όψιν τα ακόλουθα:

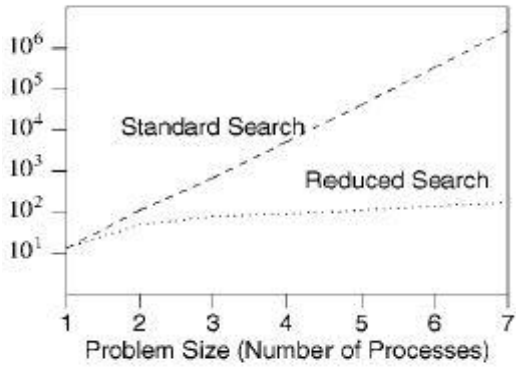
- *Αριθμός καταστάσεων*: Εξαιτίας του προβλήματος της έκρηξης του χώρου καταστάσεων θα πρέπει να μειώνεται ο αριθμός των καταστάσεων.

- *Μέγεθος διανύσματος κατάστασης*: Δηλώνει το ποσό της μνήμης που απαιτείται και πρέπει να ελαχιστοποιείται για την αύξηση της απόδοσης.
- *Μέγεθος στοίβας αναζήτησης*: Απαιτείται και εδώ ελαχιστοποίηση.
- *Χρόνος επαλήθευσης*: σημαντικό στοιχείο που πρέπει να απασχολεί τον αναλυτή ιδιαίτερα κατά την επαλήθευση μεγάλων-πολύπλοκών συστημάτων λογισμικού.

2.4.2 Το εργαλείο SPIN και η τεχνική της μερικής μείωσης

Το εργαλείο SPIN χρησιμοποιεί την τεχνική της μερικής μείωσης για την μείωση του αριθμού των προσβάσιμων καταστάσεων που πρέπει να εξερευνηθούν για την ολοκλήρωση της επαλήθευσης. Η μείωση αυτή βασίζεται πάνω στην παρατήρηση ότι η ορθότητα της LTL πρότασης αποτελεί έχει να κάνει με την σειρά με την οποία μεταχειρίζονται οι χώροι καταστάσεων των μοντελοποιημένων συστημάτων (concurrent ή ανεξάρτητα εκτέλεσης συστήματα) στην διάρκεια της πρώτα-σε-βάθος (depth-first) αναζήτησης. Αντί το εργαλείο να παράγει (μέσω του μοντέλου) έναν εξαντλητικό χώρο αναζήτησης που να περιλαμβάνει όλες τις ακολουθίες εκτέλεσης σαν ‘μονοπάτια’, ο επαληθευτής που δημιουργείται παράγει έναν μειωμένο χώρο περιλαμβάνοντας μονάχα εκείνες τις καταστάσεις που είναι αντιπροσωπευτικές για την προδιαγραφή του συστήματος που επαληθεύεται.

Η εφαρμογή αυτής της τεχνικής χαρακτηρίζεται από μια στατική μείωση, όπως περιγράφθηκε και στο παρών κεφάλαιο που πριν την έναρξη της πραγματικής επαλήθευσης αναγνωρίζει τις περιπτώσεις εκείνες όπου με ασφαλή τρόπο μπορεί να γίνει μείωση των καταστάσεων.



Εικόνα 2.4.2 Διάγραμμα χώρου αναζήτησης σε σχέση με τον αριθμό των διεργασιών κατά τον έλεγχο μοντέλων με την κανονική αναζήτηση και την αναζήτηση με POR

Στην εικόνα 2.4.2 δίνεται ένα παράδειγμα μέτρησης του αριθμού των προσβάσεων στις καταστάσεις που πρέπει να παραχθούν από το εργαλείο για το πέρας της επαλήθευσης ενός αλγόριθμου επιλογής. Παρουσιάζει την όσο το δυνατόν καλύτερη προσέγγιση του αλγορίθμου όπου οι καταστάσεις αρχικά αυξάνονται με εκθετικό τρόπο και με την τεχνική της μείωσης αυξάνονται με γραμμικό τρόπο. Τέλος ένα άλλο σημαντικό χαρακτηριστικό αυτής της στατικής μεθόδου μείωσης του SPIN είναι ότι δεν χρειάζεται αρκετούς πόρους σε μνήμη σε αντίθεση με τις εξαντλητικές μεθόδους.

2.4.3 Process Meta Language (PROMELA)

Θα ήταν σωστό πριν περάσουμε στο παράδειγμα που ουσιαστικά δείχνει την λειτουργία του εργαλείου, να γίνει μια αναφορά στην γλώσσα και στις τεχνικές που χρησιμοποιεί αυτή για την μοντελοποίηση συστημάτων. Ειδική έμφαση δίνεται στις διεργασίες που ορίζονται και οι οποίες αποτελούν απαραίτητα στοιχεία για αναπαράσταση. Όπως αναφέρθηκε, το εργαλείο SPIN είναι ικανό να ελέγχει μοντέλα τα οποία αναπαρίστανται από μια γλώσσα αρκετά διαφορετική από τις γλώσσες που χρησιμοποιούν σήμερα οι προγραμματιστές. Πρόκειται για μια γλώσσα αναπαράστασης και μοντελοποίησης με το όνομα PROMELA, που χρησιμοποιείται κυρίως για την αφηρημένη μοντελοποίηση πρωτοκόλλων επικοινωνίας. Η PROMELA ταιριάζει απόλυτα επίσης στην μοντελοποίηση πρακτόρων επικοινωνίας. Η «συζητήσεις» μεταξύ πρακτόρων αναπαρίστανται σαν διεργασίες, τα μονοπάτια αυτών των συζητήσεων μοντελοποιούνται σαν κανάλια και οι μεταβλητές που χρησιμοποιούνται ορίζονται και ελέγχονται χωρίς καμία αλλαγή. Όλες οι δηλώσεις είναι είτε εκτελέσιμες είτε μπλοκάρονται από άλλες δηλώσεις περιμένοντας με την σειρά τους να εκτελεστούν. Έτσι, προτάσεις μπορεί να μπλοκαριστούν από μία εντολή `if` εάν η συνθήκη αυτής είναι `false`. Οι προτάσεις αυτές μπορούν να εκτελεστούν την στιγμή εκείνη που η συνθήκη θα γίνει `true`. Κάτι τέτοιο παρέχει μηχανισμούς συγχρονισμού επικοινωνιών μεταξύ διεργασιών έχοντας ως αποτέλεσμα μια εκτελέσιμη διεργασία (`responder`) να περιμένει για ένα μήνυμα από μια άλλη διεργασία (`initiator`). Όλες οι διεργασίες στην PROMELA ορίζονται με την λέξη proctype. Παρακάτω ακολουθεί ένα παράδειγμα μιας δήλωσης ενός proctype :

```
proctype ProcessA()
{
byte newVariable;
newVariable = 3
}
```

Το όνομα της διεργασίας αυτής είναι το `ProcessA` και μέσα στις αγκύλες περικλείεται το σώμα της δήλωσης αυτής. Παρατηρείται η δήλωση μιας τοπικής μεταβλητής `newVariable` τύπου `byte`, και έναν αρχικό ορισμό της μεταβλητής αυτής δίνοντάς της την τιμή 3. Στην PROMELA τα ερωτηματικά ‘ ; ’ και τα βέλη ‘ ->’ διαχωρίζουν δηλώσεις μεταξύ τους. Στο παράδειγμα δεν χρειάζεται ερωτηματικό ή βέλος στην τελευταία δήλωση. Το βέλος χρησιμοποιείται μερικές φορές και σαν ένα τρόπο για να δειχθεί μια αιτιατή (causal) σχέση μεταξύ δύο δηλώσεων. Για παράδειγμα :

```
byte newVariable = 2;
proctype ProcessA()
{
(newVariable == 1) -> newVariable = 3
}
proctype ProcessB()
{
newVariable = newVariable – 1
}
```

Στο παράδειγμα αυτό δηλώνονται δύο διεργασίες, οι `ProcessA` και `ProcessB`. Μετά την καθολική (`global`) δήλωση της μεταβλητής `newVariable` έξω από τις διεργασίες με την τιμή 2 παρατηρείται ότι η διεργασία Α περιέχει δύο δώσεις στο σώμα της ενώ η Β περιέχει μία δήλωση η οποία και ελαττώνει την `newVariable` κατά 1. Πάντα μια δήλωση είναι εκτελέσιμη και γι αυτό και η διεργασία Β δεν μπλοκάρεται αλλά εκτελείται άμεσα. Εάν η συνθήκη δεν είναι αληθής τότε η διεργασία μπλοκάρεται μέχρι αυτή να γίνει αληθής. Κάτι τέτοιο συμβαίνει στην περίπτωση της διεργασίας Α όπου δεν ισχύει η ισότητα της `newVariable` με 1 οπότε μπλοκάρεται η πρόταση. Μια proctype αποτελεί μια διεργασία. Δεν μπορεί να εκτελεστεί αυτόνομα μόνη της. Μία άλλη δήλωση πρέπει να είναι αυτή η οποία και θα θέσει την διεργασία προς εκτέλεση. Το εργαλείο SPIN κάνει την χρήση μιας διεργασίας με το όνομα `init` για να πυροδοτήσει τις υπόλοιπες διεργασίες. Η διεργασία `init` είναι όμοια με την

διαδικασία `main` στα προγράμματα Java. Ένα παράδειγμα μιας διεργασίας `init` είναι και το παρακάτω :

```
init
{
run ProcessA();
run ProcessB()
}
```

Σε αυτήν την περίπτωση, η λέξη κλειδί `run` πυροδοτεί τις δύο διεργασίες. Θα ήταν σωστό στη συνέχεια να δοθούν κάποιοι βασικοί ορισμοί εννοιών που συναντώνται συχνά στην περίπτωση ενασχόλησης με το θέμα του έλεγχου μοντέλων. Ουσιαστικά πρόκειται για μια περιγραφή των πιο συνηθισμένων προβλημάτων που καλείται να ανίχνευση και να αντιμετωπίσει ο έλεγχος μοντέλων, γι αυτό και η περιγραφή τους θεωρείται αναγκαία.

2.5 Ο πιθανοκρατικός ελεγκτής μοντέλων PRISM

Το PRISM είναι ένα πιθανοκρατικό εργαλείο αυτόματου ελέγχου μοντέλων (model checker), που χρησιμοποιείται για ανάλυση και μοντελοποίηση συστημάτων που παρουσιάζουν συμπεριφορά σχετική με πιθανότητες [91]. Ο έλεγχος βασισμένος σε πιθανότητες είναι μια τυπική μέθοδος επαλήθευσης (formal verification technique) βασιζόμενη στην κατασκευή ενός ακριβούς μαθηματικού μοντέλου ενός συστήματος το οποίο και πρόκειται να αναλυθεί. Το PRISM υποστηρίζει τρεις τύπους μοντέλων με βάση την πιθανοκρατική ανάλυση που στοχεύουμε να πραγματοποιήσουμε, σύμφωνα με [56].

- Μαρκοβιανές Αλυσίδες Διακριτού Χρόνου (Discrete-time Markov chains, DTMCs)
- Μαρκοβιανές Διεργασίες Απόφασης (Markov Decision Processes, MDPs)
- Μαρκοβιανές Αλυσίδες Συνεχούς Χρόνου (Continuous Time Markov Chains, CTMCs)

Τα μοντέλα εκφράζονται στο εργαλείο με τη γλώσσα προδιαγραφών PRISM-meta language που ορίζεται στο [56], μια απλή και υψηλού επιπέδου γλώσσα μοντελοποίησης. Οι ιδιότητες των μοντέλων αναλύονται στο εργαλείο χρησιμοποιώντας την γλώσσα προδιαγραφών (property specification language) του PRISM η οποία βασίζεται σε δύο πιθανοκρατικές χρονικές λογικές [56]. Την

PCTL (Probabilistic Computation Tree Logic), για μοντέλα τύπου DTMC και MDP. Και την CSL (Continuous Stochastic Logic), για τα CTMCs μοντέλα. Το PRISM για να ελέγξει τις ιδιότητες των μοντέλων, είτε χρησιμοποιεί τυπικές μεθόδους επαλήθευσης βασισμένες σε αριθμητικούς υπολογισμούς, είτε ανάλυση με προσομοίωση διακριτών γεγονότων.

Για να αναλύσουμε ένα πιθανοκρατικό μοντέλο που έχει περιγραφεί και κατασκευαστεί με το PRISM πρέπει να ορίσουμε μία ή περισσότερες ιδιότητες του μοντέλου που να μπορέσουν να αξιολογηθούν από το εργαλείο. Στο PRISM αυτό γίνεται χρησιμοποιώντας χρονική λογική και χρησιμοποιώντας ιδιότητες που εκφράζονται σε PCTL για μοντέλα τύπου DTMC και MDP, και σε CSL για μοντέλα τύπου CTMC. Και οι δύο λογικές είναι προεκτάσεις της κλασικής χρονικής λογικής CTL. Στην πραγματικότητα το PRISM υποστηρίζει μια πληθώρα προεκτάσεων και προσθηκών των δύο αυτών λογικών [56][57]. Παρακάτω φαίνονται ορισμένα παραδείγματα ορισμού ιδιοτήτων στο PRISM.

- *P>=1[true U terminate]*, σημαίνει ότι ο αλγόριθμος τερματίζει με επιτυχία.
- *"init"=>P<0.1 [trueU <=100num_errors>5]*, από μια αρχική κατάσταση, *η πιθανότητα να συμβούν περισσότερα των 5 λάθη τις 100 πρώτες μονάδες χρόνου, είναι μικρότερη του 0.1* .
- *"down"=> P>0.75[!"fail" U[1,2]"up"]* ,όταν συμβεί ένας τερματισμός, *η πιθανότητα επανεκκίνησης του συστήματος σε μία ή δύο ώρες, χωρίς να συμβούν περαιτέρω λάθη, είναι μεγαλύτερη του 0.75*.
- *S<0.01[num_sensors<min_sensors],*στη διάρκεια εκτέλεσης, *η πιθανότητα ένας αριθμός από ακατάλληλους αισθητήρες να είναι λειτουργικός ,είναι μικρότερη του 0.75*.

Πρέπει να σημειωθεί ότι οι παραπάνω πιθανότητες είναι ισχυρισμοί που επιβεβαιώνονται ή απορρίπτονται. Αυτό συμβαίνει διότι οι αναφορές στις πιθανότητες σχετίζονται με ένα άνω ή κάτω όριο το οποίο μπορεί να ελεηθεί αν είναι σωστό ή λάθος. Ωστόσο στο PRISM μπορούμε να ορίσουμε ιδιότητες που έχουν αριθμητική τιμή.

Επίσης το PRISM προσφέρει την δυνατότητα να υπολογιστούν τιμές για ιδιότητες ενός εύρους παραμέτρων και τον σχεδιασμό γραφημάτων που σχετίζονται με τα αποτελέσματα των πειραμάτων. Αυτός είναι συνήθως ένας

πολύ χρήσιμος τρόπος για την αναγνώριση συγκεκριμένων προτύπων και γενικών κατευθύνσεων στην συμπεριφορά του συστήματος.

Μια ιδιότητα αξιολογείται σύμφωνα με μια κατάσταση του μοντέλου. Για την παραπάνω γραμματική όλες οι ιδιότητες αντιστοιχούν σε Boolean τιμές. Για κάθε κατάσταση του μοντέλου, μια ιδιότητα είναι είτε true είτε false για αυτήν την κατάσταση. Ισοδύναμα μπορεί να θεωρηθεί πως η ιδιότητα ισχύει ή δεν ισχύει για αυτήν την κατάσταση. Η σημασιολογία είναι η ίδια όπως και στην προτασιακή λογική:

- *true αν είναι true σε όλες τις καταστάσεις*
- *false αν είναι false σε όλες τις καταστάσεις*
- *expr είναι true αν η έκφραση expr είναι true*
- *!prop είναι true αν δεν είναι true η prop*
- *prop1 & prop2 είναι true αν και η prop1 και η prop2 είναι true*
- *prop1 | prop2 είναι true αν η prop1 είτε η prop2 είναι true*
- *prop1 => prop2 είναι αληθής αν η prop1 παράγει την prop2*

Για περισσότερες πληροφορίες για τον έλεγχο μοντέλων με τον ορισμό ιδιοτήτων με πιθανοκρατική λογική, ο αναγνώστης μπορεί να προστρέξει στο Παράρτημα Α της διατριβής αυτής.

2.6 Έλεγχος εγκυρότητας ιδιοτήτων με έλεγχο μοντέλων

Στην παράγραφο αυτή θα περιγραφούν οι πιο συνήθεις ιδιότητες που ελέγχονται με τους υπάρχοντες ελεγκτές μοντέλων. Με τον συνδυασμό και ορισμό κατάλληλων προτάσεων ορθότητας, αλλά και με βάση τις αρχές μοντελοποίησης που ακολουθήθηκαν για την υλοποίηση του υπό εξέταση μοντέλου, οι σχεδιαστές πρωτοκόλλων ασφαλείας μπορούν να ελέγξουν τις ιδιότητες που αυτοί επιθυμούν για τα πρωτόκολλά τους. Παρατίθενται παραδείγματα ορισμού ιδιοτήτων είτε με περιγραφή της ιδιότητας είτε με χρήση της γλώσσας προδιαγραφών PROMELA του ελεγκτή μοντέλων SPIN.

2.6.1 Αδιέξοδο (Deadlock)

Ένα αδιέξοδο αποτελεί μια κατάσταση στην οποία δύο προγράμματα υπολογιστών μοιράζονται την ίδια πηγή και επιδρούν το ένα πάνω στο άλλο με

το να αποτρέπουν την πρόσβαση πάνω σε αυτή την πηγή, έχοντας ως αποτέλεσμα το μπλοκάρισμα και των δύο προγραμμάτων [46]. Όταν τα λειτουργικά συστήματα εκτελούν ένα μόνο πρόγραμμα, τότε όλες οι πηγές του συστήματος είναι ανοιχτές για το πρόγραμμα αυτό. Στην περίπτωση όμως που τα λειτουργικά συστήματα εκτελούν πολλά προγράμματα μαζί, παρεμβαίνει ανάμεσά τους έτσι ώστε τα προγράμματα να ζητούν πηγές δυναμικά. Κάτι τέτοιο μπορεί να οδηγήσει σε αδιέξοδο. Ένα παράδειγμα είναι και το παρακάτω:

- *Το πρόγραμμα_1 ζητά μια πηγή Α και την λαμβάνει 46*
- *Το πρόγραμμα_2 ζητά μια πηγή Β και την λαμβάνει*
- *Το πρόγραμμα_1 ζητά την πηγή Β και περιμένει από το πρόγραμμα_2 να την απελευθερώσει*
- *Το πρόγραμμα_2 ζητά την πηγή Α και περιμένει από το πρόγραμμα_1 να την απελευθερώσει*

Στην περίπτωση αυτή κανένα από τα δύο προγράμματα (ούτε το 1 αλλά ούτε και το πρόγραμμα 2) μπορεί να προχωρήσει μέχρι το άλλο να απελευθερώσει την πηγή του. Το λειτουργικό σύστημα πέφτει σε δίλημμα και δεν γνωρίζει τι ενέργεια να εκτελέσει για να αποφύγει την κατάσταση αυτή. Ως εναλλακτική λύση μπορεί να παρουσιαστεί ο τερματισμός ενός από των δύο προγραμμάτων. Έτσι μπορεί κανείς να συνειδητοποιήσει ότι η γνώση του τρόπου με τον οποίο μπορεί να διαχειριστεί τις καταστάσεις αυτές έχει μεγάλο αντίκτυπο τόσο στην ανάπτυξη λειτουργικών συστημάτων αλλά και στην ανάπτυξη συστημάτων επικοινωνίας αλλά και απλών προγραμμάτων (αρκεί να φανταστεί κανείς ότι στην θέση των προγραμμάτων βρίσκονται δύο εντολές και στην θέση των πηγών δύο βάσεις δεδομένων, έχοντας τις εντολές αυτές να ζητούν η μια πρόσβαση και ανάκτηση της βάσης της άλλης)

2.6.2 Αδυναμία Τερματισμού (Livelock)

Ο όρος αδυναμία τερματισμού (Livelock) αποτελεί κ*ατάσταση στην οποία ένα κρίσιμο σημείο μιας διεργασίας αδυνατεί να τερματίσει την εκτέλεσή της*. Κάτι τέτοιο μπορεί να συμβεί γιατί τα δομικά στοιχεία της συγκεκριμένης εργασίας συνεχίζουν να δημιουργούν επιπλέον ανάγκες για την ίδια, μετά από την παροχή του κρίσιμου τμήματος για την αρχική ανάγκη και πριν η συγκεκριμένη εργασία «καθαρίσει» την ουρά της (από την υπάρχουσα εργασία). Η αδυναμία

τερματισμού διαφέρει από ο αδιέξοδο (deadlock) στο σημείο ότι η διεργασία δεν μπλοκάρεται ή περιμένει κάτι να συμβεί, αλλά έχει μία άπειρη εικονική ποσότητα εργασίας να φέρει σε πέρας, αδυνατώντας να τερματίσει. Ένα παράδειγμα αδυναμίας τερματισμού αποτελεί μια διακοπή (interrupt) ενός λειτουργικού συστήματος. Εάν στον πυρήνα ενός λειτουργικού συστήματος φτάσουν πάρα πολλές αιτήσεις χωρίς η διαδικασία αυτή να σταματά, το λειτουργικό σύστημα δεν θα είναι σε θέση να εξυπηρετήσει κάποιες από τις αιτήσεις αυτές, γιατί θα ξοδέψει όλη την υπολογιστική δύναμη του επεξεργαστή στο να επεξεργαστεί τις αιτήσεις που συνεχίζουν να φτάνουν.

2.6.3 Η ιδιότητα ασφαλείας (Safety Property)

Μια ιδιότητα ασφαλείας βεβαιώνει ότι *τίποτα κακό δεν θα συμβεί κατά την εκτέλεση μια διεργασίας ενός προγράμματος ή ενός συνόλου καταστάσεων*. Ένα παράδειγμα παραβίασης της ιδιότητας θα φαίνεται από τα λανθασμένα αποτελέσματα της εκτέλεσης. Η παρακάτω LTL πρόταση αντικατοπτρίζει την συγκεκριμένη ιδιότητα:

```
[]({readySignal == 1} → (){ackSignal == 0})
```

Η πρόταση αυτή διαβάζεται ως εξής: *Πάντα, όταν το readySignal είναι ίσο με 1 τότε το ackSignal θα ισούται με 0 στον επόμενο κύκλο*. Όπως φαίνεται γίνεται χρήση του τελεστή always, [] και του τελεστή next cycle, ( ). Ένα δεύτερο παράδειγμα μιας ιδιότητας Safety είναι και το ακόλουθο :

```
[]({readySignal == 1} → (-){ackSignal == 0})
```

Πρόκειται για παράδειγμα ίδιο με το πρώτο με την μόνη διαφορά, αντί του τελεστή next cycle (), χρησιμοποιείται ο τελεστής previous cycle, (-). Επομένως η πρόταση θα διαβάζεται: *Πάντα, όταν το readySignal είναι ίσο με 1 τότε το ackSignal θα ισούται με 0 στον προηγούμενο κύκλο*.

2.6.4 Η ιδιότητα Βιωσιμότητας (Liveness Property)

Μια ιδιότητα βιωσιμότητας βεβαιώνει *ότι εντέλει κάτι καλό θα συμβεί κατά την εκτέλεση μια διεργασίας ενός προγράμματος ή ενός συνόλου καταστάσεων*. Έστω η παρακάτω LTL πρόταση:

```
<>({out1==1} && ()()[]{out2 < 2} && (-){out3==0})
```

Η πρόταση αυτή διαβάζεται ως εξής: *Τελικά, η out1 θα είναι ίση με 1 στη συνέχεια μετά από δύο κύκλους πάντα η out2 θα είναι μικρότερη του 2 και στον προηγούμενο κύκλο η out3 θα είναι ίση με το 0*. Το παράδειγμα αυτό χρησιμοποιεί τον τελεστή eventually (με σύμβολο το διαμάντι <>), τον τελεστή always [ ] και τους τελεστές next cycle ( ) και previous cycle (-).

2.6.5 Καταστάσεις Τερματισμού (End States)

Εάν μια διεργασία δεν ολοκληρώνει την εκτέλεσή τους πριν από τον τερματισμό της εκτέλεσης του μοντέλου, ο ελεγκτής μοντέλων μπορεί να σηματοδοτεί την διεργασία με την σημαία λανθασμένης κατάστασης τερματισμού (invalid end-state). Αυτή είναι μια συνηθισμένη τεχνική που χρησιμοποιείται για την ανίχνευση αδιεξόδου του μοντέλου. Εάν ο σχεδιαστής του συστήματος σχεδιάσει μια διεργασία για να σταματήσει πριν αυτή ολοκληρώσει τότε η διεργασία αυτή πρέπει να «σημαδευτεί» με την ετικέτα end.

2.6.6 Καταστάσεις προόδου (Progress States)

Διάφοροι ελεγκτές μοντέλων όπως το SPIN, χρησιμοποιεί τις καταστάσεις προόδου για την ανίχνευση παρουσίας δηλώσεων οι οποίες εκτελούνται είτε άπειρα σε σχέση με τον χρόνο είτε όχι, προσθέτοντας σε αυτές την ετικέτα progress ακριβώς για τον λόγο που θέλει το εργαλείο να δει κάθε πόσο εκτελείται μια δήλωση. Το εργαλείο θα επιστρέψει λάθος στην περίπτωση εάν δεν μπορέσει να εκτελέσει μια διεργασία *progress* για άπειρες φορές. Με άλλα λόγια, οποιαδήποτε διεργασία έχει την ετικέτα *progress* δεν μπορεί να παραμείνει μπλοκαρισμένη χωρίς αυτή να εκτελεστεί. Για παράδειγμα, έστω μιας διεργασία Α σε PROMELA :

```
proctype ProcessA(){
do
:: chanAtoB!p -> progress: chanAtoB?v
od}
```

Η παρουσία της ετικέτας *progress* απαιτεί εκτέλεση της δήλωσης *chanAtoB?y* (που θα σημαίνει ότι το κανάλι *chanAtoB* θα περιμένει να λάβει ένα μήνυμα y) άπειρες φορές. Ο μόνος τρόπος αυτή η δήλωση να εκτελεστεί άπειρες φορές είναι μόνο αν και η δήλωση *chanAtoB!p* (που θα σημαίνει ότι στο κανάλι

chanAtoB θα σταλεί από άλλη διεργασία του μοντέλου ένα μήνυμα y), όπου θα εκτελεστεί και αυτή άπειρες φορές.

2.6.7 Καταστάσεις αποδοχής (Accept States)

Μία κατάσταση αποδοχής συμπροφέρεται ακριβώς αντίθετα από μία κατάσταση προόδου. Χρησιμοποιείται για την ανίχνευση της ορθότητας προδιαγραφών των δηλώσεων σε ένα μοντέλο. Εάν ο ελεγκτής μοντέλων βρει μια κατάσταση αποδοχής να εκτελείται άπειρες φορές, τότε επιστρέφεται λάθος. Ο χρήστης μπορεί εύκολα να θέσει μια κατάσταση «παγίδα» προσθέτοντάς την ετικέτα accept και στην συνέχεια το εργαλείο θα ελέγξει για έναν άπειρο αριθμό εάν μπορεί να εισέλθει την κατάσταση «παγίδα». Εάν μπορέσει τότε η συνθήκη που οδηγεί στην παγίδα έχει γίνει αληθής και το Spin θα έχει καταφέρει να βρει λάθος. Για παράδειγμα έχοντας το παράδειγμα:

```
proctype ProcessA()
{ do
:: chanAtoB!p -> accept: chanAtoB?v
od}
```

Η διεργασία αυτή στο τέλος θα μπλοκαριστεί στην αρχή της δήλωσης *chanAtoB!p*. Εάν αυτή δεν μπλοκαριστεί τότε η διεργασία θα αδυνατούσε να τερματίσει, που σημαίνει ότι υπάρχει λάθος. Και αυτό γιατί η κατάσταση αποδοχής θα επισκέπτονταν άπειρες φορές αφού η διεργασία δεν θα μπορεί να σταματήσει κάτι που σημαίνει ότι υπάρχει λάθος.

2.6.8 Ισχυρισμοί (Never claims)

Ειδικά για τον αυτόματο ελεγκτή μοντέλων SPIN, ο αναλυτής μπορεί να χρησιμοποιεί τις δηλώσεις never-claim για να ορίσει χρονικούς τύπους ιδιοτήτων. Αυτοί οι ισχυρισμοί χρησιμοποιούνται για τον έλεγχο μη επιθυμητών ή λανθασμένων καταστάσεων με βάση χρονικούς περιορισμούς που εισάγονται από τον αναλυτή στο μοντέλο. Το SPIN για παράδειγμα, θα επιστρέψει λάθος εάν βρει κάποια ακολουθία εκτέλεσης η οποία να τερματίζει στο σημείο όπου ένας ισχυρισμός never-claim έχει τερματίσει στο τέλος μιας διεργασίας. Συνδυάζοντας τις προτάσεις ισχυρισμών με τις καταστάσεις αποδοχής που προαναφέρθηκαν το εργαλείο μπορεί να ανιχνεύσει λανθασμένους άπειρες

(κυκλικές) συμπεριφορές με το να σημαδεύει το σύνολο των δηλώσεων σαν ένα never-claim μαζί με μία ετικέτα accept δημιουργώντας έτσι μία κατάσταση αποδοχής. Στη συνέχεια το SPIN ελέγχει τις επιλεγμένες δηλώσεις για άπειρες φορές εάν μπορεί να εισέλθει μέσα στην μπλοκαρισμένη κατάσταση αποδοχής.

2.6.9 Επιβεβαιώσεις (Assertions)

Η πρόταση σε PROMELA : `assert(destid==a)`, αποτελεί μια επιβεβαίωση. Μία επιβεβαίωση είναι μία Boolean συνθήκη η οποία πρέπει να ικανοποιείται όταν μια διεργασία φτάσει σε μια συγκεκριμένη κατάσταση. Εάν η συνθήκη είναι αληθής τότε η επιβεβαίωση δεν έχει καμία επίδραση. Αντίθετα, η ορθότητα της δήλωσης παραβιάζεται εάν υπάρχει τουλάχιστον μία ακολουθία εκτέλεσης στην οποία η συνθήκη είναι λάθος την στιγμή που η επιβεβαίωση γίνεται εκτελέσιμη.

2.6.10 Κύκλοι Στασιμότητας (Non-progress Cycles)

Με σκοπό να μπορεί κάποιος να ισχυριστεί την απουσία ενός non-progress cycle, θα πρέπει να βρεθεί ένας τρόπος να οριστούν καταστάσεις η οποίες αποδεικνύουν την πρόοδο της εκτέλεσης ενός μοντέλου συστήματος. Μια ετικέτα κατάστασης προόδου σημειώνει ότι μια κατάσταση πρέπει να εκτελεστεί για να δηλώσει την πρόοδο της εκτέλεσης του μοντέλου. Οι ακολουθίες εκτέλεσης οι οποίες παραβιάζουν την συγκεκριμένη ετικέτα ονομάζονται καταστάσεις κύκλων στασιμότητας.

2.6.11 Χρονικοί ισχυρισμοί (Temporal Claims)

Οι χρονικοί ισχυρισμοί αποτελούν δηλώσεις ισχυρισμού οι οποίες ορίζουν την χρονική διάταξη των ιδιοτήτων για καταστάσεις του μοντέλου. Αλλά από την στιγμή που όλα τα κριτήρια ορθότητας στην γλώσσα προδιαγραφών βασίζονται πάνω σε ιδιότητες που ισχυριζόμαστε ότι είναι αδύνατες, οι χρονικοί ισχυρισμοί πρέπει επίσης να εκφράζουν ιδιότητες οι οποίες είναι αδύνατες. Για παράδειγμα ένας τέτοιος ισχυρισμός ακολουθεί παρακάτω σε PROMELA:

```
never
{ do
```

```
   :: skip
   :: condition -> break
  od;
accept: do
         :: condition
        od;}
```

2.7 Συμπεράσματα Κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκαν και περιγράφθηκαν οι τυπικές μέθοδοι ανάλυσης συστημάτων και οι βασικές αρχές που διέπουν αυτές. Δόθηκε έμφαση στην χρησιμότητα των τεχνικών αυτών, και ειδικότερα του αυτόματου ελέγχου μοντέλων που ασχολείται αυτή η διατριβή. Παρουσιάστηκε το γνωστό φαινόμενο της έκρηξης του χώρου των καταστάσεων που συναντάται συχνά κατά την μοντελοποίηση συστημάτων και ειδικά των πρωτοκόλλων ασφαλείας, αναλύοντας τις τεχνικές που έχουν προταθεί σήμερα για την αντιμετώπισή του. Περιγράφονται τα δύο εργαλεία στα οποία βασίζονται τα επόμενα κεφάλαιο αυτής της διατριβής. Ο αυτόματος ελεγκτής μοντέλων SPIN, και η γλώσσα προδιαγραφών του PROMELA, προσπαθώντας με αυτό τον τρόπο να κατανοήσει ο αναγνώστης την όλη ερευνητική περιοχή των τυπικών μεθόδων ανάλυσης μέσω του συγκεκριμένου εργαλείου. Στο ίδιο μήκος, παρουσιάζεται και ο πιθανοκρατικός ελεγκτής μοντέλων PRISM αναφέροντας πλεονεκτήματα και μειονεκτήματα χρήσης του κατά την επαλήθευση συστημάτων και ιδιαίτερα πρωτοκόλλων ασφαλείας. Τέλος, περιγράφονται προσεγγίσεις δημιουργίας μοντέλων εισβολέων με την χρήση των τυπικών μεθόδων για την αποτελεσματικότερη ανάλυση των πρωτοκόλλων ασφαλείας, περιγράφοντας τα χαρακτηριστικά αυτών, όπως παρουσιάζονται στην βιβλιογραφία σήμερα.

Κεφάλαιο 3ο

Πρωτόκολλα Ασφαλείας και Τυπικές Μέθοδοι

3.1 Εισαγωγή

Σήμερα, οι τυπικές μέθοδοι ανάλυσης συστημάτων τείνουν να αποτελέσουν ένα αναπόσπαστο κομμάτι της φάση δημιουργίας λογισμικού, ειδικότερα για συστήματα κρίσιμα στην ασφάλεια. Το μεγαλύτερο πλεονέκτημα των μεθόδων αυτών είναι τα επιτυχή τους αποτελέσματα στην έγκαιρη διάγνωση λαθών, στον σχεδιασμό λογισμικού, μετά από τον συνδυασμό της τυπικής ανάλυσης και της τυπικής (εξαντλητικής) επαλήθευσης του συστήματος. Έτσι, με βάση το [18], οι τυπικές μέθοδοι ανάλυσης συστημάτων και ειδικότερα, πρωτοκόλλων ασφαλείας μπορούν να:

- Οριοθετήσουν την επίδραση κατά την εκτέλεση ενός πρωτοκόλλου όπως για παράδειγμα για το αν επηρεάζει μια εκτέλεση διεπαφής χρήστη διαδικτυακού πρωτοκόλλου επικοινωνίας το περιβάλλον της
- Περιγράφουν την συμπεριφορά του πρωτοκόλλου με ακρίβεια
- Ορίζουν τις επιθυμητές ιδιότητες του πρωτοκόλλου ασφαλείας που πρέπει να ισχύουν

- Αποδεικνύουν εάν ένα πρωτόκολλο πληρεί τις ιδιότητές του. Εάν όχι, οι περισσότερες από αυτές τις μεθόδους επιστρέφουν ένα παράδειγμα-ίχνους για το σημείο εκείνο που βρέθηκε το λάθος
- Ωθούν τον αναλυτή-σχεδιαστή του πρωτοκόλλου ασφαλείας, να αντιληφθεί την εργασία υλοποίησης του συστήματος με έναν ακριβή τρόπο.

Με βάση τα παραπάνω, συμπεραίνεται ότι κάθε αναλυτής/σχεδιαστής οποιουδήποτε συστήματος πρέπει να λαμβάνει υπ' όψιν του, ότι για την αποτελεσματική λειτουργία αυτών, δεν επαρκεί απλά ο σχεδιασμός και η υλοποίησή τους. Αντίθετα με την καταγραφή των προσδοκώμενων λειτουργιών του συστήματος, διακρίνονται οι ιδιότητές του που πρέπει ανά πάσα χρονική στιγμή να εκπληρούνται, διατηρώντας παράλληλα μια υψηλού επιπέδου προσφερόμενη ποιότητα και ασφάλεια, για τους τελικούς χρήστες.

Από τη στιγμή όπου και τα διαδικτυακά πρωτόκολλα επικοινωνιών αποτελούν υλοποιήσεις *συστημάτων κατανεμημένου λογισμικού* [17], ο επιπρόσθετος έλεγχος για την ποιότητα και την ασφάλεια των υπηρεσιών τους, τίθεται ως το κύριο χαρακτηριστικό τους. Σήμερα, οι τυπικές μέθοδοι ανάλυσης συστημάτων υιοθετούνται από τις μεγαλύτερες εταιρείες παραγωγής κρίσιμου σε ασφάλεια λογισμικού, με σκοπό την επαλήθευση των εγγυήσεων που παρέχουν τα προϊόντα τους. Στις επόμενες παραγράφους του κεφαλαίου αυτού, θα δοθούν περιγραφές σε σχέση με εφαρμογή των τυπικών μεθόδων σε πρωτόκολλα ασφαλείας, των ιδιοτήτων ασφαλείας που προσφέρουν και ελέγχονται, αλλά και ο συνδυασμός του ελέγχου αυτού με εξειδικευμένα μοντέλα εισβολέων, για την πληρέστερη ανάλυση των πρωτοκόλλων.

3.2 Τυπική ανάλυση πρωτοκόλλων ασφαλείας

Η σχεδίαση ασφαλών (κρυπτογραφικών) πρωτοκόλλων αποτελεί μια πολύπλοκη και δύσκολη διεργασία για τους αναλυτές. Μέχρι πρότινος, οι ερευνητές κατέφευγαν σε τυχαίες προσομοιώσεις των συστημάτων που σχεδίαζαν, με σκοπό την αποτύπωση τυχόν λαθών ή δυσλειτουργιών από την εκτέλεση των συστημάτων τους. Οι τυπικές μέθοδοι ανάλυσης, τόσο οι αυτόματες όσο και οι τεχνικές θεωρίας-απόδειξης, συνεισφέρουν σε ένα πολύ

χρήσιμο εργαλείο για την απόδειξη ασφάλειας και λειτουργικότητας των συστημάτων και ιδιαίτερα των πρωτοκόλλων που αναπτύσσονται. Τα περισσότερα πρωτόκολλα ασφαλείας που έχουν προταθεί σήμερα, θεωρούνται πολύπλοκα, ανάλογα με το πλήθος των διαδικαστικών εντολών που απαιτούνται για να ολοκληρωθούν επιτυχώς αλλά και των μηχανισμών ασφαλείας που εμπεριέχουν προς διατήρηση αρχών ασφαλείας. Σχετική αναζήτηση στην βιβλιογραφία σήμερα, αποδεικνύει την ύπαρξη αρκετών περιπτώσεων όπου πολλά από τα προτεινόμενα πρωτόκολλα ασφαλείας [45] έχουν βρεθεί με σημαντικά λάθη, όπου παραβιάζεται η ασφάλεια των συμμετεχόντων σε αυτά.

Τα διάδοση και η χρήση των πρωτοκόλλων ασφαλείας εξελίσσεται ραγδαία στο ηλεκτρονικό εμπόριο και στις ειδικότερα στις ηλεκτρονικές τραπεζικές συναλλαγές, όπου η ασφάλεια κρίνεται ως το μείζον χαρακτηριστικό τους. Πρόκειται για συστήματα όπου τα επιμέρους κομμάτια (υλοποιήσεις τους) κρίνονται ως κρίσιμα σε ασφάλεια κατανεμημένοι μηχανισμοί, εύκολοι στην περιγραφή τους, αλλά δύσκολοι να αναλυθούν χειρωνακτικά από τους αναλυτές. Για παράδειγμα, για τον σχεδιασμό και την επαλήθευση ενός πρωτοκόλλου ασφαλείας όπως αυτό των Needham και Schroeder [74], ο αναλυτής θα πρέπει να κατανοήσει την βασική κρυπτογραφία που χρησιμοποιείται, μιας και οι μηχανισμοί κρυπτογράφησης και αυθεντικοποίησης που χρησιμοποιούνται θεωρούνται αναπόσπαστο κομμάτι για κάθε πρωτόκολλο ασφαλείας που εξετάζεται.

Με βάση το [45], κάθε πρωτόκολλο ασφαλείας θα πρέπει να πληρεί συγκεκριμένες προϋποθέσεις-ιδιότητες ασφαλείας με το σε όλη την διάρκεια της λειτουργίας του. Οι ιδιότητες αυτές, όπως περιγράφονται στο [37] συνοψίζονται στις παρακάτω:

- Ακεραιότητα πληροφοριών (integrity): Είναι η ιδιότητα των δεδομένων να υφίστανται σε προκαθορισμένο φυσικό μέσο ή χώρο και να είναι ακριβή. Δηλαδή η μη-εξουσιοδοτημένη τροποποίηση της πληροφορίας θα πρέπει να αποτρέπεται, ενώ κάθε αλλαγή του περιεχομένου των δεδομένων να είναι αποτέλεσμα εξουσιοδοτημένης και ελεγχόμενης ενέργειας.

- Εμπιστευτικότητα πληροφοριών (confidentiality): Η ιδιότητα των δεδομένων να καθίστανται αναγνώσιμα μόνο από εξουσιοδοτημένα λογικά υποκείμενα, όπως φυσικές οντότητες και διεργασίες λογισμικού.
- Διαθεσιμότητα πληροφοριών (availability): Η αποτροπή της προσωρινής ή μόνιμης άρνησης διάθεσης της πληροφορίας σε κάθε εξουσιοδοτημένο λογικό υποκείμενο του συστήματος.
- Αυθεντικοποίηση των οντοτήτων (authenticity): Η επαλήθευση πιστοποίησης της ταυτότητας των οντοτήτων που συμμετέχουν στο πρωτόκολλο.

Ένα πρωτόκολλο ασφαλείας σχεδιάζεται πρώτιστος για να μπορέσει να παρέχει μία ή περισσότερες από τις παραπάνω ιδιότητες ασφαλείας. Οι συγκεκριμένοι μηχανισμοί που εμπεριέχονται στο πρωτόκολλο, ποικίλουν ανάλογα με το περιβάλλον το οποίο προορίζεται να λειτουργήσει το πρωτόκολλο, όπως για παράδειγμα η υπολογιστική ισχύ που πρέπει να κατανάλωση η κάθε οντότητα για την σωστή ολοκλήρωση της επικοινωνίας. Έτσι, ο κάθε σχεδιαστής επιλέγει κρυπτογραφικούς μηχανισμούς όπως ψηφιακές υπογραφές, συμμετρική ή ασύμμετρη κρυπτογράφηση ή συναρτήσεις κατακερματισμού [32][20][37][36][80].

Η ανίχνευση και η αποφυγή λαθών σε ένα πρωτόκολλο ασφαλείας αποτελούν τους δύο βασικούς παράγοντες χρήσης των τυπικών μεθόδων για την επαλήθευση αυτών. Η προδιαγραφές ενός συστήματος-πρωτοκόλλου ορίζονται με ειδικές γλώσσες προδιαγραφών που παρέχει το εκάστοτε εργαλείο [29] για τον επακριβή καθορισμό των επιθυμητών ή ανεπιθύμητων καταστάσεων ενός συστήματος. Για παράδειγμα, ο αναλυτής μπορεί να ελέγξει ιδιότητες εγγύησης ασφάλειας, όπως ότι κάτι κακό ποτέ δεν πρόκειται να συμβεί αλλά και ιδιότητες ορθής λειτουργικότητας, που αναφέρονται στην βιβλιογραφία ως ιδιότητες βιωσιμότητας (liveness properties), που σημαίνουν ότι κάτι καλό-σωστό συμβαίνει απείρως συχνά.

Το ίδιο το πρωτόκολλο ασφαλείας που θέλουμε να εξετάσουμε, περιγράφεται με την γλώσσα προδιαγραφών που δέχεται το εκάστοτε εργαλείο που έχουμε επιλέξει για την ανάλυση. Το πρωτόκολλο ασφαλείας περιγράφεται συνήθως ως ένα σύνολο διεργασιών που αλληλεπιδρούν μεταξύ τους προς την ολοκλήρωση των στόχων τους. Οι διεργασίες αυτές αντιπροσωπεύουν συνήθως

τις συμμετέχουσες οντότητες του πρωτοκόλλου, οι οποίες και στην όλη φάση λειτουργίας τους, είναι σχεδιασμένες να εκτελούν εντολές σε ένα εχθρικό περιβάλλον, το οποίο αντιπροσωπεύεται από μια άλλη οντότητα-διεργασία που αποτελεί τον εισβολέα του πρωτοκόλλου.

Στο χειρότερο σενάριο της ανάλυσης, ο εισβολέας τίθεται ως ο απόλυτος κυρίαρχος του επικοινωνιακού όπου μπορεί να υποκλέψει όλα τα μηνύματα του πρωτοκόλλου ή ακόμα και να αποτρέψει το πρωτόκολλο να τερματίσει σωστά. Η ανταλλαγή των μηνυμάτων μεταξύ των οντοτήτων του πρωτοκόλλου μπορεί να αναλυθεί εφαρμόζοντας στην όλη επικοινωνία διαφορετικές τεχνικές παραπλάνησης των οντοτήτων από τον εισβολέα. Για παράδειγμα, η ασφάλεια ενός πρωτοκόλλου μπορεί να μελετηθεί από τον έλεγχο μοντέλων ενός πρωτοκόλλου ασφαλείας, με την σύγκριση των χώρων των καταστάσεων που παράγονται, με παρουσία ενός μοντέλου εισβολέα και με την απουσία του. Με την προϋπόθεση ότι ένας εισβολέας κατέχει πιθανές ενέργειες εκμετάλλευσης των μηνυμάτων ενός πρωτοκόλλου, για την εξόρυξη απόρρητων ή όχι πληροφοριών που έχουν να κάνουν με τις οντότητες του πρωτοκόλλου. Οι διαφορές στις δύο αναλύσεις του παραγόμενου χώρου των καταστάσεων μπορούν να δώσουν πληροφορίες στον αναλυτή για την αποτελεσματικότητα ή όχι επιθέσεων που μπορεί να εξαπολύσει ο εισβολέας εναντίον των συμμετεχόντων [31][35].

Περνώντας στους κρυπτογραφικούς μηχανισμούς που χρησιμοποιούν τα πρωτόκολλα ασφαλείας, ή ανάλυση τους με τις τυπικές μεθόδους και ειδικά με τον έλεγχο μοντέλων αποτελούν έναν σημαντικό παράγοντα από τον οποίο εξαρτάται το μέγεθος τόσο του χώρου των καταστάσεων, όσο και η ευκολία διεκπεραίωσης της όλης ανάλυσης. Η πλειοψηφία των μηχανισμών αυτών συνηθίζεται να αναπαρίστανται από συγκεκριμένους κανόνες (rewriting rules) [1], οι οποίοι είναι σύμφωνοι με διαδικασίες όπως (συμμετρική ή ασύμμετρη) κρυπτογράφηση, αποκρυπτογράφηση και επαλήθευση ενός μηνύματος. Ο εισβολέας γνωρίσει την λειτουργία των κανόνων αυτών, αλλά μπορεί να τους φέρει ης πέρας μόνο αν για παράδειγμα γνωρίζει το σωστό κλειδί της αποκρυπτογράφησης ενός κρυπτογραφημένου μηνύματος. Τέτοιοι μηχανισμοί, όπως θα δούμε και σε επόμενα κεφάλαια περιλαμβάνουν:

- *Συμμετρική κρυπτογραφία*

- *Ασύμμετρη κρυπτογραφία*
- *Συναρτήσεις κατακερματισμού*
- *Ψηφιακές Υπογραφές*
- *Μηχανισμοί αυθεντικοποίησης μηνυμάτων*

Η ξεχωριστή αναπαράσταση των μηχανισμών αυτών, έγκειται στην σωστή αφαιρετική ικανότητα του κάθε αναλυτή να περιγράψει με ακρίβεια την λειτουργία τους, χωρίς αυτό να σημαίνει την απευθείας υλοποίησή τους στο περιβάλλον ενός αυτόματου ελεγκτή μοντέλων. Έχοντας υπ' όψιν ότι αρκετοί από τους μηχανισμούς αυτούς βασίζονται σε παραγωγή και χειραγώγηση τυχαίων αριθμών (Nonces), το εύρος του πεδίου τιμών των αριθμών αυτών, σε περίπτωση που υλοποιηθεί με μια τυπική μέθοδο ανάλυσης, θα καθιστά την ολοκλήρωσή της αδύνατη. Κάτι τέτοιο θα οφείλεται στην έκρηξη του χώρου των καταστάσεων που μπορούν να παραχθούν από έναν τέτοιο μηχανισμό μιας και η απλή απαρίθμηση όλων των δυνατών καταστάσεων (και όλων των διακριτών περιπτώσεων με διαφορετική τιμή τυχαίου αριθμού), οδηγεί την ανάλυση σε αποτυχία.

3.3 Εχθρική τυπική ανάλυση πρωτοκόλλων ασφαλείας

Ένα από τα μεγαλύτερα κίνητρα της τυπικής ανάλυσης των πρωτοκόλλων ασφαλείας βασίζεται στην λειτουργία ενός αντιπροσωπευτικού μοντέλου, όσον αφορά το υπό-εξέταση πρωτόκολλο, μαζί με μία οντότητα-εισβολέα η οποία και προσπαθεί να προσομοιώσει εχθρικές συμπεριφορές που τυγχάνουν σήμερα στα επικοινωνιακά συστήματα ασφαλείας. Σήμερα αρκετοί σχεδιαστές πρωτοκόλλων ασφαλείας τείνουν όλο και περισσότερο της χρήσης οντοτήτων εισβολέων με σκοπό της, από κάθε πλευράς, εξαντλητικής ανάλυσης και αναζήτησης λαθών στο τελικό προϊόν τους. Η ανακάλυψη αυτού τους είδους των λαθών ασφαλείας που μπορεί να υποκύψουν τα σημερινά πρωτόκολλα εξαρτάται τις περισσότερες φορές από την επιτυχή μοντελοποίηση επιθέσεων που μπορεί ένας επιτιθέμενος (εισβολέας) να εξαπολύσει. Σε τέτοιες περιπτώσεις όμως, η ανάλυση και επαλήθευση των συγκεκριμένων συστημάτων είναι πιθανόν να οδηγήσει σε περιπτώσεις που είναι γνωστές ως έκρηξη του χώρου καταστάσεων (State Space Explosion) [95] [97]. Ιδανικό σενάριο σε

αυτήν την περίπτωση θα μπορούσε να αποτελέσει η δημιουργία μοντέλων εισβολέων που, χωρίς να μειονεκτούν σε σχέση με νομότυπες οντότητες πρωτοκόλλων, θα έχουν την δυνατότητα παρεμβολής και ίσως ανακάλυψης μιας κατάστασης στο συγκεκριμένο πρωτόκολλο που μπορεί να αποτελέσει λάθος σχεδιασμού του. Ακολουθώντας την προτεινόμενη μεθοδολογία κατασκευής των εν λόγω εισβολέων και ταυτόχρονα με την χρήση εργαλείων αυτόματου ελέγχου μοντέλων, δίνεται η δυνατότητα να βρεθούν τέτοιες μη-ασφαλής καταστάσεις σε ένα πρωτόκολλο, που συνηγορούν την εύρεση επίθεσης σε αυτό.

Η πιο συνήθης τεχνική κατά την διάρκεια της μοντελοποίησης ενός πρωτοκόλλου ασφαλείας είναι η δημιουργία δύο οντοτήτων, συχνά αποκαλούμενοι στην βιβλιογραφία ως Bob και Alice. Οι δύο αυτές οντότητες θα είναι υπεύθυνες για την μοντελοποίηση του πρωτοκόλλου, ακολουθώντας με ακρίβεια τα απαιτούμενα βήματα για την επιτυχή ολοκλήρωσή του. Παρόλα αυτά θα πρέπει να σημειώσουμε την αφαιρετικότητα που πρέπει να εισαχθεί και να υιοθετηθεί καθ' όλη την διάρκεια μοντελοποίησης του υπό-εξέταση πρωτοκόλλου. Σε τέτοιες περιπτώσεις μηχανισμοί πρωτοκόλλου όπως ψηφιακές υπογραφές, μηχανισμοί κρυπτογράφησης και συναρτήσεις κατακερματισμού, δεν υλοποιούνται αλλά αναπαρίστανται συμβολικά προς αποφυγήν του φαινομένου της έκρηξης του χώρου των καταστάσεων.

Κατά την διάρκεια μοντελοποίησης εισβολέων ανάμεσα σε πρωτόκολλα ασφαλείας, ο σχεδιασμός των μοντέλων εισβολέων πρέπει να διέπεται από συγκεκριμένες υποθέσεις υλοποίησης. Τέτοιες υποθέσεις περιλαμβάνουν συνθήκες όπως οι ακόλουθες: α) Η χρησιμοποιούμενη μέθοδος κρυπτογράφησης θεωρείται απαραβίαστη β) Ο εισβολέας μπορεί να αποτρέψει οποιοδήποτε μήνυμα από την άφιξή του στον αναμενόμενο παραλήπτη του και γ) Ο εισβολέας μπορεί να δημιουργήσει μηνύματα από μόνος του. Ως αποτέλεσμα αυτών των υποθέσεων, η ανάλυση του ελέγχου μοντέλων συμπεριφέρεται σε κάθε μήνυμα προερχόμενο από μία νόμιμη οντότητα με παραλήπτη τον εισβολέα, και κάθε μήνυμα με προορισμό μια έντιμη οντότητα ότι προέρχεται από τον εισβολέα. Ουσιαστικά υποτίθεται ότι σύστημα αποτελεί μια μηχανή η οποία και βρίσκεται υπό τον πλήρη έλεγχο του εισβολέα με σκοπό την δημιουργία σειράς ενεργειών-επιθέσεων. Η συμπεριφορά του εισβολέα ορίζεται ως «ένας κανόνας-βάσης κυριαρχίας της σύνθεσης, από-σύνθεσης,

κρυπτογράφησης-αποκρυπτογράφησης (με γνωστά κλειδιά) και μνήμης χρησιμοποίησης των καταγεγραμμένων μηνυμάτων. Στην επόμενη παράγραφο θα παρουσιαστεί μια σύνοψη των εργασιών με το μεγαλύτερο αντίκτυπο όσον αφορά την δημιουργία εισβολέων με την βοήθεια του ελέγχου μοντέλων. Οι περισσότερες από αυτές βασίζονται στο γενικό μοντέλο εισβολέα Dolev-Yao [34] το οποίο και παρουσιάζει αρκετές αδυναμίες ως προς τον περιορισμό του στο να ανακαλύψει ένα μήνυμα το οποίο και προορίζονταν να παραμείνει μυστικό ή στο να παράγει μηνύματα με σκοπό την εχθρική αντιπροσώπευση ενός έντιμου συμμετέχοντα. Οποιαδήποτε αποτυχία στην μυστικότητα και στην αυθεντικοποίηση μηνυμάτων και οντοτήτων αντίστοιχα, συνηγορούν σε μία άγνωστη επίθεση του υπό εξέταση πρωτοκόλλου ασφαλείας.

3.4 Μοντέλα εισβολέων

Ο έλεγχος μοντέλων των πρωτοκόλλων ασφαλείας έχει πρόσφατα συνδυαστεί με την με την ανάπτυξη εξειδικευμένων μοντέλων εισβολέων, με σκοπό την ανακάλυψη λαθών όπως παραβίασης ιδιοτήτων ιδιωτικότητας ή αυθεντικοποίησης των οντοτήτων του πρωτοκόλλου. Στους υπάρχοντες ελεγκτές μοντέλων, ο εισβολέας συνήθως ορίζεται σαν ένας κανόνας άντλησης μηνυμάτων για την χειραγώγηση πληροφορίας η οποία και θα βοηθήσει τον εισβολέα να κάνει γνωστό ένα μήνυμα (ή μέρος αυτού) που προορίζεται να είναι κρυφό (σε ένα πρωτόκολλο) ή την παραγωγή μηνυμάτων για την μίμηση (impersonation) μιας από τις οντότητες του πρωτοκόλλου.

Μια από τις πρώτες θεωρίες εισβολέων που χρησιμοποιήθηκαν εκτενώς στις σχετικές έρευνες, βασιζόμενη στο μοντέλο εισβολέα των Dolev και Yao, είναι και αυτή που χρησιμοποιεί το εργαλείο *Ανακριτής* (Interrogator) [71]. Δεδομένης μιας τελικής κατάστασης ενός συστήματος στην οποία ο εισβολέας γνωρίζει μια λέξη (μήνυμα), η οποία θα έπρεπε να είναι μυστική καθ' όλη την διάρκεια της συνόδου του πρωτοκόλλου, ο Interrogator δοκιμάζει όλους τους τρόπους με τους οποίους θα μπορέσει να δημιουργήσει μονοπάτια, όπου η προαναφερθείσα κατάσταση (λάθους) του συστήματος μπορεί να προσεγγιστεί. Εάν βρεθεί ένα τέτοιο μονοπάτι, τότε έχει ανακαλυφθεί ένα λάθος ασφαλείας στο πρωτόκολλο. Η πεπερασμένη ανάλυση των καταστάσεων για τα

κρυπτογραφικά πρωτόκολλα, έχει αναπτυχθεί μέσα από τη σχετική βιβλιογραφία με ραγδαίο τρόπο. Μερικά από τα εργαλεία τα οποία δοκιμάζουν την εφαρμογή της τεχνικής αυτής, είναι το εργαλείο BRUTUS [28], το Murφ [72] και το FDR (Failures Divergence Refinement) [82].

Παρόλα αυτά, οι εγγυήσεις ασφαλείας που πρέπει να προσφέρουν τα σημερινά πρωτόκολλα ασφαλείας, δεν μπορούν να εκφράζονται ως ιδιότητες απουσίας παράβασης της μυστικότητας ή της αυθεντικοποίησης. Μια τυπική περίπτωση του παραπάνω αποτελεί και το γνωστό σύνολο επιθέσεων επανάληψης (replay attacks), όπου ο εισβολέας έχει ως στόχο την επαναληπτική αποστολή προηγούμενων καταγεγραμμένων μηνυμάτων, σε μια προσπάθεια να σαμποτάρει την τρέχουσα σύνοδο του πρωτοκόλλου στο οποίο παρεμβάλλεται. Στο [99] οι συγγραφείς περιγράφουν ότι οι αποτυχίες ανταλλαγής πληροφορίας μεταξύ οντοτήτων ενός πρωτοκόλλου σε πεπερασμένα πλαίσια χρόνου, μπορεί να είναι ευάλωτο σε επιθέσεις επαναληπτικής αποστολής μηνύματος οι οποίες δεν συγκαταλέγονται στην κατηγορίας παραβιάσεων ιδιοτήτων μυστικότητας ή αυθεντικοποίησης. Τέτοιες επιθέσεις αναλύθηκαν εκτενέστερα στο [87] με εξειδικευμένες modal logics, όπως η λογική BAN (που πήρε το όνομά της από τους εφευρέτες της Burrows, Abadi και Needham [20]). Μια λογική η οποία μετέπειτα αποδείχθηκε ότι περιείχε λάθη, τα οποία περιγράφθηκαν στο [75]. Άλλες μελέτες, όπως η [67]συμπεραίνουν ότι η ιδιότητα της αυθεντικοποίησης αποτελεί μια εξαρτώμενη έννοια μιας και δεν υπάρχει ένας ενιαίος ορισμός της, όπου όλα τα πρωτόκολλα ασφαλείας πρέπει να ικανοποιούν.

Η πιο λεπτομερής περιγραφή του εισβολέα Dolev-Yao δίνεται στην έκδοση ενός εισβολέα με την ονομασία *Τεμπέλης Κατάσκοπος* (Lazy Spy). Ο εισβολέας αυτός αρχικά περιγράφθηκε με βάση τα ίχνη ενός μοντέλου βασισμένο σε CSP (Communicating Sequential Processes) στο εργαλείο FDR, ο οποίος αργότερα ενσωματώθηκε στο εργαλείο Casper [61]. Το Casper αποτελεί ένα περιβάλλον για αυτόματο έλεγχο μοντέλων για ημι-αυτοματοποιημένες περιγραφές μοντέλων σε CSP που αφορούν πρωτόκολλα ασφαλείας. Η λειτουργία του Casper βασίζεται σε ένα παραμετροποιημένο σύνολο κανόνων που αντλούν πληροφορίες από τα ανταλλασσόμενα μηνύματα μέσω της κρυπτογράφησής τους, χρησιμοποιώντας μια *τεμπέλικη* στρατηγική αναζήτησης για λάθη, η οποία

και εξετάζει ένα μέρος των συνολικών καταστάσεων του εισβολέα που είναι προσεγγίσιμες από καταστάσεις που θα βρεθεί το υπό εξέταση πρωτόκολλο.

Το εργαλείο NRL (NRL Protocol Analyzer) [69] αποτελεί ένα άλλο γνωστό εργαλείο το οποίο βασίζεται στην χρήση του εισβολέα Dolev-Yao. Όπως και στην περίπτωση του *Ανακριτή*, ο αναλυτής ορίζει μια μη-ασφαλή κατάσταση (λάθους) και το εργαλείο προσπαθεί εξαντλητικά να βρει ένα μονοπάτι από την αρχική κατάσταση προς σε αυτή. Ένα αρκετά ενδιαφέρον χαρακτηριστικό στο εργαλείο NRL, είναι ότι επιτρέπει τον παραπάνω έλεγχο για έναν απεριόριστο αριθμό συνόδων (session) ενός πρωτοκόλλου, σε ένα μονοπάτι.

Στο [38] οι συγγραφείς παρουσιάζουν μια αναλυτική επισκόπηση των πιο σημαντικών αναλύσεων του χώρου των καταστάσεων κατά την διάρκεια του ελέγχου μοντέλων, που έχουν γίνει από το 1999. Ένας άλλος εισβολέας που βασίζεται εν μέρει στον εισβολέα DY, αλλά και στον *Τεμπέλη Κατάσκοπο* που περιγράφθηκε παραπάνω, υλοποιήθηκε στον on-the-fly ελεγκτή μοντέλων της εργαλειοθήκης ασφαλείας AVISPA, που περιγράφεται στο [16]. Ο εισβολέας αυτός αποφεύγει την απευθείας καταμέτρηση και καταγραφή όλων των δυνατών μηνυμάτων που μπορεί να παράγει, με την αποθήκευση και χειραγώγηση περιορισμών, οι οποίοι θέτουν επακριβώς τα όρια για το ποια – χρήσιμη και μόνο- πληροφορία πρέπει να παραχθεί. Τα αποτελέσματα αυτής της συμβολικής αναπαράστασης επαληθεύονται με έναν καθοδηγούμενη κατ' απαίτηση τρόπο, μειώνοντας το συνολικό δένδρο της αναζήτησης, χωρίς να αποκλείονται επιθέσεις από τον εισβολέα.

Στο [23] οι συγγραφείς παρουσιάζουν ένα *process algebraic* μοντέλο εισβολέα για την επαλήθευση ιδιοτήτων βιωσιμότητας σε πρωτόκολλα ασφαλείας. Γι αυτή την τάξη των ιδιοτήτων, ο προτεινόμενος εισβολέας αποδεικνύεται ότι είναι ισοδύναμος με τον εισβολέα των Dolev-Yao, αφού δεν καθυστερεί επ' αόριστον την αποστολή των μηνυμάτων. Το μοντέλο του εισβολέα περιορίζεται από έναν απλό περιορισμό δικαιοσύνης (fairness) με την προσθήκη του χαρακτηριστικού που ο εισβολέας δεν σταματά την ροή των μηνυμάτων προς τον έντιμη οντότητα-προορισμό τους. Με αυτό τον τρόπο ο εισβολέας είναι ικανός στο να βρίσκει επιθέσεις (παράγοντας και ένα counterexample) για την υπό εξέταση ιδιότητα βιωσιμότητας.

Στην σχετική βιβλιογραφία, παρουσιάζονται σημαντικές ερευνητικές προσπάθειες για βελτιώσεις ή επεκτάσεις εισβολέων ανάλογα με την περίπτωση επιθέσεων και γενικότερα ιδιοτήτων ασφαλείας που στοχεύει να ελέγξει η όλη ανάλυση. Οι περισσότεροι από αυτούς τους εισβολείς βασίζονται στο γενικό μοντέλο εισβολέα των Dolev και Yao [6][10][23][25][26][53], προσπαθώντας να παρέχουν μια ενοποίηση ενός ειδικού-ιδιοτήτων εισβολέα με την πλειοψηφία των τεχνικών αναζήτησης και μείωσης του παραγόμενου χώρου καταστάσεων.

Εστιάζοντας στο πρόβλημα της μείωσης του χώρου των καταστάσεων για την αποφυγή του φαινομένου ΕΧΚ, μια σειρά από σημαντικές εργασίες έχουν δημοσιευθεί μέχρι σήμερα, ειδικότερα με τον συνδυασμό τους με την τεχνική της μερικής μείωσης (P.O.R), [6][7][21][22][88]. Στο [53] οι συγγραφείς προτείνουνε τον αυτόματο έλεγχο μοντέλων πρωτοκόλλων ασφαλείας με προκαταρκτικές παραμετροποιήσεις που έχουν να κάνουν τόσο με το εργαλείο που χρησιμοποιείται, όσο και με τις ιδιότητες που επιθυμεί ο αναλυτής να ελέγξει.

Στο [87] οι συγγραφείς αποδεικνύουν την εγκυρότητα δύο βελτιώσεων του μοντέλου του εισβολέα DY. Η πρώτη βελτίωση αφορά μια τεχνική η οποία θέτει τον εισβολέα σε θέση να υποκλέπτει όλα τα μηνύματα του πρωτοκόλλου ασφαλείας. Στην δεύτερη βελτίωση, ο εισβολέας έχει την ικανότητα να μην επιτρέπει την αποστολή μηνυμάτων σε καταστάσεις όπου ο παραλήπτης αυτών είναι έτοιμος να στείλει ένα άλλο μήνυμα. Μια τέτοια τεχνική για τον εισβολέα μπορεί να μειώσει τον χώρο των καταστάσεων, που όπως δείχνεται και στην συγκεκριμένη ερευνητική εργασία μπορεί για πολύπλοκα επικοινωνιακά συστήματα πρωτοκόλλων να φτάσει και μείωση σε 43%. Ένας άλλος εισβολέας που επίσης βασίζεται επίσης στο μοντέλο του DY προτείνεται στο [26], για την ανάλυση μη πεπερασμένων αριθμού συνόδων ενός πρωτοκόλλου ασφαλείας ή ενός πρωτοκόλλου με μη πεπερασμένο αριθμό μηνυμάτων.

Η τεχνική της συμβολικής μείωσης [65] του χώρου των καταστάσεων, αποτελεί μια από τις ερευνητικές προσπάθειες που γίνονται στον χώρο του ελέγχου μοντέλων. Με βάση αυτή, ο παραγόμενος χώρος των καταστάσεων αντιλαμβάνεται ως ένα συμβολικά μεταφραζόμενο πεδίο τιμών , όπου με βάση τα [29]και [83] μπορεί ο συμμετρικός πλεονασμός μεταξύ των υπαρχουσών διεργασιών του μοντέλου, να εξαλείφει. Κάθε συμβολικά αναπαριστώμενη κατάσταση μπορεί να περιγράψει –θεωρητικά- έναν πιθανόν μη πεπερασμένο

αριθμό διακριτών καταστάσεων με την βοήθεια συμβολικών μεταβλητών που έχει ορίσει εξ αρχής ο αναλυτής για τις προδιαγραφές του υπό εξέταση μοντέλου.

Στο [89] οι συγγραφείς παρουσιάζουν την τεχνική Athena, που βασίζεται σε μια διαφορετική αναπαράσταση του μοντέλου, όπου σε αντίθεση με τον παραδοσιακό αυτόματο έλεγχο μοντέλων στα πρωτόκολλα ασφαλείας, ένα σύνολο από εκτελέσεις του υπό εξέταση μοντέλου του πρωτοκόλλου, που διαφέρουν μόνο από την ταξινομημένη σειρά του συγχρονισμού των διαφορετικών διεργασιών, αναπαρίστανται στον χώρο των καταστάσεων ως μια κατάσταση. Κάτι τέτοιο επιτυγχάνεται από την έξυπνη επέκταση της αναπαράστασης του επονομαζόμενου strand-χώρου των καταστάσεων. Με μια ημιαυτόματη διαδικασία από την πλευρά του αναλυτή, ο ίδιος έχει την ευχέρεια να μειώσει τον χώρο των καταστάσεων από εκείνες που αντιπροσωπεύουν εκτελέσεις, όπου με βάση τεχνικές θεωρητικής απόδειξης (theorem-proving techniques), δεν θα συνεισφέρουν στην ιδιότητα που βρίσκεται προς επαλήθευση.

Τέλος στο [82] οι συγγραφείς επεκτείνουν ένα πλαίσιο ελέγχου μοντέλων βασισμένο και αυτό στον εισβολέα DY. Η επέκταση αυτή σχεδιάστηκε για την επαλήθευση πρωτοκόλλων ασφαλείας, η οποία και βασίζεται σε πολύ-σύνολα κανόνων προδιαγραφών που καλούνται κανόνες ελέγχου πρόσβασης. Αποτελεί έναν στατικό έλεγχο των κανόνων που μπορεί να ορίσει ο αναλυτής με σκοπό τον έλεγχο γενικών ιδιοτήτων ασφαλείας.

3.5 Σύνοψη των κοινών απειλών ασφαλείας

Ένα Πληροφοριακό Σύστημα το οποίο διαχειρίζεται ευπαθή δεδομένα και βασίζεται επιπλέον στην αξιοποίηση των δυνατοτήτων του διαδικτύου εκτίθεται σε μία σειρά σημαντικών απειλών , οι οποίες απαιτείται να αντιμετωπισθούν αποτελεσματικά. Ως απειλή ορίζεται *μία πιθανή ενέργεια ή ένα γεγονός που μπορεί να προκαλέσει την απώλεια ενός ή περισσοτέρων ιδιοτήτων ασφάλειας ενός πληροφοριακού συστήματος* [36]. Οι απειλές αυτές δεν προέρχονται μόνο από κακόβουλες ενέργειες που προκαλούνται από τρίτους με στόχο την κατοχή ή την απαξίωση πολύτιμων δεδομένων. Είναι πιθανό να

δημιουργηθούν από το εσωτερικό του συστήματος εξαιτίας σχεδιαστικών λαθών και αδυναμιών. Οι κυριότερες από αυτές περιγράφονται παρακάτω, περιλαμβάνοντας και τη σχετική αγγλική ορολογία, με βάση το [38] :

- Παρακολούθηση γραμμών επικοινωνίας (*tapping*) : Παρακολουθώντας τις επικοινωνιακές γραμμές μπορεί κανείς να αποκτήσει μη εξουσιοδοτημένη προσπέλαση σε μετακινούμενα δεδομένα, με πιθανό αποτέλεσμα να παραβιαστεί η ιδιωτικότητά τους.
- Ανάλυση κυκλοφορίας (*traffic analysis*) : Για δεδομένες διευθύνσεις πηγής και προορισμού η παρακολούθηση των διακινούμενων δεδομένων μπορεί να οδηγήσει σε ανάπτυξη ενός προτύπου (*pattern*) κυκλοφορίας. Η στατιστική και μόνο ανάλυση της επικοινωνίας, χωρίς απαραίτητα να γίνεται ανάγνωση των ίδιων των δεδομένων, μπορεί να οδηγήσει σε χρήσιμα συμπεράσματα για κάποιον τρίτο.
- Αποτυχία ή καταστροφή υλικού (*hardware failure*) : Σημαντική απειλή στη διαθεσιμότητα ενός υπολογιστικού συστήματος αποτελεί η ενδεχόμενη καταστροφή του χρησιμοποιούμενου υλικού, είτε από κακόβουλη ενέργεια, είτε από αστοχία υλικού είτε από φυσική αιτία.
- Πλαστογράφηση διευθύνσεων δικτύου (*spoofing*) : Καταργείται η ιδιότητα της μονοσήμαντης αντιστοίχισης των διευθύνσεων δικτύου σε μία συγκεκριμένη θέση, με αποτέλεσμα τα διακινούμενα δεδομένα να χάνουν την ιδιότητα της αυθεντικότητας προέλευσης.
- Υποκλοπή συνθηματικών (*password stealing*) : Ένα συνθηματικό μπορεί να διαρρεύσει σε έναν δυνητικό εισβολέα είτε από αμέλεια του χρήστη του συστήματος είτε μετά από παρακολούθηση των διακινούμενων πακέτων (*sniffing*) είτε με τη χρήση της μεθόδου ωμής δοκιμής (*brute force attack*).
- Αξιοποίηση καταπακτών (*trapdoors exploiting*) : Οι καταπακτές είναι γνωστές ή άγνωστες αδυναμίες των υπηρεσιών του συστήματος που επιτρέπουν την υπέρβαση των μηχανισμών ασφάλειας για την προσπέλαση στους πόρους του συστήματος. Η ύπαρξη των αδυναμιών αυτών γίνεται γνωστή στους εισβολείς έπειτα από δοκιμαστική

ανίχνευση που πραγματοποιούν σε όλες τις θύρες επικοινωνίας του συστήματος (*port-scanning*)

- Μη εξουσιοδοτημένη τροποποίηση (*unauthorized modification*) : Η κακόβουλη τροποποίηση των δεδομένων ενός συστήματος έπεται της παρακολούθησης των γραμμών επικοινωνίας ή της παρείσφρησης στο σύστημα έπειτα από υποκλοπή συνθηματικού ή αξιοποίηση καταπακτών.
- Άρνηση παροχής υπηρεσίας (*Denial of Service*) : Σε αυτή την περίπτωση ο εισβολέας επιχειρεί να επηρεάσει αρνητικά τη διαθεσιμότητα μίας υπηρεσίας, αφού έχει παρεισφρήσει στο σύστημα που την παρέχει. Το ίδιο μπορεί να συμβεί όταν ο εισβολέας καταφέρει εγκαταστήσει λογισμικό που καταναλώνει ανεξέλεγκτα όλους τους διαθέσιμους πόρους του συστήματος ή του δικτύου, με αποτέλεσμα οι υπόλοιπες υπηρεσίες να παραμείνουν ουσιαστικά ανενεργές.
- Κατανεμημένη επίθεση άρνησης παροχής υπηρεσίας (*Distributed Denial of Service*) : Η λογική είναι η ίδια με την άρνηση παροχής υπηρεσίας, με τη διαφορά ότι ο εισβολέας έχει εγκαταστήσει το κακόβουλο λογισμικό σε δεκάδες συστήματα αφού έχει παρεισφρήσει σε αυτά και τα χρησιμοποιεί ως μεσάζοντες (*agents*). Τα συστήματα αυτά με τη σειρά τους επιτίθενται συντονισμένα προς τον τελικό στόχο με δραματικές συνέπειες στους πόρους του συστήματος αυτού, αλλά και στο δίκτυο που οδηγεί προς αυτό.
- Κατάχρηση πόρων (*misuse of resources*) : Μία μη εξουσιοδοτημένη οντότητα είναι πιθανό να υποκλέψει πόρους ενός συστήματος, όπως κύκλους του επεξεργαστή, εύρος ζώνης δικτύου, χωρητικότητα δίσκων, είτε για να εξυπηρετηθούν διεργασίες του εισβολέα είτε για να προκληθεί άρνηση παροχής υπηρεσίας.
- Διάψευση εκτέλεσης ενέργειας (*repudiation of action*) : Μία οντότητα μπορεί να αρνηθεί ότι δημιούργησε και απέστειλε ένα μήνυμα ή ότι τροποποίησε κάποια δεδομένα, εφόσον δεν υπάρχουν επαρκή αποδεικτικά στοιχεία. Ομοίως ο παραλήπτης του μηνύματος μπορεί να διαψεύσει την παραλαβή του και την ανάγνωση του περιεχομένου του.

- Εσωτερικοί κίνδυνοι (*internal threats*) : Είναι πιθανό μέλη του απασχολούμενου προσωπικού σε μία επιχείρηση να υποκλέψουν χρήσιμες πληροφορίες για παράνομη χρήση. Παράλληλα η έλλειψη ασφάλειας στην φυσική πρόσβαση στο υλικό του συστήματος δημιουργεί επιπλέον κινδύνους.
- Πλαστοπροσωπία (*masquerade*) : Στο επίπεδο εφαρμογής είναι πιθανό η προέλευση ενός μηνύματος να φαίνεται διαφορετική από την πραγματική

3.6 Συμπεράσματα Κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκαν οι βασικές αρχές που διέπουν τα πρωτόκολλα ασφαλείας όταν αυτά υπόκεινται έλεγχο με τις τυπικές μεθόδους ανάλυσης συστημάτων και ιδιαίτερα με τον αυτόματο έλεγχο μοντέλων. Ορίστηκαν οι απειλές που συναντιόνται σήμερα εναντίον των πρωτοκόλλων ασφαλείας, οι εγγυήσεις ασφαλείας που προσφέρουν τα πρωτόκολλα, καθώς και προσεγγίσεις μοντελοποίησης πρωτοκόλλων ασφαλείας με εργαλεία ελέγχου μοντέλων όπως το SPIN και το PRISM. Τέλος περιγράφθηκαν σημερινά προβλήματα ασφαλείας, όπου προκαλούν την αποτυχία πρωτοκόλλων ασφαλείας κατά την διάρκεια λειτουργίας τους στις διαδικτυακές επικοινωνίες. Εξειδικευμένα μοντέλα εισβολέων, που ορίζονται με την βοήθεια τεχνικών αυτόματου ελέγχου μοντέλων, προσπαθώντας να εντοπίσουν παραβιάσεις ασφαλείας, κατά τον έλεγχο των πρωτοκόλλων. Στα επόμενα κεφάλαια περιγράφονται τα μοντέλα των εισβολέων ξεχωριστά, που δημιουργήθηκαν στα πλαίσια αυτής της διατριβής.

Κεφάλαιο 4ο

Το Μοντέλο Εισβολέα Πολλαπλών Επιθέσεων

4.1 Εισαγωγή

Ο έλεγχος μοντέλων των κρυπτογραφικών πρωτοκόλλων έχει εξελιχθεί σε μια σημαντική και χρήσιμη μέθοδο για την ανακάλυψη λαθών, καθώς επιτρέπει την δυνατότητα δημιουργίας εξειδικευμένων οντοτήτων-εισβολέων για την παραπλάνηση του τελικού στόχου του υπό-εξέταση πρωτοκόλλου ασφαλείας. Στην παρούσα βιβλιογραφία, τα υπάρχοντα εργαλεία ανάλυσης της ασφάλειας των επικοινωνιακών συστημάτων βασίζονται σε γενικού-σκοπού μοντέλα εισβολέων τα οποία θεωρούνται ως απόγονοι του γνωστού εισβολέα Dolev-Yao. Στο παρόν κεφάλαιο παρουσιάζεται έναν εναλλακτικό μοντέλο εισβολέα το οποίο βασίζεται στην λεπτομερή ανάλυση των βημάτων διεξαγωγής διαφόρων επιθέσεων σε πρωτόκολλα ασφαλείας. Το επονομαζόμενο **μοντέλο εισβολέα-πολλαπλών επιθέσεων (ΕΠΕ)** [6] είναι σχεδιασμένο σαν μια ανοιχτή βάση επιθέσεων όπου ο χρήστης έχει την δυνατότητα να ενοποιήσει και να ολοκληρώσει ποικίλου είδους ενεργειών βασισμένα σε επιμέρους βασικές τακτικές επίθεσης (Attack Tactics) [10]. Οι συγκεκριμένες τακτικές επίθεσης μπορούν να συνδυαστούν με τέτοιο τρόπο έτσι ώστε να δημιουργήσουν επιθέσεις (από απλές μέχρι και σύνθετες) εναντίον πρωτοκόλλων ασφαλείας,

όπως λ.χ. μια επίθεση άρνησης εξυπηρέτησης (Denial of Service Attack). Στην συγκεκριμένη προσέγγιση το μοντέλο του εισβολέα προσαρμόζεται σε αυτό του υπό-εξέταση πρωτοκόλλου, εξαπολύοντας επιθέσεις με σκοπό τον έλεγχο αντοχής του πρωτοκόλλου σε ένα εχθρικό περιβάλλον. Η εγκυρότητα της ανάλυσης ελέγχεται με συγκεκριμένου τύπου επιχειρημάτων-επιβεβαιώσεων που ορίζει ο χρήστης κατά την διάρκεια της ανάλυσης. Με την μέθοδο αυτή ο χρήστης δεν περιορίζεται στον έλεγχο ιδιοτήτων που υπαγορεύει ο εκάστοτε ελεγκτής μοντέλων. Συνήθως οι ιδιότητες αυτές περιορίζονται σε ιδιότητες που έχουν να κάνουν με παραβιάσεις της μυστικότητας και της αυθεντικοποίησης των συμμετεχόντων στο πρωτόκολλα. Με χρήση του μοντέλου ΕΠΕ, δίνεται η δυνατότητα δόμησης επιθέσεων (και επομένως εξέτασης του πρωτοκόλλου ασφαλείας) οι οποίες και δεν μπορούν να ελεγχθούν με τους υπάρχοντες ελεγκτές μοντέλων. Η συγκεκριμένη μεθοδολογία δημιουργίας εισβολέων βασισμένα στο μοντέλο ΕΠΕ δημιουργήθηκε στο περιβάλλον του ελεγκτή μοντέλων SPIN στην γλώσσα εισόδου PROMELA. Με την βοήθεια του ΕΠΕ ελέχθησαν το πρωτόκολλα ασφαλείας μικροπληρωμών PayWord και Micromint που προτάθηκαν από τους Rivest και Shamir στο [81]. Ως αποτέλεσμα του συγκεκριμένου ελέγχου ήταν η ανακάλυψη (μέσω εξαντλητικής επαλήθευσης) ενός αγνώστου μέχρι σήμερα λάθους στο πρωτόκολλο ασφάλειας PayWord.

Στη συνέχεια περιγράφεται μια διαφορετική προσέγγιση στον σχεδιασμό και υλοποίηση ενός μοντέλου εισβολέα. Στην προσέγγιση αυτή παρουσιάζεται ο Εισβολέας Πολλαπλών Επιθέσεων (ΕΠΕ) οποίος βασίζεται στο μοντέλο του Dolev-Yao με την διαφορά ότι το μοντέλο αποτελείται από μια ανοιχτή βάση επιθέσεων, όπου ο αναλυτής έχει την ευχέρεια να ορίσει όποια ενέργεια-επίθεση επιθυμεί να εξαπολύσει προς το υπό εξέταση πρωτοκόλλου. Σε αυτό το κεφάλαιο οι τακτικές επιθέσεων περιγράφονται φορμαλιστικά και στη συνέχεια συνδυάζονται σε έναν απλό Dolev-Yao τύπου εισβολέα μέσα στο περιβάλλον του ελεγκτή μοντέλων SPIN [93]. Επεκτείνοντας την αρχική εισαγωγή η οποία και παρουσιάστηκε στο[10], αναδιοργανώνουμε και ενοποιούμε τις τακτικές των επιθέσεων μέσα στο μοντέλο ΕΠΕ. Παράλληλα, μεταλλάσσουμε τον σχεδιασμό του εισβολέα έτσι ώστε να μπορεί να συνδυάσει με συγκεκριμένο τρόπο τις διαθέσιμες τακτικές επιθέσεων – ανάλογα πάντα με τη σύνοδο του πρωτοκόλλου- με σκοπό την δημιουργία επιθέσεων.

4.2 Χαρακτηριστικά ασφαλείας

Το μοντέλο του εισβολέα ΕΠΕ αποτελείται από 5 (πέντε) τακτικές επιθέσεων, δημιουργώντας κατ' επέκταση 4 (τέσσερις) διαφορετικές επιθέσεις, ενάντια στο υπό-εξέταση πρωτόκολλο. Παρόλο που οι τακτικές επιθέσεων είναι δομούνται από απλά ακολουθιακά βήματα, ο εισβολέας έχει την ελευθερία να δοκιμάσει τις τακτικές των επιθέσεων ως απομονωμένες επιθέσεις (χωρίς δηλαδή τον συνδυασμό τους). Οι τακτικές επιθέσεων που περιλαμβάνονται σε αυτήν την έκδοση του εισβολέα ΕΠΕ είναι οι ακόλουθες:

- *Υποκλοπή Μηνύματος (Message Interception)*
- *Παραβίαση Ακεραιότητας Μηνύματος (Message Integrity Violation)*
- *Αντανακλάσεις (Deflections)*
- *Εκτροπές (Reflections)*
- *Απευθείας Επαναλήψεις (Straight Replays)*

Με βάση τις τακτικές επιθέσεων, δημιουργούνται οι ακόλουθες επιθέσεις:

- *Ελαττωματικών Τύπων (Type Flaws)*
- *Πλαστοπροσωπίας (Impersonation)*
- *Παράλληλης Συνόδου (Parallel Sessions)*
- *Άρνησης Εξυπηρέτησης (Denial of Service)*

Αν και η προτεινόμενη προσέγγιση δεν εγγυάται ότι καλύπτει όλες τις πιθανές επιθέσεις που μπορούν να εξαπολύσει ένας εισβολέας σε ένα πρωτόκολλο, το μοντέλο του εισβολέα ΕΠΕ δεν περιορίζει την γενική ικανότητα του ελεγκτή μοντέλων να 'αιχμαλωτίσει' γενικά λάθη στο υπό-εξέταση πρωτόκολλο (λ.χ. αδιέξοδα ή μη επιτρεπτές καταστάσεις τερματισμού). Επίσης δεν αποκλείεται το γεγονός στο μοντέλο του εισβολέα, να προσπαθήσει να δημιουργήσει καταστάσεις λάθους σε ένα πρωτόκολλο, οι οποίες και δεν συγκαταλέγονται σαν λάθη ιδιοτήτων αυθεντικοποίησης ή μυστικότητας [32][79][82][83].

Βασικό χαρακτηριστικό του εισβολέα ΕΠΕ αποτελεί η δομή του, μιας και συνίσταται από μια ανοιχτή βάση επιθέσεων, όπου και ο αναλυτής μπορεί να υλοποιήσει επιπρόσθετες επιθέσεις, δημιουργώντας έναν πιο ισχυρό επιτιθέμενο. Κάτι τέτοιο θα μεγάλωνε το εύρος των επιθέσεων που θα μπορούσαμε να ελέγξουμε σε ένα πρωτόκολλο, αλλά από την άλλη θα επιβάρυνε τον συνολικό χώρο των καταστάσεων που θα παρήγαγε το μοντέλο στην φάση

της επαλήθευσης. Ως λύση για τις επιθέσεις αυτές που δεν συγκαταλέγονται στις προαναφερθείσες έρχεται να δώσει λύση η γλώσσα LTL και η ικανότητα του ελεγκτή μοντέλων SPIN, να ελέγχει το δένδρο των καταστάσεων με βάση την φόρμουλα LTL που θέλουμε να επαληθευθεί. Με την βοήθεια της εκφραστικότητας της LTL, μπορεί ο αναλυτής να ορίσει καταστάσεις επιθέσεων που χαρακτηρίζονται στην βιβλιογραφία ως μη-αποποίηση της ευθύνης (non-repudiation) [55], δικαιοσύνης (fairness), (accountability)[29], (abuse-freeness) [28] και άλλες ιδιότητες ασφαλείας που αναφέρονται σε πρωτόκολλα ασφαλείας με έμφαση στο ηλεκτρονικό εμπόριο [42][76].

Ένα ενδιαφέρον χαρακτηριστικό της προτεινόμενης μεθοδολογίας, είναι ο συγκριτικά μικρότερος χώρος καταστάσεων που παράγεται, κατά τον έλεγχο μοντέλων με την βοήθεια του εισβολέα ΕΠΕ, στο περιβάλλον του SPIN. Καθώς ο εισβολέας ΕΠΕ αποτελεί μια 'ανοιχτή-προς- κλείσιμο βάση από επιθέσεις, ο αναλυτής έχει την ευχέρεια να συμπεριλάβει στο σχήμα το εισβολέα εκείνες τις τακτικές επιθέσεων που θέλει να δοκιμάσει το πρωτόκολλο ασφαλείας του. Οι τακτικές εν συνεχεία θα συνδυαστούν με σκοπό την διενέργεια εχθρικών ενεργειών προς όλους του συμμετέχοντες του πρωτοκόλλου με σκοπό την ανακάλυψη προβλημάτων ασφαλείας που μπορούν να δημιουργηθούν στο πρωτόκολλο, εξαιτίας της παρέμβασης ενός τρίτου. Κάτι τέτοιο μπορεί να οδηγήσει τον εισβολέα σε εφαρμογή του σε μεγαλύτερα και πιο πολύπλοκα επικοινωνιακά συστήματα, ελέγχοντας για παραβάσεις προκαλούμενες από εισβολείς [87], λαμβάνοντας υπ' όψιν τον παραγόμενο χώρο καταστάσεων κατά την διάρκεια του αυτόματου ελέγχου μοντέλων. Με την βοήθεια της προσέγγισης του εισβολέα ΕΠΕ, που παρουσιάζεται στο κεφάλαιο αυτό, υλοποιήθηκε ένας καινούργιος εισβολές ΠΕ, (ο οποίος παρουσιάζεται στο κεφάλαιο 6) [8], έχοντας την ικανότητα, μετά από μεταβολή της βάσης των τακτικών επιθέσεων, να μπορεί ο εισβολέας να δημιουργεί ενέργειες με σκοπό την εξέταση των πρωτοκόλλων για επιθέσεις DoS. Χάρη σε αυτόν ανακαλύφθηκε ένα άγνωστο σφάλμα τύπου DoS για το πρωτόκολλο HIP [50].
Ενώ στο [11]επίσης έγινε χρήση του περιβάλλοντος του αυτόματου ελεγκτή μοντέλων SPIN, στο [8] μοντελοποιήθηκε ο εισβολέας (και οι απαραίτητες ενέργειές του) στο περιβάλλον του πιθανοκρατικού ελεγκτή μοντέλων PRISM [91], με χρήση της θεωρίας των Μαρκοβιανών αλυσίδων διακριτού χρόνου

(DTMC). Ένα τέτοιο παράδειγμα δείχνει την χρηστικότητα της προτεινόμενης μεθοδολογίας δημιουργίας εισβολέων στον έλεγχο μοντέλων, καθώς και την μεταφερσιμότητά της σε διαφορετικά περιβάλλοντα ελέγχου μοντέλων. Στο συγκεκριμένο κεφάλαιο επίσης παρουσιάζονται ενδεικτικά τα αποτελέσματα της προσομοίωσης και της επαλήθευσης για δύο πρωτόκολλα μίκρο-πληρωμών, το Micromint και το PayWord, τα οποία και προτάθηκαν αρχικά στο [81].

4.3 Το Μοντέλου Εισβολέα Πολλαπλών Επιθέσεων (ΕΠΕ)

Στην ενότητα αυτή θα παρουσιαστεί και θα περιγραφεί φορμαλιστικά το μοντέλο εισβολέα ΕΠΕ. Για την δόμηση του εισβολέα, υιοθετούμε την απαισιόδοξη υπόθεση ότι πριν την εξέταση του πρωτοκόλλου ασφαλείας, ο εισβολέας έχει τον απόλυτο έλεγχο του επικοινωνιακού μέσου, καθώς και όλες τις ικανότητες χειραγώγησης των μηνυμάτων, που παραδέχεται ο εισβολέας των Dolev και Yao (όπως παρουσιάστηκαν σε προηγούμενο κεφάλαιο). Ειδικότερα, ο εισβολέας μπορεί και κρυφακούει όλα τα ανταλλασσόμενα μηνύματα, αναλύοντάς τα με σκοπό τον έλεγχο αν κατέχει το αντίστοιχο κλειδί αποκρυπτογράφησης του μηνύματος. Επίσης ο εισβολέας έχει την ικανότητα να παράγει καινούργια μηνύματα από την *γνώση* του, αποστέλλοντάς τα σε οποιοδήποτε συμμετέχοντα του πρωτοκόλλου.

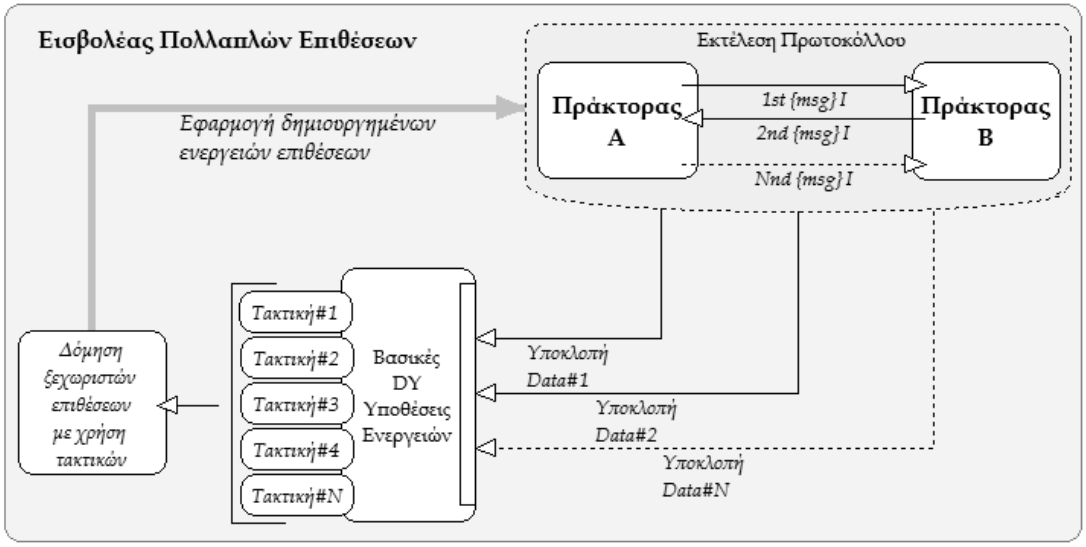
Τα παραγόμενα μηνύματα αυτά, δημιουργούνται από ήδη υπάρχοντα μηνύματα με την εφαρμογή πάνω σε αυτά (4) τεσσάρων βασικών ενεργειών: κρυπτογράφηση, αποκρυπτογράφηση, συναλύσωση (concatenation), προβολή (projection). Οποιαδήποτε προσπάθεια για την απαρίθμηση όλων των ουσιωδών μηνυμάτων που ο εισβολέας μπορεί να παράγει και να αποστείλει, μπορεί να οδηγήσει σε έναν τεράστιο χώρο καταστάσεων, κατά την διάρκεια του αυτοματοποιημένου ελέγχου μοντέλων. Οι προσεγγίσεις του ελέγχου μοντέλων που παρουσιάστηκαν στην προηγούμενη ενότητα, επιδιώκουν την διατήρηση της γενικής περιγραφής του μοντέλου του εισβολέα, με την εφαρμογή ειδικών τεχνικών για την αποφυγή του φαινομένου της έκρηξης του χώρου των καταστάσεων. Παρόλα αυτά, οι τεχνικές αυτές είναι εφαρμόσιμες αλλά και εξαρτώμενες από ένα μικρό σύστημα που αναπαριστά το πρωτόκολλο, το οποίο θέλουμε να εξετάσουμε. Ακόμη και εάν δεν πρόκειται να βρεθεί κάποια

επίθεση ασφαλείας στο σύστημα (μοντέλο) αυτό, τότε παραμένει να υπάρχει η πιθανότητα να υπάρχει κάποια παραβίαση ασφαλείας σε ένα μεγαλύτερο σύστημα το οποίο αναπαριστά το υπό εξέταση πρωτόκολλο. Η συγκεκριμένη θεωρία προτάθηκε αρχικά στο [62] και αποτελεί το γνωστή *συνθήκη απουσίας ολοκλήρωσης του ελέγχου μοντέλων*.

Στο μοντέλο εισβολέα ΕΠΕ, επιδιώκεται μια προσέγγιση περιγραφής λιγότερο γενική αλλά συμπληρωματική για την δημιουργία νέων μηνυμάτων από την πλευρά του εισβολέα η οποίος και διαθέτει μια *ανοιχτή-προς-κλείσιμο* βάση προκαθορισμένων τακτικών επιθέσεων. Η δομή και ο αριθμός όλων των πιθανών μηνυμάτων που μπορεί να κατασκευάσει ο εισβολέας περιορίζεται από τα πρότυπα των μηνυμάτων και τον αριθμό των αρχικών μηνυμάτων των διαθέσιμων τακτικών επιθέσεων. Το μοντέλο του εισβολέα ΕΠΕ μπορεί να θεωρηθεί ως δύο παράλληλες διεργασίες, όπου η πρώτη στοχεύει στην υποκλοπή των ανταλλασσόμενων μηνυμάτων, και η δεύτερη για την εκτέλεση μη-ντετερμινιστικών ακολουθιών από ενέργειες των τυχαία επιλεγμένων (και διαθέσιμων) επιθέσεων, για την παρούσα σύνοδο του υπό εξέταση πρωτοκόλλου (Εικόνα 4.3.1).

Με την λήψη του πρώτου διεφθαρμένου μηνύματος από κάποια οντότητα-θύμα του πρωτοκόλλου, το εφαρμοζόμενο βήμα της επιλεγμένης επίθεσης επιτυγχάνει και το αντίστοιχο ίχνος εκτέλεσης που δημιουργείται, μπορεί να οδηγήσει σε μια άκυρη κατάσταση (η οποία θεωρείται ως σφάλμα ασφαλείας) ή μια παραβίαση μιας προτάσεως ορθότητας που έχει οριστεί στο πρωτόκολλο (και πρέπει να ισχύει σε όλες τις καταστάσεις του).

Εάν η οντότητα-θύμα δεν κάνει αποδεκτό το μήνυμα του εισβολέα, τότε εισέρχεται σε μια κατάσταση λάθους-τερματισμού (fail-stop), η οποία και αποτρέπει την οντότητα να συνεχίσει την παρούσα σύνοδο του πρωτοκόλλου. Η ορθότητα εκτέλεσης ενός πρωτοκόλλου, είτε εκφράζεται σαν μια πρόταση ορθότητας για την προσέγγιση μιας αποδεκτής κατάστασης τερματισμού ή ως μια πρόταση επιβεβαίωσης, δεν περιορίζει τον έλεγχο για λάθη τα οποία δεν συγκαταλέγονται στις ομάδες λαθών μυστικότητας (secrecy) ή αυθεντικοποίησης (authentication).



Εικόνα 4.3.1 Το μοντέλο του Εισβολέα ΕΠΕ

Ένα ατομικό μήνυμα (atomic message) μπορεί να προέρχεται από τα παρακάτω σύνολα:

- *Κλειδιά* (Keys), με όρους που αναπαριστούν τα κλειδιά που χρησιμοποιούνται για την κρυπτογράφηση των μηνυμάτων, έτσι ώστε κάθε κλειδί $k \in Keys$ θα έχει και ένα αντίστροφο κλειδί $k^{-1} \in Keys$, στην ασύμμετρη κρυπτογραφία. Για τις περιπτώσεις τις συμμετρικής κρυπτογραφίας το κλειδί της κρυπτογράφησης και αποκρυπτογράφησης είναι το ίδιο, όπου $k = k^{-1}$
- *Πράκτορες (Agents)*, το οποίο αποτελεί ένα σύνολο από τα ονόματα (ταυτότητες) των έντιμων συμμετεχόντων στο πρωτόκολλο
- *Τυχαίοι Αριθμοί (Nonces)*, οι αποτελούν ένα μη πεπερασμένο σύνολο από τυχαίους αριθμούς. Τα μέλη του συνόλου αυτού χρησιμοποιούνται ως *χρονοσφραγίδες* (timestaps), όπου επισυνάπτονται στα μηνύματα του πρωτοκόλλου με σκοπό την ένδειξη της χρονική δημιουργίας του κάθε μηνύματος
- *Δεδομένα (Data)*, τα οποία αποτελούν ένα σύνολο από συμβολοσειρές (strings), που ανταλλάσσονται μεταξύ των συμμετεχόντων στο πρωτόκολλο. Από την πλευρά του εισβολέα, δεδομένα μπορούν να παραχθούν χωρίς όμως απαραίτητα αυτά, να έχουν κάποιο νόημα. Το είδος των δεδομένων αυτών, θα ονομάζονται *πλαστά δεδομένα* (bogus data), *bg_data*, και θα χρησιμοποιούνται από τον εισβολέα για τις

ενέργειες διαφθοράς πάνω σε προηγούμενα υποκλεμμένα μηνύματα, τα οποία στις περισσότερες περιπτώσεις είναι κρυπτογραφημένα

Θα χρησιμοποιήσουμε τον συμβολισμό I ,*Intruder*, για να αναφερθούμε στον εισβολέα, όπου $I \notin Agents$. Επίσης ορίζουμε την δυαδική σχέση: *is_key_of* = {(k, *id*): $k \in Keys$, $id \in Agents \cup \{I\}$, "*key k is used by the participant id*"}, έτσι ώστε $|is\_key\_of\ (k)| = 1$ στην περίπτωση της ασύμμετρης ή $|is\_key\_of\ (k)| = 2$ στην περίπτωση της συμμετρικής κρυπτογράφησης, αντίστοιχα.

Το σύνολο *Msgs* των ανταλλασσόμενων μηνυμάτων ορίζεται επαγωγικά πάνω στην ασύνδετη σχέση $AMsgs = Keys \cup Agents \cup \{I\} \cup Nonces \cup Data$,η οποία αναπαριστά το σύνολο των ατομικών μηνυμάτων ($Set_i \cap Set_j = \varnothing$ για κάθε δύο Set_i, Set_j από τα ενοποιημένα σύνολα). Θα έχουμε:

- Εάν το $\alpha \in AMsgs$ τότε $\alpha \in Msgs$.
- Εάν $msg_x \in Msgs$ και $msg_y \in Msgs$ τότε $msg_x \cdot msg_y \in Msgs$, όπου η $\cdot$ αναπαριστά την συναλύσωση μηνυμάτων (concatenation)
- Εάν $msg \in Msgs$ και $k \in Keys$ τότε $\{msg\}_k \in Msgs$.

Κάθε ag $\in$ *Agents* μπορεί να επιχειρήσει να εκτελέσει το πρωτόκολλο για ένα περιορισμένο αριθμό συνόδων, έστω $\#Ses_{\text{ag}}$, όπου κάθε τέτοια απόπειρα αποτελεί μια ξεχωριστή σύνοδο πρωτοκόλλου *noSes*, όπου $1 \le noSes \le \#Ses_{\text{ag}}$. Σε μια σύνοδο πρωτοκόλλου, ο πράκτορας ag παίζει είτε το ρόλο της εναρκτήριας οντότητας (Initiator) είτε της οντότητας του ανταποκριτή (Responder). Ορίζουμε ως $sent_n^{\text{ag}_{noSes}}$ την πεπερασμένη μήκους συναλύσωσης μηνυμάτων που αποστάλθηκαν από τον ag $\in$ *Agents* κατά την διάρκεια της συνόδου *noSes*:

$$sent_n^{\text{ag}_{noSes}} = (sent_{n-1}^{\text{ag}_{noSes}} \cdot msg_n),$$

με τον πρώτο όρο να αποτελεί την *κενή ακολουθία* (null sequence), η οποία θα είναι $sent_0^{\text{ag}_{noSes}} = ()$. Η ακολουθία $sent_n^{\text{ag}_{noSes}}$ αναπαριστά το ιστορικό του πράκτορα ag (history) για την σύνοδο *noSes*, μετά την αποστολή του msg_n. Ορίζουμε ως $rcvd_n^{\text{ag}_{noSes}}$ την πεπερασμένη μήκους συναλύσωσης μηνυμάτων που ελήφθησαν από τον ag $\in$ *Agents* κατά την διάρκεια της συνόδου *noSes*. Σε ένα οποιοδήποτε χρονικό στιγμιότυπο, ορίζεται ως *γνώση του συμμετέχοντα* (*participant's knowledge*) για την συγκεκριμένη σύνοδο του πρωτοκόλλου ως:

$$\text{ag}_{\text{knowledge}} = \bigcup_{\text{ag}_j} \{rcvd^{ag_j}_{\max(i)}\} \cup \text{ag}_{in\_knowledge},$$

για κάθε $1 \leq j \leq \#Ses_{\text{ag}}$, όπου $\text{ag}_{in\_knowledge}$ αναπαριστά την *αρχική βάση γνώσης* του ag (κλειδιά, πράκτορες, ταυτότητες κα) και $i > 0$ αναπαριστώντας τους όρους των συναλυσώσεων ληφθέντων ακολουθιών. Μια σύνοδος πρωτοκόλλου για έναν έντιμο πράκτορα $\text{ag} \in Agents$ ορίζεται φορμαλιστικά ως 5-tuple $\langle \text{ag}, j, \text{ag}_{knowledge}, \text{ag}^{j}_{history}, P \rangle$, όπου $1 \leq j \leq \#Ses_{\text{ag}}$, με το P να αποτελεί μια *περιγραφή διεργασίας* (process description) το οποίο και ουσιαστικά περιέχει μια ακολουθία ενεργειών που πρέπει να εκτελεστούν. Θεωρούμε τις αυτούσιες ενέργειες-εντολές αποστολή, **send** και λήψη, **receive** για την αποστολή και λήψη μηνυμάτων από/προς τους συμμετέχοντες του πρωτοκόλλου. Οι υποθέσεις που συζητήθηκαν σε προηγούμενες παραγράφους για τον εισβολέα των Dolev και Yao, υπονοούν ότι για μια συγκεκριμένη χρονική στιγμή, η συνολική γνώση του εισβολέα για την παρούσα σύνοδο πρωτοκόλλου θα είναι:

$$I_{knowledge} = \bigcup_{\text{ag}_j} \{sent^{ag_j}_{\max(i)}\} \cup I_{in\_knowledge},$$

για κάθε $1 \leq j \leq \#Ses_{\text{ag}}$, $\text{ag} \in Agents \cup \{I\}$, όπου ο συμβολισμός $I_{in\_knowledge}$ αναπαριστά την αρχική βάση-γνώσης για τον εισβολέα, και όπου $i \geq 1$ να αναπαριστά τους όρους που υπέκλεψε από την εφαρμογή του πρωτοκόλλου.

Το *μοντέλο του πρωτοκόλλου* (protocol model) ορίζεται ως μια ασύγχρονη σύνθεση μοντέλων για κάθε σύνοδο του πρωτοκόλλου, συμπεριλαμβανομένου και του μοντέλου του εισβολέας που παρεμβάλλεται πάντα της επικοινωνίας των έντιμων οντοτήτων. Η συμπεριφορά του εισβολέα θα εξαρτάται πάντα από τις ορισμένες και διαθέσιμες τακτικές επιθέσεων που θα έχει στην δομή του. Οι τακτικές επιθέσεων επιλέγονται μη ντετερμινιστικά και εκτελούνται σαν μια μοναδική νηματοειδής ενέργεια. Κάθε πιθανή *εκτέλεση* του μοντέλου ανταποκρίνεται σε μια πεπερασμένη εναλλακτική *ακολουθία καθολικών καταστάσεων* (global states) και ενεργειών, όπου:

$\tau = s_0\, \alpha_1\, s_1\, \alpha_2 \ldots s_n$, για $n \in N$ έτσι ώστε $s_{i-1} \xrightarrow{a_i} s_i$ για $0 < i \leq n$, και για την σχέση μετάβασης $\rightarrow$ που ορίζεται ως :

$$\rightarrow \subseteq S \times PS \times A \times Msgs \times S,$$

όπου S είναι το σύνολο των καθολικών καταστάσεων, PS είναι το σύνολο των εκτελούμενων συνόδων του πρωτοκόλλου και με A να αναπαρίσταται το σύνολο

τον ονομάτων-ενεργειών. Μια σημαντική τεχνική που έχουμε εισάγει στο προτεινόμενο μοντέλο του εισβολέα, είναι η ικανότητά του να συνδυάζει τις διαθέσιμες τακτικές επιθέσεων με σκοπό την δημιουργία πακέτων επιθέσεων, πιο πολύπλοκων απαιτήσεων.

Πίνακας 4.3.1 Πίνακας συμβολισμού για τον εισβολέα ΕΠΕ

$\{msg\}_k$: Το μήνυμα *msg* είναι κρυπτογραφημένο με το κλειδί *k*	$ag_{knowledge}$ Η γνώση του πράκτορα ag για την τρέχουσα σύνοδο του πρωτοκόλλου σε ένα συγκεκριμένο χρονικό στιγμιότυπο
$msg_x \cdot msg_y$ Συναλύσωση των μηνυμάτων msg_x και msg_y	**send** (*s*, *r*, *msg*) Η ενέργεια όπου ο *s* στέλνει ένα μήνυμα *msg* στον *r*
is_key_of (*k*) Επιστρέφει τους ιδιοκτήτες του κλειδιού *k*	**receive**(*r*, *s*, *msg*): Η ενέργεια όπου ο *r* λαμβάνει ένα μήνυμα *msg* από τον *s*
$\#Ses_{ag}$ Ο μέγιστος αριθμός των αριθμών συνόδων του πρωτοκόλλου που επιτρέπεται στον πράκτορα ag να συμμετάσχει ως εναρκτήρια οντότητα ή ως ανταποκριτής	$\langle ag, j, ag_{knowledge}, a g^{j}_{history}, P \rangle$ Η πλειάδα η οποία αναπαριστά μια σύνοδο πρωτοκόλλου *j* για έναν πράκτορα ag σε ένα συγκεκριμένο χρονικό στιγμιότυπο; το *P* αποτελεί μια περιγραφή διεργασίας η οποία δίνεται ως μια ακολουθία από ενέργειες που πρέπει να εκτελεστούν
$sent_n^{ag_{noSes}}$ Πεπερασμένη μήκους ακολουθία συναλύσωσης μηνυμάτων που στάλθηκαν από τον ag στη διάρκεια της συνόδου *noSes*	$I_{knowledge}$ Η γνώση του εισβολέα για την τρέχουσα σύνοδο του πρωτοκόλλου
$rcvd_n^{ag_{noSes}}$ Πεπερασμένη μήκους ακολουθία συναλύσωσης μηνυμάτων που ελήφθησαν από τον ag στη διάρκεια της συνόδου *noSes*	$s_{i-1} \xrightarrow{a_i} s_i$ Μετάβαση απ μια καθολική κατάσταση s_{i-1} στην κατάσταση s_i ως αποτέλεσμα της ενέργειας α_i
$a g^{noSes}_{history}$ Το ιστορικό του πράκτορα ag για την σύνοδο του πρωτοκόλλου *noSes* σε ένα συγκεκριμένο χρονικό στιγμιότυπο	*exists*(*str*, *msg*) Κατηγόρημα boolean που επιστρέφει αποτέλεσμα εάν ή όχι μια συμβολοσειρά *str* εμφανίζεται στο μήνυμα $msg \in Msgs$

Το αποτέλεσμα αυτού θα είναι η δημιουργία ενός ενοποιημένο μοντέλου, με βάση τις τακτικές επιθέσεων, (στο SPIN και στην PROMELA θα πρόκειται για ακολουθιακά βήματα εκτέλεσης), προσπαθώντας να προσομοιώσει πολύπλοκα σενάρια επιθέσεων, όπως για παράδειγμα επιθέσεις που είναι γνωστές ως επιθέσεις άρνησης εξυπηρέτησης (DoS attack). Στην συνέχεια παρέχεται ο

πίνακας 4.3.1, με τον συμβολισμό που θα χρησιμοποιηθεί για την περιγραφή των επιθέσεων.

4.4 Τακτικές επιθέσεων του εισβολέα ΕΠΕ

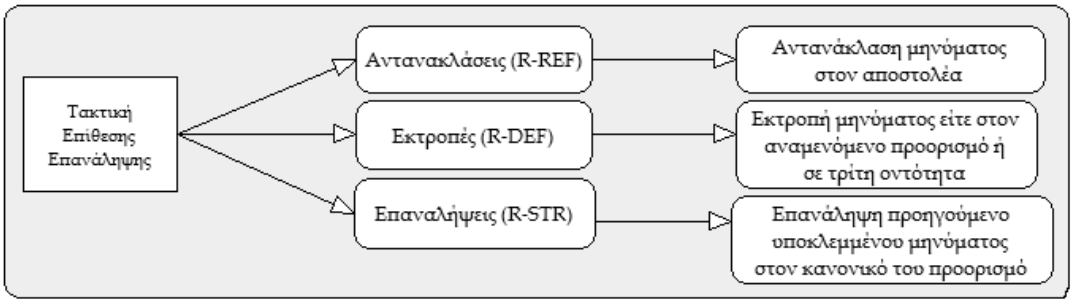
Σε αυτή την ενότητα παρουσιάζονται και περιγράφονται φορμαλιστικά μια σειρά από βασικές τακτικές επιθέσεων που επιλέξαμε να δομούν τον εισβολέα. Πρώτα παρουσιάζονται οι απλές τακτικές επίθεσης, οι οποίες και με τον κατάλληλο συνδυασμό τους μπορούν να δημιουργήσουν, πιο πολύπλοκες ενέργειες επιθέσεων. Έτσι, στη συνέχεια παρουσιάζονται και οι επιθέσεις που συνδυαστικά μπορούν να δημιουργηθούν από το μοντέλο του εισβολέα ΕΠΕ. Αξίζει να σημειωθεί ότι οι συγκεκριμένες τακτικές επιθέσεων αποτελούν τις κυριότερες επιθέσεις που παρουσιάζονται σήμερα στην βιβλιογραφία.

4.4.1 Τακτική Επίθεσης Υποκλοπής Μηνύματος (INCPT)

Η τακτική επίθεσης υποκλοπής ενός μηνύματος λαμβάνει χώρα μετά από την ύπαρξη μιας ενέργειας **send**(ag, *v*, *msg*), από έναν πράκτορα ag, $v \in Agents$ και ενός μηνύματος $msg \in Msgs$, εάν δεν υπάρξει κάποιο **receive**(*v*, *u*, *msg*) με $u \in$ {ag, *I*} στο ίχνος της εκτέλεσης του συστήματος. Το ίδιο ακριβώς θα ισχύει και στην περίπτωση ενός κρυπτογραφημένου μηνύματος, έστω $\{msg\}_k$ όπου δεν θα υπάρξει ενέργεια **receive** (*v*, *u*, $\{msg\}_k$). Είναι φανερό ότι η συγκεκριμένη τακτική επίθεσης είναι προεπιλεγμένη (και άρα πάντα ενεργοποιημένη) για το μοντέλο του εισβολέα, εξαιτίας των παραδοχών που πάρθηκαν για την τοποθέτησή του στο επικοινωνιακό σύστημα (ο εισβολέας είναι ο κυρίαρχος όλης της επικοινωνίας που διεξάγεται μέσω του υπό εξέταση πρωτοκόλλου).

4.4.2 Τακτική Επίθεσης Επανάληψης (R-REF, R-DEF, R-STR)

Οι τακτικές επιθέσεων επανάληψης λαμβάνουν χώρα όταν ο εισβολέας αποπροσανατολίζει ένα μήνυμα που υπέκλεψε (είτε μια διεφθαρμένη έκδοση αυτού) στην ίδια ή διαφορετική σύνοδο πρωτοκόλλου, σε έναν μια ή περισσότερες οντότητες. Υιοθετούμε την ταξινόμηση των επιθέσεων επανάληψης που παρουσιάζονται στο [67], όπως φαίνονται και στην εικόνα 4.4.1.



Εικόνα 4.4.1 Τακτικές Επιθέσεων Επανάληψης

Αντανακλάσεις (REFlections,R-REF):

Σε μια *τακτική επίθεσης αντανάκλασης*, ο εισβολέας αποστέλλει ένα υποκλεμμένο μήνυμα (ή διεφθαρμένο) πίσω στον αποστολέα του. Και σε αυτή την περίπτωση διακρίνονται περιπτώσεις αντανάκλασης μηνύματος στην ίδια σύνοδο του πρωτοκόλλου (*Run-internal reflections*) ή σε διαφορετική σύνοδο (*Interleaving reflections*) του πρωτοκόλλου αλλά στον ίδιο τον αποστολέα (που συμμετέχει και στις δυο). Η επίθεση *R-REF* λαμβάνει χώρα μετά από μια ενέργεια **send** (*v*, ag, *msg*), με το μήνυμα *msg* να αναπαριστά ένα μη κρυπτογραφημένο μήνυμα $msg \in Msgs$ ή μετά από μια ενέργεια **send**(*v*, ag, $\{msg\}_k$) με $I \notin is\_key\_of\ (k) \wedge k^{-1} \in I_{knowledge}$. Το αποτέλεσμα των προηγούμενων ενεργειών οδηγεί σε μια καθολική κατάσταση όπου:

$$exists(msg, sent^{v_j}_{\max(i)}) = \texttt{true}$$ ή αντίστοιχα $$exists(\{msg\}_k, sent^{v_j}_{\max(i)}) = \texttt{true}$$

για $1 \leq j \leq \#Ses_v$, με $i \geq 1$ αναπαριστώντας τους όρους από μια ακολουθία συναλύσωσης μηνυμάτων που έχουν υποκλαπεί. Στην ίδια περίπτωση, συγκαταλέγεται και η ενέργεια του εισβολέα, ο οποίος μετά

Όταν εφαρμόζεται η επίθεση αντανάκλασης μηνύματος, ο εισβολέας αλλάζει το μήνυμα *msg* το οποίο βασίζεται στην γνώση του $I_{knowledge}$

χρησιμοποιώντας το καινούργιο $msg' \in Msgs$ μέσα από μια ενέργεια **send**(*I*, *v*, *msg΄*) ή αντίστοιχα **send**(*I*, *v*, $\{msg'\}_{k'}$) για κάποιο $k' \in I_{knowledge}$ έτσι ώστε $v \in is\_key\_of\ (k')$. Η επίθεση R-REF επιτυγχάνει μόνο όταν ο *v* επιχειρήσει την ενέργεια **receive**(*v*, *I*, *msg΄*) ή αντίστοιχα την ενέργεια **receive** (*v*, *I*, $\{msg'\}_{k'}$) με τα συνακόλουθα αποτελέσματα:

Run-internal reflections:

$$exists(msg', rcvd^{v_j}_{\max(i)}) = \texttt{true}$$ ή $$exists(\{msg'\}_{k'}, rcvd^{v_j}_{\max(i)}) = \texttt{true}$$

Κλασσικές ή interleaving reflections:

$\exists\, j' \neq j$: *exists*(msg', $rcvd^{v_{j'}}_{\max(i)}$) = `true` ή *exists*($\{msg'\}_{k'}$, $rcvd^{v_{j'}}_{\max(i)}$) = `true`

Εκτροπές (DEFlections,R-DEF):

Σε μια *επίθεση εκτροπής* ο εισβολέας αποκλίνει ένα πιθανόν διεφθαρμένο μήνυμα σε κάποιο συμμετέχοντα ο οποίος μπορεί να είναι είτε ο κανονικός δέκτης του μηνύματος, είτε ο αποστολέας αυτού. Και σε αυτή τη περίπτωση οι επιθέσεις εκτροπής στην ίδια σύνοδο του πρωτοκόλλου θα ονομάζονται *Run-internal deflections*. Οι επιθέσεις σε διαφορετικές συνόδους του πρωτοκόλλου (*Interleaving deflections)* με χρήση μηνυμάτων από συνόδους πρωτοκόλλου που έχουνε τερματίσει.

Επαναλήψεις STraight (R-STR):

Σε μια επίθεση επανάληψης *straight* ο εισβολέας εξαποστέλλει ένα προηγούμενο υποκλεμμένο μήνυμα στον κανονικό προορισμό του. Εξαρτώμενο από το εάν η συγκεκριμένη επίθεση εφαρμόζεται στην ίδια, σε διαφορετική ή σε ταυτόχρονη σύνοδο του πρωτοκόλλου, οι επιθέσεις straight χαρακτηρίζονται και αυτές είτε ως *run-internal*, *interleaving* ή *κλασσικές* επιθέσεις επανάληψης.

4.4.3 Τακτική Επίθεσης Παραβίασης Ακεραιότητας (INTV)

Μια επίθεση *παραβίασης ακεραιότητας* μηνύματος, λαμβάνει χώρα όταν ένα ληφθέν μήνυμα *msg* ή $\{msg\}_k$ προερχόμενο από μια ενέργεια **receive**(*v*, *u*, *msg*) ή **receive**(*v*, *u*, $\{msg\}_k$) αντίστοιχα, αντικαθίσταται από ένα *msg'* ή ένα $\{msg\}_k' = \{msg\}_k \cdot bg\_data$. Το καινούργιο (διεφθαρμένο από τον εισβολέα) μήνυμα, μπορεί να χρησιμοποιηθεί αργότερα, κυρίως για τον συνδυασμό της τακτικής αυτής για την ενοποίηση άλλων πιθανών επιθέσεων.

Με την βοήθεια όλων των παραπάνω τακτικών επιθέσεων που ορίστηκαν, θα μπορούσαμε να τις συνδυάσουμε με τον κατάλληλο τρόπο, για την δημιουργία ολοκληρωμένων, πιο εφαρμόσιμων και ρεαλιστικών επιθέσεων, προς τις έντιμες οντότητες που χρησιμοποιούν το πρωτόκολλο. Αξίζει να σημειωθεί, ότι με την βοήθεια του αυτόματου ελέγχου μοντέλων, οι επιθέσεις αυτές εξετάζονται για την αποτελεσματικότητά τους, και καθεμία ξεχωριστά, αλλά και ως συνδυασμός των παρακάτω επιθέσεων.

4.4.4 Επίθεση Ελαττωματικών Τύπων (Type flaw, TFLAWS)

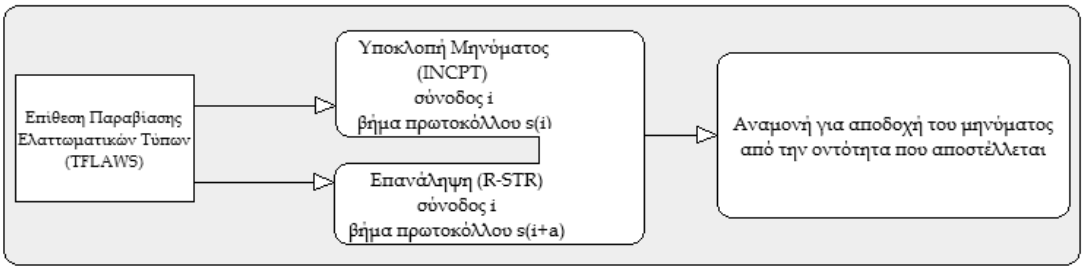
Μια επίθεση *ελαττωματικών τύπων* εμφανίζεται όταν ο παραλήπτης ενός μηνύματος αποδέχεται ένα μήνυμα ως σωστό, αλλά υποθέτει μια διαφορετική διερμηνεία της ακολουθίας των δυφίων από ότι αυτή που έχει προσδώσει ο συμμετέχοντας του πρωτοκόλλου που δημιούργησε το μήνυμα αυτό. Οι επιθέσεις *TFLAWS* ακολουθούν την ακολουθία ενεργειών των τακτικών επιθέσεων επαναλήψεων που συζητήθηκαν στις προηγούμενες παραγράφους (*message replays, R-REF, R-DEF, R_STR*), με τον συνδυασμό τους προαιρετικά με την τακτική επίθεσης *υποκλοπής μηνύματος* (*message interception*, *INTCP*), με σκοπό την αποτροπή λήψης του αυθεντικού μηνύματος από τον παραλήπτη, σε ένα συγκεκριμένο βήμα του πρωτοκόλλου. Ο εισβολέας *I*, μπορεί να πυροδοτήσει μια επίθεση *TFLAWS*, μετά από την ενέργεια διαφθοράς ενός υποκλεμμένου μηνύματος $msg \in Msgs$ το οποίο βρίσκεται στην γνώση $I_{knowledge}$, δημιουργώντας ένα νέο μήνυμα $msg' \in Msgs$. Η αμέσως επόμενη ενέργεια για τον εισβολέα είναι είτε **send** (I, v, msg') ή **send** (I, v, $\{msg'\}_{k'}$) για κάποιο $k' \in I_{knowledge}$ όπου $v \in is\_key\_of\ (k')$. Η συγκεκριμένη επίθεση θα επιτυγχάνει εάν υπάρξει μια καθολική κατάσταση του πρωτοκόλλου όπου, μετά την ενέργεια **receive** (v, I, msg') ή αντίστοιχα **receive** (v, I, $\{msg'\}_{k'}$) υπάρξει ένα ατομικό μήνυμα *amsg*, ώστε:

$$exists(amsg,\ rcvd^{v_j}_{\max(i)}) = \text{true},\ 1 \le j \le \#Ses_v$$

με $i \ge 1$ αναπαριστώντας τους όρους της ακολουθίας $rcvd^{v_j}_n$ για δύο σύνολα Set_i και Set_j της ασύνδετης ένωσης *Amsgs*, $amsg \in Set_i \cap Set_j$.

Ένας χρήσιμος που εισαγάγουμε για την περιγραφή αυτής της επίθεσης είναι και αυτός του *βήματος του πρωτοκόλλου* (*protocol step), s(i)*. Ένα βήμα πρωτοκόλλου *s(i)* θα ορίζεται ως η αριθμητική ακολουθία των διακεκριμένων βημάτων αυθεντικοποίησης που απαιτούνται για την επιτυχή ολοκλήρωση μιας συνόδου πρωτοκόλλου (Εικόνα 4.4.2). Η περιγραφόμενη καθολική κατάσταση παραβίασης που δημιουργείται από την συγκεκριμένη επίθεση, εκφράζει το γεγονός ότι ουσιαστικά υπάρχει περίπτωση ένα ατομικό μήνυμα, το οποίο αρχικά προορίστηκε για ένα βήμα πρωτοκόλλου *s(i)* (π.χ. ένας τυχαίος αριθμός,

Nonce), να διερμηνεύεται ως ένα σωστό μήνυμα άλλου τύπου (πχ. αριθμητικά δεδομένα), σε ένα άλλο βήμα πρωτοκόλλου της ίδιας συνόδου, *s(i+a)*.



Εικόνα 4.4.2 Επίθεση Ελαττωματικών Τύπων

Κάτι τέτοιο μπορεί να συμβεί μόνο όταν και οι δύο τύποι μηνύματος έχουν το ίδιο μήκος ακολουθίας δυφίων, και ο εισβολέας από την πλευρά του αντικαθιστά με ένα ατομικό μήνυμα ίδιας δομής δυφίων, έτσι ώστε να ξεγελάσει τον τελικό αποδέκτη να το αποδεχτεί, με βάση την περιγραφή διεργασίας του *P*. Σε αυτό το σημείο, αξίζει να ειπωθεί ότι οι *TFLAWS* επιθέσεις [43], μπορεί να μην οδηγήσουν σε άμεση παραβίαση της ασφάλειας ενός πρωτοκόλλου, αφού υπάρχει η δυνατότητα στο απλή συμβολοσειρά των δυφίων του ατομικού μηνύματος που χρησιμοποιείται από τον εισβολέα *I* να είναι άγνωστο σε αυτόν (και ως εκ' τούτου η ιδιότητα μυστικότητας του μηνύματος να είναι απαραβίαστη).

Παρόλα αυτά, εάν για παράδειγμα ένας *τυχαίος αριθμός Nonce* χρησιμοποιούνταν ως κλειδί, κάτι τέτοιο δεν θα ήταν η καλύτερη επιλογή, γιατί στόχος δημιουργίας των τυχαίων αριθμών είναι να είναι μοναδικοί για κάθε σύνοδο του πρωτοκόλλου σε αντίθεση με κλειδιά τα οποία γενικά πρέπει να θεωρούνται μη-προβλέψιμα. Οι επιθέσεις *TFLAWS* μπορούν να οδηγήσουν σε αποτυχίες ιδιοτήτων ασφαλείας πέρα από αυτές που συγκαταλέγονται στις ομάδες ιδιοτήτων μυστικότητας ή αυθεντικοποίησης, όπως για παράδειγμα ιδιότητες ανωνυμίας ή μη-αποποίηση της ευθύνης [55].

4.4.5 Επίθεση Πλαστοπροσωπίας (IMPersonation ,IMP)

Μια μη-ασφαλή για την επίδραση μιας επίθεσης πλαστοπροσωπίας *IMP* αποτελεί οποιαδήποτε κατάσταση στην οποία ο εισβολέας *I* μπορεί να αναγνώσει τα περιεχόμενα ενός μηνύματος που στάλθηκαν από κάποιο

πράκτορα ag ∈ *Agents*, ο οποίος παίζει το ρόλο της εναρκτήριας οντότητας σε μια καινούργια σύνοδο του πρωτοκόλλου:

$$\{\exists\ sent_1^{ag_{noSes}} \in I_{knowledge},\ ag \in Agents,\ 1 \leq noSes \leq \#Ses_{ag}\ :$$

$$\{\ sent_1^{ag_{noSes}} = msg \text{ για κάποιο μη κρυπτογραφημένο μήνυμα } msg \in Msgs\}$$

$$\vee\ \{\ sent_1^{ag_{noSes}} = \{msg\}_k :\ is\_key\_of\ (k) = I \vee (is\_key\_of\ (k) \neq I \wedge k^{-1} \in I_{knowledge})\}\}$$

Η επίθεση *IMP* λαμβάνει χώρα όταν ο εισβολέας διενεργεί τις επόμενες τρεις ακολουθιακές ενέργειες κατά ενός πράκτορα-θύματος $v \in Agents$, όπου $v \notin is\_key\_of\ (k)$ και $v \neq$ ag:

send (I, v, msg'), **receive** (I, v, $sent_1^{v_{newSes}}$), **send** (I, ag, $sent_1^{v_{newSes}}$)

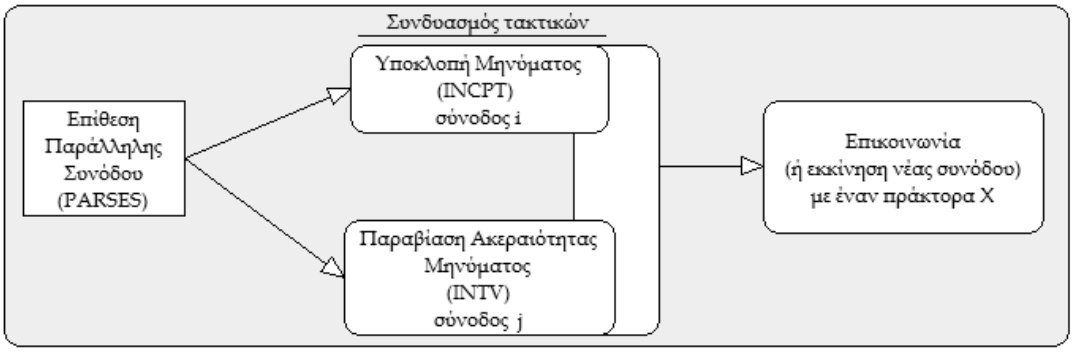
όπου $msg' = sent_1^{ag_{noSes}}$, όπου το τελευταίο αποτελεί μη κρυπτογραφημένο μήνυμα ή $msg' = \{msg\}_{k'}$, με το $k' \in I_{knowledge}$ και $v \in is\_key\_of\ (k')$. Επίσης, το v_{newSes} αποτελεί έναν μοναδικό αναγνωριστικό της συνόδου του πρωτοκόλλου *newSes*, στην οποία το θύμα *v ενεργεί ως οντότητα ανταποκριτής* και το Boolean κατηγόρημα *exists*(v, $sent_1^{v_{newSes}}$) είναι false. Εάν το κατηγόρημα αυτό ήταν true, τότε ο πράκτορας ag θα αντιλαμβάνονταν ότι ο ανταποκριτής της συνόδου του πρωτοκόλλου ag_{noSes} δεν είναι η αρχικά επιλεγμένη οντότητα προς επικοινωνία, και ως επακόλουθο θα τερματίσει την σύνοδο.

4.4.6 Επίθεση Παράλληλης Συνόδου (Parallel session, PARSES)

Οι επιθέσεις παράλληλων συνόδων ενός πρωτοκόλλου, λαμβάνουν χώρα με την *ακολουθιακή διενέργεια επιθέσεων επανάληψης* σε ταυτόχρονες συνόδους του υπό εξέταση πρωτοκόλλου, στις οποίες ο εισβολέας χειραγωγεί τις συμμετέχουσες οντότητες (εναρκτήρια οντότητα ή οντότητα ανταποκριτή), με σκοπό την αποτροπή επιτυχούς τερματισμού μιας ή περισσότερων συνόδων του πρωτοκόλλου. Ο εισβολέας μπορεί, κάτω από συγκεκριμένες συνθήκες να χρησιμοποιήσει σαν διαλόγους τύπου *επανάληψης, ως μάντης (oracle),* με σκοπό να αποκαλύψει στοιχεία τέλεια κρυπτογραφημένων μηνυμάτων. Ένα τέτοιο παράδειγμα παρουσιάζεται στην επίθεση *BAN-Yahalom* στο [75], παραβιάζοντας επιτυχώς την ασφάλεια ιδιοτήτων που δεν συγκαταλέγονται στις ιδιότητες μυστικότητας ή αυθεντικοποίησης.

Σε μια επίθεση *PARSES* (Εικόνα 4.4.3), η σειρά εκτέλεσης *τ* περιλαμβάνει ένα σύνολο κυκλικών ενεργειών, που ξεκινούν με την διενέργεια μιας **send** (ag, *ν*, *msg*) ή **send** (ag, *ν*, $\{msg\}_k$) έχοντας ως αποτέλεσμα:

$exists(msg, sent^{ag_j}_{max(i)})$ = `true` ή αντίστοιχα $exists(\{msg\}_k, sent^{ag_j}_{max(i)})$ = `true`.



Εικόνα 4.4.3 Επίθεση παράλληλης συνόδου

Ο εισβολέας *I* δημιουργεί μια καινούργια σύνοδο πρωτοκόλλου ν'_{newSes} η απαντά σε μια ανοιχτή σύνοδο ν'_m (με $\nu' \in Agents$ συμπεριλαμβανομένου των ag και *ν*), για την οποία η τελευταία ενέργεια της περιγραφής της διεργασίας *P* δεν περιλαμβάνεται επίθεμα στην ακολουθία εκτέλεσης *τ*. Η επίθεση εφαρμόζεται πιθανόν μετά από ένα υποκλεμμένο μήνυμα $msg \in Msgs$ (βασιζόμενο στην γνώση $I_{knowledge}$), προκαλώντας την αποστολή ενός $msg' \in Msgs$ με την ενέργεια **send** (*I*, *ν΄*, *msg΄*) ή **send** (*I*, *ν΄*, $\{msg'\}_{k'}$) για κάποιο $k' \in I_{knowledge}$ έτσι ώστε $\nu' \in is\_key\_of(k')$. Η επανάληψη αυτή επιτυγχάνει εάν ο κύκλος εκτέλεσης τερματίσει με μια ενέργεια *receive* από τον *ν΄*, φανερώνοντας έτσι μια καθολική κατάσταση όπου:

$exists(msg', rcvd^{\nu'_m}_{max(i)})$ = `true` η αντίστοιχα $exists(\{msg'\}_{k'}, rcvd^{\nu'_m}_{max(i)})$ = `true`,

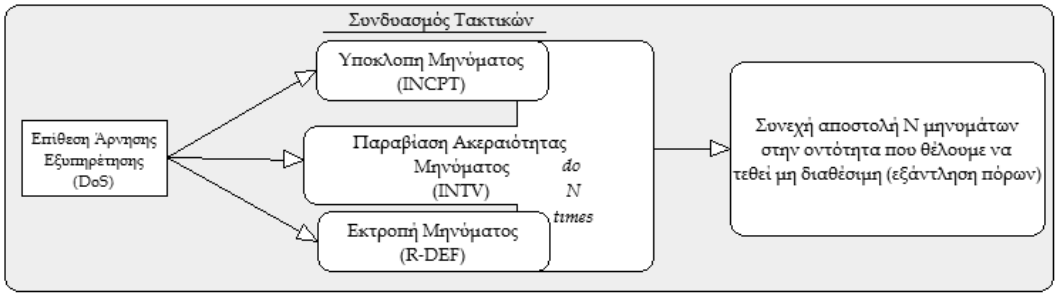
με το max(*i*) = 1, εάν το *m* μια καινούργια σύνοδο πρωτοκόλλου (*newSes*). Ένας αριθμός διαδοχικών τέτοιων επιθέσεων επανάληψης σε περισσότερες από μια συνόδους του πρωτοκόλλου, μπορούν να οδηγήσουν σε μια *καθολική κατάσταση τερματισμού* (*fail-stop global state*), μια γενικά *μη αποδεκτή κατάσταση τερματισμού* ή μια *παραβίαση μιας πρότασης επιβεβαίωσης* του πρωτοκόλλου (*protocol correctness assertion)*. Οι τελευταίες πιθανές καταστάσεις αποκαλύπτουν μια άγνωστη επίθεση ασφαλείας τύπου *PARSES*.

4.4.7 Επίθεση Άρνησης Εξυπηρέτησης (Denial of Service, DoS)

Μια επίθεση άρνησης εξυπηρέτησης μπορεί να θεωρηθεί σαν την διενέργεια μιας σειρά από τακτικές επιθέσεων. Η επίθεση *DoS* μπορεί να λάβει χώρα οποιαδήποτε στιγμή μετά από την εκτέλεση μιας ενέργειας **send**(*v*, ag, *msg*), με το μήνυμα *msg* να αναπαριστά ένα μη κρυπτογραφημένο μήνυμα $msg \in Msgs$ ή μετά από την εκτέλεση μιας ενέργειας **send**(*v*, ag, $\{msg\}_k$) όπου $I \notin is\_key\_of\,(k) \wedge k^{-1} \in I_{knowledge}$. Οι προλεχθείς ενέργειες θα έχουν ως αποτέλεσμα την καθολική κατάσταση όπου:

$$exists(msg,\ sent^{v_j}_{\max(i)}) = \texttt{true} \text{ ή αντίστοιχα } exists(\{msg\}_k,\ sent^{v_j}_{\max(i)}) = \texttt{true}$$

για $1 \le j \le \#Ses_v$, με $i \ge 1$ αναπαριστώντας τους όρους από μια ακολουθία συναλύσωσης μηνυμάτων που έχουν υποκλαπεί. Στην εφαρμοζόμενη επίθεση *DoS* ο εισβολέας διαφθείρει ένα μήνυμα *msg* με βάση τη γνώση του $I_{knowledge}$ και χρησιμοποιεί αυτό το μήνυμα $msg' \in Msgs$ σε μια ενέργεια **send** (*I*, *v*, *msg*΄) ή αντίστοιχα **send** (*I*, *v*, $\{msg'\}_{k'}$) για κάποιο $k' \in I_{knowledge}$ έτσι ώστε $v \in is\_key\_of\,(k')$. Σε αυτό το σημείο ο εισβολέας επιχειρεί ουσιαστικά μια επίθεση παραβίασης της ακεραιότητας μηνύματος, στο υποκλεμμένο *msg* ή $\{msg\}_k$. Η συγκεκριμένη διαφθορά του μηνύματος μπορεί να διενεργηθεί είτε σε κρυπτογραφημένο ή μη κρυπτογραφημένο μήνυμα, καθώς ο εισβολέας έχει την ικανότητα να προσαρτεί *πλαστά δεδομένα*, *bg_data*, σε οποιοδήποτε μήνυμα επιθυμεί. Έτσι θα έχουμε αντίστοιχα: $msg'=msg \cdot bg\_data$ ή $\{msg\}_k' = \{msg\}_k \cdot bg\_data$.



Εικόνα 4.4.4 Επίθεση άρνησης εξυπηρέτησης DoS

Έχοντας τον εισβολέα σε θέση να αλλάζει τα υποκλεμμένα μηνύματα, διενεργεί την επίθεση παραβίασης της ακεραιότητας *N* φορές, αναπαριστώντας με αυτό τον τρόπο *N* διαφορετικές έντιμε αιτήσεις. Για να γίνει αυτό κάθε πλαστό δεδομένο *bg_data* που προσαρτεί ο εισβολέας δημιουργώντας μια

συναλύσωση (διεφθαρμένου) μηνύματος, πρέπει να είναι διαφορετικό για κάθε υποκλεμμένο μήνυμα. Σε συνδυασμό όλων των παραπάνω ο εισβολέας θα ακολουθεί τις παρακάτω ενέργειες:

exists(*msg*, $sent^{v_j}_{\max(i)}$)= `true`

do for N times

msg´=msg·bg_data(i), i=0..N;

bg_data'=bg_data(i) · bg_data(i+a), 0<a < N-i

send*(I, v, msg')*

end_do

Με σκοπό την επιτυχία μιας *DoS* επίθεσης, όπως φαίνεται στην εικόνα 4.4.4., από την πλευρά του αποδέκτη *v*, πρέπει να διενεργηθεί για κάθε μήνυμα *msg'* μια ενέργεια **receive***(v, I, msg´).* Έτσι για κάθε *msg'* θα έχουμε:

$$exists(msg', rcvd^{v_j}_{\max(i)}) = \texttt{true} \text{ ή } exists(\{msg'\}_{k'}, rcvd^{v_j}_{\max(i)}) = \texttt{true}$$

Η επίδραση αυτής της επίθεσης *DoS* [1][58] από την πλευρά των διαθέσιμων πόρων του αποδέκτη μπορεί να είναι είτε σε αποθέματα μνήμης (όταν το *N>>0*) ή στην υπολογιστική ισχύ που είναι αναγκαία για τον αποδέκτη *v* όπου λ.χ. μετά από την λήψη κάποιων $\{msg\}_k' = \{msg\}_k \cdot bg\_data$, εισέρχεται σε μια (υπερβολικού κόστους) διεργασία αποκρυπτογράφησης όλων των πλαστών-διεφθαρμένων μηνυμάτων.

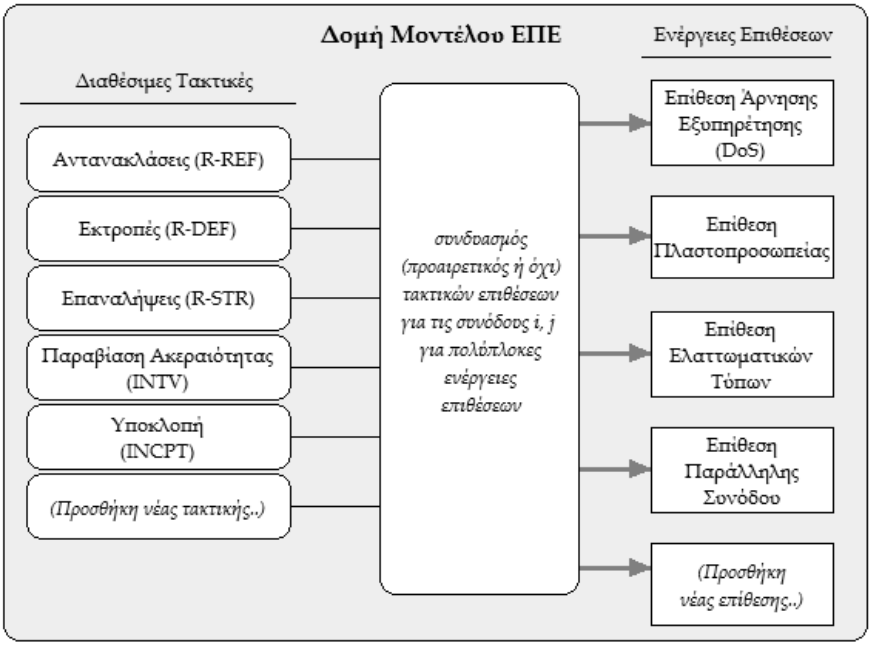
4.5 Επαλήθευση δυο πρωτοκόλλων μικροπληρωμών

Σε αυτή την ενότητα υλοποιείται ο εισβολέας ΕΠΕ που παρουσιάστηκε, και δοκιμάζεται στο περιβάλλον δύο μοντέλων πρωτοκόλλων ασφαλείας που χρησιμοποιούνται σήμερα για μικροπληρωμές, και προτάθηκαν από τους Rivest et al στο [81]. Για την σωστή λειτουργία του μοντέλου εισβολέα ΕΠΕ θα πρέπει το πρωτόκολλου που θέλουμε να επαληθεύσουμε να τερματίζει κατ' αρχήν σωστά με βάση τις προδιαγραφές του, και χωρίς την παρουσία του εισβολέα, σε πρώτη φάση. Ως εκ τούτου ορίζουμε τα παρακάτω τέσσερα (4) βήματα για τον σχεδιασμό και υλοποίηση του μοντέλου του πρωτοκόλλου και του εισβολέα ΕΠΕ:

- Καταγραφή των ιδιοτήτων ασφαλείας που θέλουμε να ελέγξουμε

- Επιλογή των τακτικών επιθέσεων που θέλουμε να περιέχει ο εισβολέας ΕΠΕ στη βάση του
- Επιλογή των επιθέσεων που στηρίζονται στις προηγούμενες τακτικές Υλοποίηση του πρωτοκόλλου (βήματα, μηνύματα, συμμετέχοντες) και του εισβολέα ΕΠΕ, με βάση τις προδιαγραφές του πρωτοκόλλου και της γλώσσας μοντελοποίησης του ελεγκτή μοντέλων
- Προσθήκη του εισβολέα ΕΠΕ ως ενδιάμεση οντότητα (Man-in-the-Middle)

Με αυτό τον τρόπο και βασιζόμενοι στην φορμαλιστική περιγραφή τόσο του εισβολέα όσο και των τακτικών επιθέσεων, προκύπτει ο εισβολέας ΕΠΕ που φαίνεται στην εικόνα 4.5.1. Στις επόμενες παραγράφους μοντελοποιούνται και παρουσιάζονται αποτελέσματα επαλήθευσης για τα πρωτόκολλα ασφαλών μικροπληρωμών PayWord και Micromint.



Εικόνα 4.5.1 Δομή και τακτικές επιθέσεων του εισβολέα ΕΠΕ

4.6 Το πρωτόκολλο ασφαλείας μικροπληρωμών PayWord

Το PayWord αποτελεί ένα ασφαλές πρωτόκολλο μικροπληρωμών που περιγράφηκε στο [81]. Πρόκειται για ένα πρωτόκολλο βασισμένο σε μονάδες πληρωμών (credit-based payment) και θεωρείται ως ένα εκτός-σύνδεσης πρωτόκολλο ασφαλών ηλεκτρονικών μικροπληρωμών για το διαδίκτυο. Υλοποιείται με την χρήση *αλυσίδων κατακερματισμού* (hash chains) οι οποίες

και αποκαλούνται *αλυσίδες του PayWord* (πίνακας 4.6.1). Στην συγκεκριμένη περίπτωση της συνάρτησης κατακερματισμού που χρησιμοποιείται, θα θεωρήσουμε ότι αυτή βασίζεται στην MD5 [80], η οποία έχει όρους που συμβολίζονται με `w(i)`. Σε μια σύνοδο του πρωτοκόλλου PayWord θα έχουμε τρεις συμμετέχουσες οντότητες, τον πελάτη *C*, (*Customer*), τον *B* (*Broker)* και την τράπεζα *V* (*Vendor)*. Ο customer *C* δημιουργεί μια σύνδεση με τον *Broker B*, ο οποίος εκδίδει ένα πιστοποιητικό (certificate) το οποίο και περιέχει πληροφορίες του *C* και την ταυτότητα του *B*. Το πιστοποιητικό αυτό θα εξουσιοδοτεί τον *C* να κατασκευάσει αλυσίδες PayWord, προσπαθώντας την αυθεντικοποίηση του εαυτού του, σε κάποιον (μετέπειτα) vendor *V*. Τα βασικά βήματα για την περάτωση της λειτουργίας του πρωτοκόλλου PayWord απεικονίζονται στην εικόνα 4.6.1. Με την λήψη του πιστοποιητικού, `certC`, ο *C* υπολογίζει την αλυσίδα του PayWord *w*, σε αντίστροφη σειρά, βασιζόμενος σε τυχαίους επιλεγμένους αριθμούς. Στη συνέχεια υπογράφει το commitment *M* του πρωτοκόλλου PayWord, το οποίο και αποτελείται από τον πρώτο όρο της υπολογισθείσας αλυσίδας `(w(0))` μαζί με τις απαραίτητες πληροφορίες του πελάτη *C* . Έπειτα το commitment M στέλνεται στον vector *V*.

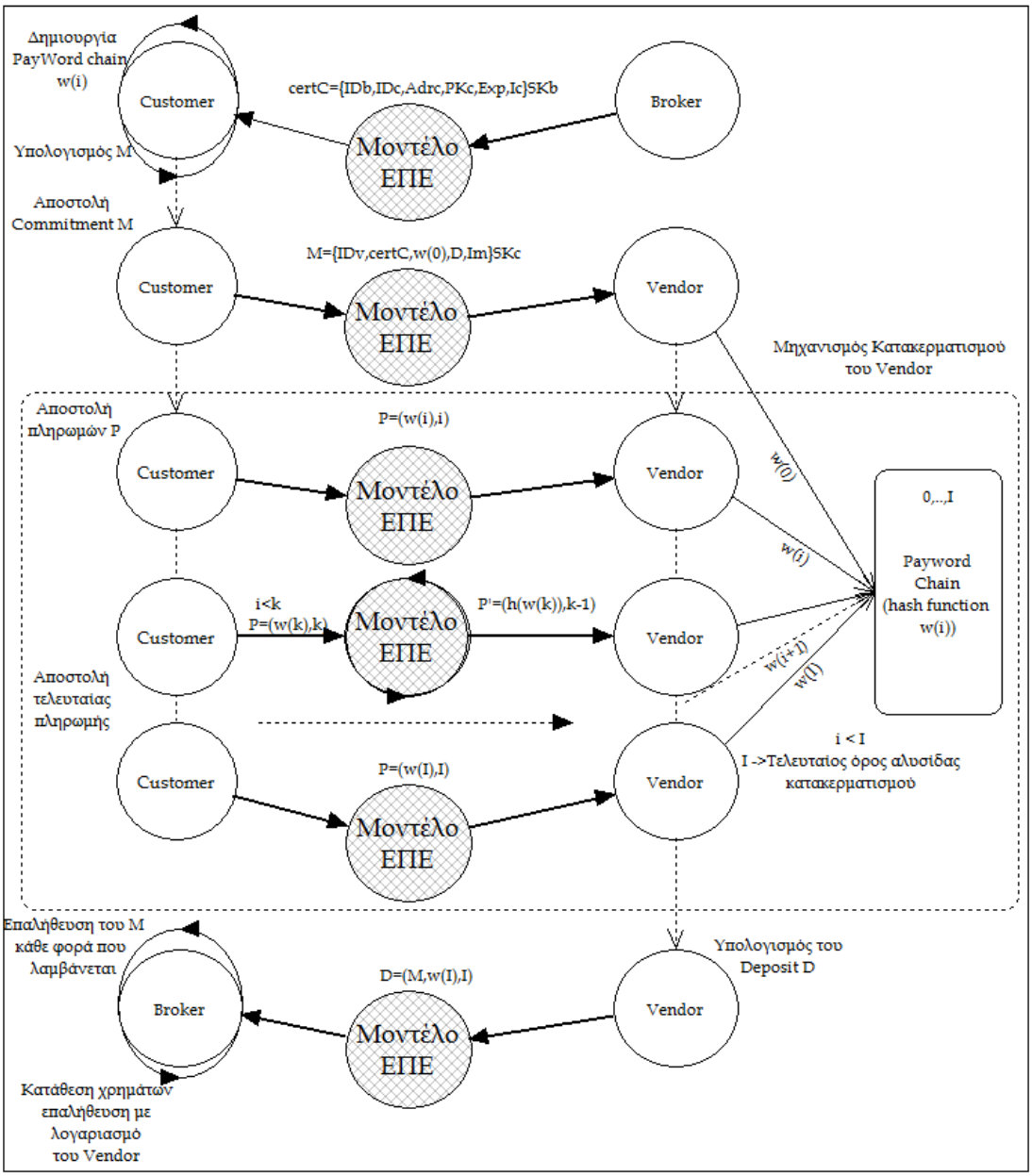
Σε κάθε μικροπληρωμή που αποπειράται να κάνει ο *C* ένας όρος της αλυσίδας του τύπου, *P*:`(w(i),i)` αποστέλλεται στον vector *V* μέχρι και την τελευταία πληρωμή, *P*:`(w(I),I)`. Στην συγκεκριμένη ανάλυση θεωρούμε το επιτιθέμενο σενάριο πληρωμής, όπου η τιμή της κάθε πληρωμής ποικίλει από το 1 μέχρι το *n*.

Πίνακας 4.6.1 Πίνακας συμβολισμών για το πρωτόκολλο PayWord

`IDc`	Customer ID	`Addrc`	Διεύθυνση του Customer
`IDb`	Broker ID	`certC`	Πιστοποιητικό του Customer
`IDv`	Vendor ID	`Exp`	Λήξη του πιστοποιητικού
`SKb`	Κλειδί του Broker's	`Ic`	Πληροφορίες του Customer
`PKc`	Δημόσιο Κλειδί του Customer	`Im`	Πληροφορίες του Vendor
`SKc`	Ιδιωτικό Κλειδί του Customer	`D`	Ημερομηνία

Ο vector *V* επαληθεύει όλες τις πληρωμές *P*, εφαρμόζοντας την επιλεχθείσα συνάρτηση κατακερματισμού *w* στις τελευταίες στην τελευταία ισχύουσα πληρωμή *v* φορές, όπου *v* είναι η τιμή της αιτούμενης πληρωμής `(w(i-v))`. Στο τέλος της ημέρας ,ο *V* αναφέρει στον *B* την τελευταία (υψηλότερη τιμή στον

πίνακα κατακερματισμού) πληρωμή `(w(I),I)` – όπου το `I=max(i)` – έχει ληφθεί από τον *C* μέσα στην τρέχουσα μέρα, μαζί με το commitment του *C*. Στην εικόνα 4.6.1 επίσης φαίνεται οι αλληλεπιδράσεις με όλα τα βήματα του πρωτοκόλλου PayWord και του εισβολέα ΕΠΕ, όπως μοντελοποιήθηκε στο περιβάλλον του ελεγκτή μοντέλων SPIN.

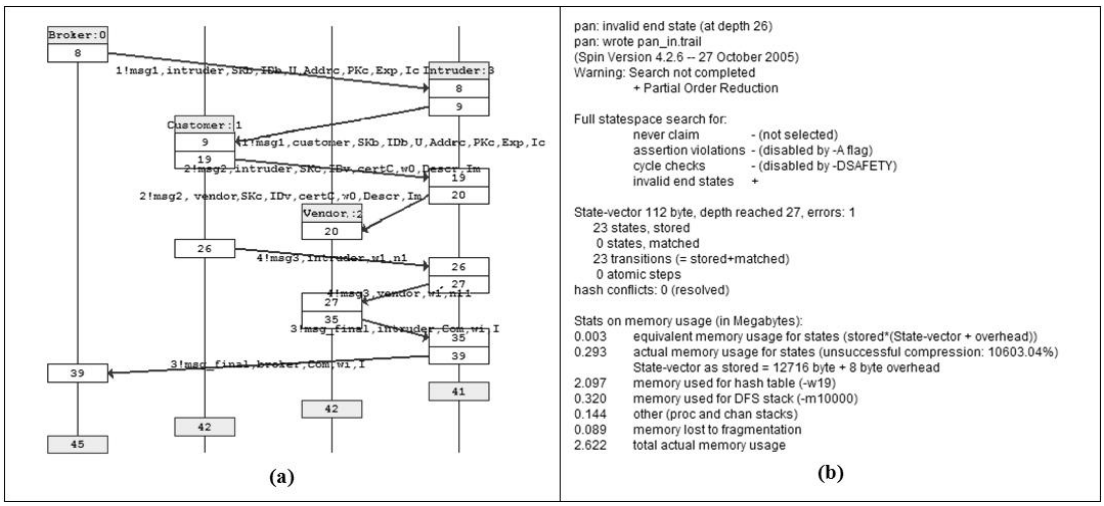


Εικόνα 4.6.1 Το μοντελοποιημένο σχήμα του πρωτοκόλλου PayWord με την αλληλεπίδραση του εισβολέα ΕΠΕ

Με την χρήση των αλυσίδων κατακερματισμού, ναι μεν διαβεβαιώνεται η μειωμένες απαιτήσεις υπολογιστικής ισχύος για τον vector *V*, αλλά από την άλλη επιτυγχάνει ο εισβολέας ΕΠΕ, βασιζόμενος στις ορισμένες τακτικές επιθέσεων

που κατέχει, να πετύχει μια επίθεση στον μηχανισμό του vector *V*, αποδεχόμενος εσφαλμένα ένα διεφθαρμένο υποκλεμμένο μήνυμα με στοιχεία όμοια της συνάρτησης κατακερματισμού που χρησιμοποιείται.

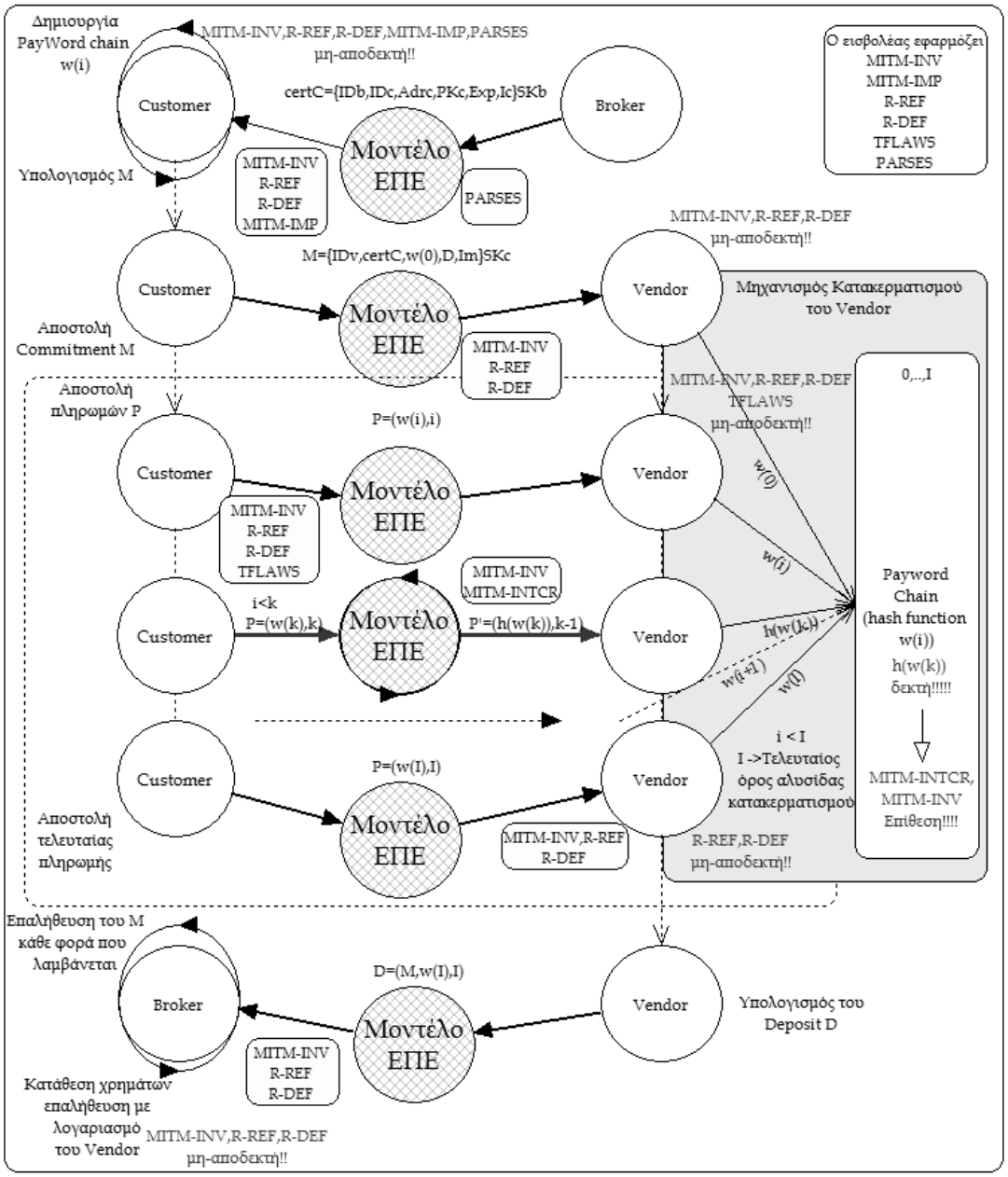
Με την προϋπόθεση ότι ο εισβολέας είναι γνώστης της συνάρτησης κατακερματισμού που χρησιμοποιείται (MD5 στην περίπτωση μας), ο εντοπισμός της επίθεσης αυτής λαμβάνει χώρα όταν ο εισβολέας υποκλέπτει ένα μήνυμα ανεξαρτήτου μεγέθους και χρονικής στιγμής και το διαφθείρει με την τακτική επίθεσης παραβίασης της ακεραιότητας (message integrity violation). Παρακάτω περιγράφεται η κατάσταση λάθους του παραγόμενου χώρου καταστάσεων, όπου ο εισβολέας επιτυγχάνει να προκαλέσει στο πρωτόκολλο PayWord (εικόνα 4.6.2α και 4.6.2β). Επίσης με την βοήθεια προσομοίωσης, αναπαριστάται γραφικά η επίθεση του εισβολέα ΕΠΕ στο πρωτόκολλο, καθώς και το σημείο όπου το πρωτόκολλο δέχεται την συγκεκριμένη επίθεση, και άρα επιβεβαιώνει το λάθος ασφαλείας του.



Εικόνα 4.6.2 (α) Επίθεση INTV για το μήνυμα πληρωμής (*P*): Ο vector *V* αποδέχεται το διεφθαρμένο μήνυμα του εισβολέα, (β) Η επίθεση INTV που αναφέρεται από την αναφορά επαλήθευσης του ελεγκτή μοντέλων SPIN

Στην κατάσταση 19, ο *C* αποστέλλει το commitment (*M*), το οποίο δεν μεταβάλλεται από τον εισβολέα ΕΠΕ, συνεχίζοντας με την πρώτη προσπάθεια πληρωμής (*P*). Στην κατάσταση 27 ο εισβολέας αλλάζει το μήνυμα `(w1,n1)` δημιουργώντας το διεφθαρμένο μήνυμα (`w1',n1-1`), το οποίο στην συγκεκριμένη χρονική στιγμή γίνεται αποδεκτό από τον vector *V*. Στο τελευταίο βήμα του PayWord, όπου ο vector *V* , αποστέλλει το μήνυμα *D* (deposit) και η σύνοδος του τελειώνει επιτυχώς, παραβιάζοντας με αυτό τον τρόπο την

ασφάλεια που προσφέρει. Μια τέτοια επίθεση, θα μπορούσε να οδηγήσει λανθασμένα τον broker B, στην διπλή χρέωση του πελάτη *C*.



Εικόνα 4.6.3 Η αποδεκτή παραβίαση ασφαλείας του πρωτοκόλλου PayWord με την αλληλεπίδραση του μοντέλου ΕΠΕ

Προς αποφυγή του συγκεκριμένου λάθους, μιας και βασίζεται στον μηχανισμό που διαθέτει ο vendor V να μην κρατάει άμεσα ιστορικό των μηνυμάτων που δέχεται και υπολογίζει κατά την επεξεργασία των αλυσίδων του PayWord, προτείνεται η χρήση διπλής παραμετρικής συνάρτησης κατακερματισμού (salted hash functions), αρχειοθετώντας με αυτόν τον τρόπο το κάθε μήνυμα (όρο της κατακερματισμένης αλυσίδας) σε έναν δεύτερο πίνακα.

Κάτι τέτοιο, και με την προϋπόθεση σύγκρισης των ληφθέντων μηνυμάτων, μπορεί και αποτρέπει την παραπάνω επίθεση του εισβολέα ΕΠΕ.

4.7 Το πρωτόκολλο μικροπληρωμών MicroMint

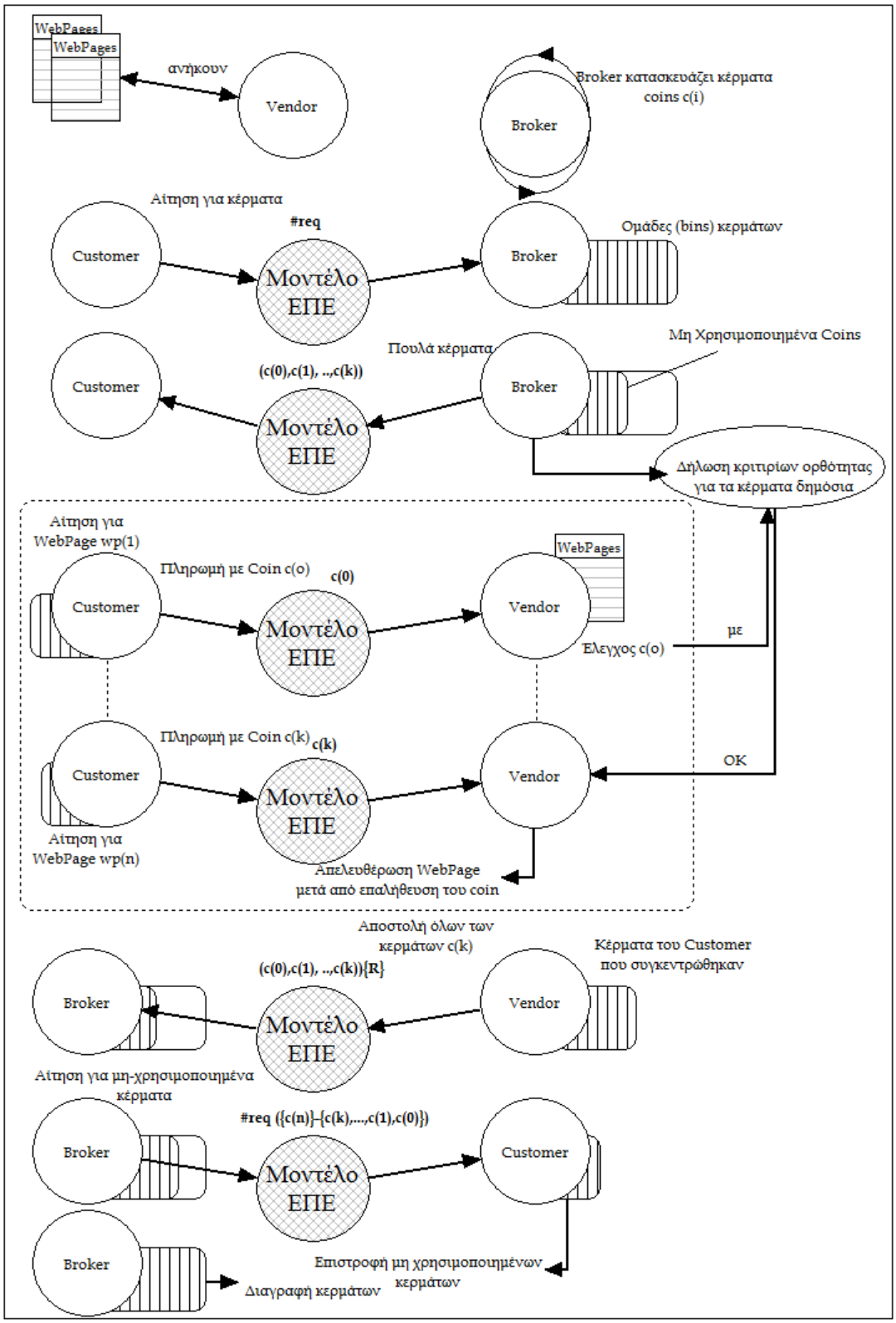
Το δεύτερο πρωτόκολλο ασφαλείας μικροπληρωμών που εξετάστηκε με το μοντέλο του εισβολέα ΕΠΕ, είναι το MicroMint, το οποίο και αυτό περιγράφθηκε στο [81]. Το MicroMint σχεδιάστηκε με τον σκοπό να είναι μικρό (σε απαιτήσεις υπολογιστικής ισχύος) και αποτελεσματικό χωρίς την ανάγκη χρήσης ασύμμετρης κρυπτογραφίας. Αυτό επιτυγχάνεται με την χρησιμοποίηση μηχανισμών αυθεντικοποίησης πέραν της κλασσικής κρυπτογραφίας, η οποία και είναι γνωστή για τις ανάγκες της σε υπολογιστική ισχύ.

Η βασική ιδέα του MicroMint βασίζεται στην διεργασία του Broker *Br*, να παράγει κέρματα, *coins,* πουλώντας τα στον πελάτη *Cm*, ο οποίος προηγουμένως έχει αιτηθεί αυτών. Ο πελάτης *Cm* μπορεί να χρησιμοποιήσει τα κέρματα αυτά όποτε θέλει αυτός να αγοράσει κάποιο προϊόν ή να χρησιμοποιήσει μια υπηρεσία επί πληρωμή. Σε απάντηση, ο κάθε vector *Ve* θα απαιτήσει, και εν τέλει, θα αποκτήσει την πρόταση της τράπεζας στον Broker *Br* εξαγοράζοντας τα κέρματα αυτά. Με αυτό τον τρόπο το MicroMint μπορεί να θεωρηθεί σαν ένα χρεωστικό (debit-based) σύστημα πληρωμής προσπαθώντας να παρέχει φυσιολογικές εγγυήσεις ασφαλείας σε χαμηλό κόστος, χωρίς πολύπλοκους και απαιτητικούς σε υπολογιστική ισχύ κρυπτογραφικές λειτουργίες. Τα κέρματα (*coins*) δημιουργούνται από τον επονομαζόμενο αλγόριθμο κατακερματισμένης συνάρτησης συγκρούσεων (Hash Function Collisions algorithm). Κάθε κέρμα αποτελεί μια συμβολοσειρά από δυφία, πεπερασμένου μεγέθους. Τα κέρματα αυτά έχουν την ιδιότητα να αποδίδουν την ίδια *εικόνα κατακερματισμού* υπό την εφαρμογή μιας ειδικά σχεδιασμένης συνάρτησης κατακερματισμού. Έτσι αναφέρονται ως *κέρματα κατακερματισμένης σύγκρουσης* (Hash Collided Coins). Κάθε κέρμα με την παραπάνω ιδιότητα, προάγει τον εύκολο έλεγχο ορθότητάς του και παράλληλα καθιστά δύσκολη την διαδικασία παραγωγής του από τρίτους. Ο πίνακας 4.7.1, παρουσιάζει τον απαραίτητο συμβολισμό για το πρωτόκολλο MicroMint που θα χρησιμοποιηθεί παρακάτω.

Πίνακας 4.7.1 Συμβολισμοί για το πρωτόκολλο MicroMint

`Cm`	Customer's ID
`Ve`	Vendor's ID
`Br`	Broker's ID
$x_1,\ldots,x_k$	k-φορές συγκρούσεις κατακερματισμού
`coins`	k-πλειάδα των $(x_1,\ldots,x_k)$
`Hashf`	Συνάρτηση κατακερματισμού
`Req_c`	Απαίτηση για κέρματα
`Req_wp`	Απαίτηση για ιστοσελίδα
`wp(n)`	Ιστοσελίδες διαθέσιμες στον Vendor *Vr*
`b(coins)`	Σύνολο από κέρματα (bins)
`exp_date`	Ημερομηνία λήξης των κερμάτων

Όπως γίνεται αντιληπτό, η ορθότητα των κερμάτων αποτελεί χαμηλής κατανάλωσης υπολογιστική διεργασία, βασιζόμενη στην προαποφασισμένη συνάρτηση κατακερματισμού μεταξύ των συμμετεχόντων του πρωτοκόλλου. Από την άλλη μεριά, και η συνάρτηση σύγκρουσης του κατακερματισμού, παρέχει μια επιπλέον ιδιότητα ασφαλείας. Αποτελεί πολύ δύσκολη διεργασία να μπορέσει κάποιος τρίτος να υπολογίσει την πρώτη σύγκρουση κατακερματισμού. Ειδικά μετά από το πέρας της πρώτης σύγκρουσης, η κλίμακα ων συγκρούσεων αυτών αυξάνεται εκθετικά. Αυτό σημαίνει ότι ένας broker *Br* του πρωτοκόλλου MicroMint έχει την δυνατότητα να παράγει έναν μεγάλο αριθμό από κέρματα με χαμηλό κόστος κατασκευής το καθένα. Με την τεχνική αυτή, με όσο το δυνατόν περισσότερα κέρματα μπορεί να παράγει και να πουλήσει ο broker *Br*, τόσο φθηνότερο θα είναι το κάθε κέρμα, φέρνοντας τον broker σε θέση να μεγιστοποιήσει το κέρδος του εάν πουλάει τα κέρματα σε μεγαλύτερο κόστος. Τα βασικά βήματα του MicroMint, με τις συμμετέχουσες οντότητες και τον εισβολέα ΕΠΕ, φαίνονται στην εικόνα 4.7.1. Η όλη διεργασία του πρωτοκόλλου περιγράφεται ως εξής: α) Ο Broker *Br* αρχικά διαθέτει την απαραίτητη υπολογιστική ισχύ για την παραγωγή των κερμάτων.



Εικόνα 4.7.1 Το πρωτόκολλο MicroMint με το μοντέλο ΕΠΕ

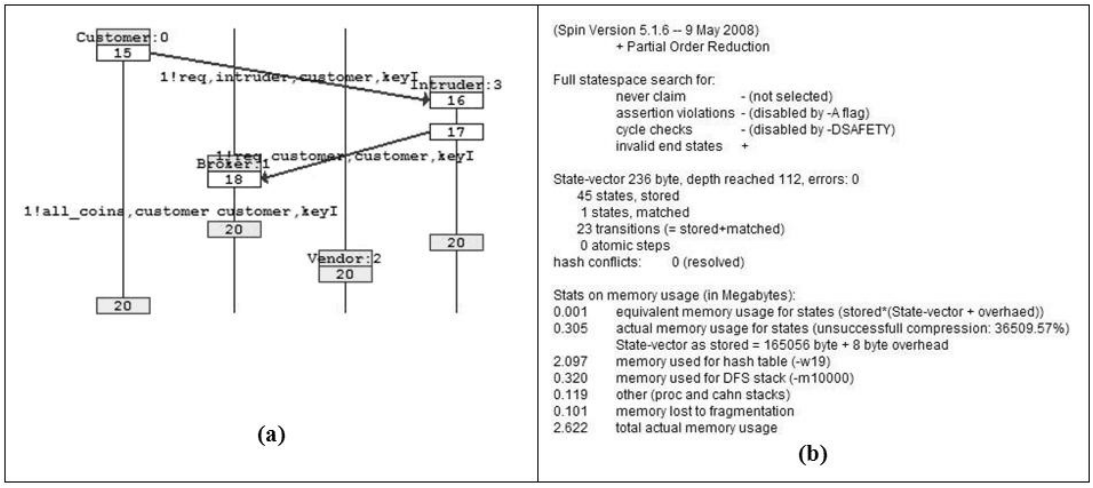
β) Στη συνέχεια πουλάει τα κέρματα αυτά στον πελάτη *Cm* οποίος και προηγουμένως έχει εκδώσει αίτηση για αυτά, με σκοπό την έναρξη αγορών από τον vector *Ve*. Για λόγους ασφαλείας, καινούργια κέρματα πρέπει να παραχθούν

με χρήση διαφορετικής συνάρτησης κατακερματισμού *Hashf* μετά από την πάροδο χρόνου *exp_date,* γ) όταν ο *Cm* αγοράζει (την πρόσβαση) μιας ιστοσελίδας, αιτείται της συγκεκριμένης σελίδας πληρώνοντας με ένα κέρμα. Ο vector *Ve* επαληθεύει το συγκεκριμένο κέρμα ελέγχοντας τα κριτήρια ορθότητας που ορίζει ο εκάστοτε broker *Br*. δ) Μετά από την παροχή επιβεβαίωσης από τον vendor, η αιτούμενη ιστοσελίδα ελευθερώνεται προς τον πελάτη *Cm*, ε) Στο τέλος της κάθε ημέρας ο vendor *Ve* επιστρέφει το σύνολο των κερμάτων στον broker *Br* για την απαραίτητη εξαγορά. στ) Στη λήξη του διαθέσιμου χρόνου λήξης του κέρματος *exp_date,* ο broker *Br* πρέπει να συλλέξει όλα εκείνα τα κέρματα από τον πελάτη *Cm* που δεν έχουν χρησιμοποιηθεί, αναθέτοντας σε οποιαδήποτε αίτηση και εάν λάβει, καινούργια κέρματα.

Μοντελοποιήθηκε το πρωτόκολλο MicroMint και όλες οι συμμετέχουσες οντότητες σε αυτό, μαζί με τον ίδια δομή του εισβολέα ΕΠΕ, όπως και στην περίπτωση του PayWord. Ο εισβολέας και σε αυτή την περίπτωση διαθέτει όλες τις προαναφερθείσες τακτικές επιθέσεων (αλλά και επιθέσεις ως ολότητες), παραμετροποιημένες όσον αφορά τις προδιαγραφές του πρωτοκόλλου MicroMint (για παράδειγμα το αναμενόμενο μήνυμα που πρόκειται να δεχθεί ένα κανάλι επικοινωνίας ορισμένο στη γλώσσα της PROMELA). Ο αναγνώστης μπορεί να ανατρέξει στο [70] για περαιτέρω τεχνικές πληροφορίες ως προς την κατασκευή του όλου μοντέλου.

Παρόλο την αποτελεσματικότητα του εισβολέα ΕΠΕ στο πρωτόκολλο PayWord, στην περίπτωση του MicroMint η εξαντλητική επαλήθευση του ελεγκτή μοντέλων SPIN, δεν εντόπισε κάποια κατάσταση λάθους. Οι εικόνες 4.7.2a και 4.7.2b παρουσιάζουν τα αποτελέσματα του προσομοίωσης και επαλήθευσης αντίστοιχα, για την διαλειτουργικότητα του MicroMint με τον εισβολέα ΕΠΕ. Κατά την διενέργεια μιας τυχαίας προσομοίωσης του μοντέλου, ο εισβολέας ΕΠΕ μπορεί να ανοίξει μόνος του μια σύνοδο του πρωτοκόλλου MicroMint, με την προϋπόθεση ότι κατέχει υποκλεμμένα μηνύματα από προηγούμενη σύνοδο του πρωτοκόλλου. Σκοπός του η διενέργεια μιας επίθεσης πλαστοπροσωπίας (Impersonation attack) της εναρκτήριας οντότητας του MicroMint. Στην συγκεκριμένη περίπτωση ο εισβολέας ΕΠΕ, όπως φαίνεται και στην εικόνα 4.7.2a, δέχεται ένα μήνυμα από τον πελάτη *C*, προωθώντας ένα διεφθαρμένο (Integrity violated) μήνυμα στον Broker *B*. Οι μηχανισμοί

επαλήθευσης όμως του Broker *B* θα απορρίψουν το ληφθέν μήνυμα προκαλώντας τον πρόωρο τερματισμό του πρωτοκόλλου. Στην εικόνα 4.7.2b βλέπουμε ότι δεν υπάρχει λάθος κατά την διάρκεια της επαλήθευσης.



Εικόνα 4.7.2 (a) Αποτελέσματα Προσομοίωσης του MicroMint με τον εισβολέα ΕΠΕ, (b) Αποτελέσματα επαλήθευσης του MicroMint

4.8 Συμπεράσματα κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκε και αναπτύχθηκε ο εισβολέας ΕΠΕ, ειδικός για την επαλήθευση πρωτοκόλλων ασφαλείας, στο περιβάλλον του αυτόματου ελεγκτή μοντέλων SPIN. Ο εισβολέας αποτελεί μια ανοικτή-προς-κλείσιμο βάση από επιθέσεις και τακτικές επιθέσεων, δίνοντας στον αναλυτή την ικανότητα να προσθέσει, να αφαιρέσει ή και να συνδυάσει επιθέσεις με σκοπό τον όσο πιο αποτελεσματικό εχθρικό έλεγχο του πρωτοκόλλου ασφαλείας. Μετά από την απαραίτητη περιγραφή των σχετικών ερευνητικών εργασιών, περιγράφθηκε φορμαλιστικά η δομή του εισβολέα, οι τακτικές επιθέσεων και οι πιο περίπλοκες επιθέσεις που υπάρχουν στην παρούσα έκδοσή του. Στη συνέχεια επιλέχθηκαν δύο πρωτόκολλα ασφαλείας μικροπληρωμών με σκοπό τον εξαντλητικό τους έλεγχο για τις ιδιότητες ασφαλείας που παρέχουν. Η υλοποίηση τόσο τον πρωτοκόλλων PayWord και MicroMint, όσο και του εισβολέα ΕΠΕ, πραγματοποιήθηκε στην PROMELA για το εργαλείο SPIN, και είναι διαθέσιμη στο [70]. Κατά την διάρκεια ανάλυσης των δύο πρωτοκόλλων, ο εισβολέας ΕΠΕ εντόπισε μια παραβίαση ασφαλείας για το πρωτόκολλο PayWord, μέσω της επίθεσης παραβίασης της ακεραιότητας του μηνύματος. Από την άλλη για το

πρωτόκολλο MicroMint, επιβεβαίωσε την ορθότητά του και την ασφάλεια των εγγυήσεων που παρέχει.

Παρόλο το εργαλείο που επιλέχθηκε για την συγκεκριμένη ανάλυση των δυο πρωτοκόλλων που παρουσιάστηκαν στις εργασίες [7][10], η φορμαλιστική περιγραφή του εισβολέα μπορεί να ακολουθηθεί για την μοντελοποίηση γενικών μοντέλων εισβολέων σε άλλα αυτοματοποιημένα εργαλεία ελέγχου μοντέλων, όπως το AVISPA, το Murφ ή το PRISM. Επίσης, όπως όλες οι προσεγγίσεις μοντελοποίησης πρωτοκόλλων που αναφέρονται στην βιβλιογραφία, έτσι και σε αυτή τα μοντέλα των πρωτοκόλλων περιορίζονται από την συνθήκη απουσίας της ολοκληρωσιμότητας του αυτόματου ελέγχου μοντέλων [62]. Η συγκεκριμένη προσέγγιση όμως δύναται να αποτελέσει ένα επιπρόσθετο μέσο-εργαλείο, το οποίο οδηγεί σε αποτελεσματικό έλεγχο μοντέλων πρωτοκόλλων ασφαλείας, ικανό να αποκαλύψει παραβιάσεις ασφαλείας βασισμένες σε γνωστές τακτικές επιθέσεων. Με τον συνδυασμό των επιθέσεων αυτών, ο αναλυτής έχει την ικανότητα να συνθέσει επιθέσεις πέρα από το εύρος εκείνων των επιθέσεων που μπορούν να ελέγξουν οι υπάρχοντες ελεγκτές μοντέλων. Επιπλέον, το μοντέλο του εισβολέα δεν θέτει περιορισμούς στον εκάστοτε ελεγκτή μοντέλων που χρησιμοποιείται, μιας και δεν αποτρέπει τον έλεγχο για γενικές παραβιάσεις ιδιοτήτων ασφάλειας, βιωσιμότητας ή δικαιοσύνης του πρωτοκόλλου. Αντίθετα, δίνεται ώθηση με την συμμετοχή μιας δυνατής κακόβουλης οντότητας να προκαλέσει (εάν μπορεί) καταστάσεις επίθεσης στο πρωτόκολλο, που θεωρούνται ως καταστάσεις παραβίασης της ασφάλειας. Τέλος, αξίζει να σημειωθεί ότι η δομή του εισβολέα ΕΠΕ, είναι τέτοια που επιτρέπει την επέκταση της βάσης των επιθέσεών του, με περισσότερες τακτικές ή και πιο πολύπλοκες επιθέσεις, με στόχο έλεγχο ιδιοτήτων ασφαλείας όπως μη-αποποίηση της ευθύνης (nor-repudiation) ή ανωνυμία (anonymity) των συμμετεχόντων οντοτήτων.

Κεφάλαιο 5ο

Το Μοντέλο Εισβολέα Διερεύνησης Μηνύματος

5.1 Εισαγωγή

Όπως αναφέρθηκε και στα προηγούμενα κεφάλαια, κάθε απόπειρα απαρίθμησης όλων των πιθανών μηνυμάτων που μπορεί να παράγει ένας εισβολέας (τύπου DY), συνδυασμένα με έναν αριθμό ενεργειών κατά τον έλεγχο πρωτοκόλλων ασφαλείας σε όλα του τα επιμέρους βήματα, οδηγεί σε έναν τεράστιο χώρο καταστάσεων, δύσκολος να αναλυθεί. Σε αυτό το κεφάλαιο θα παρουσιαστεί ένα καινούργιο μοντέλο εισβολέα, το οποίο και μπορεί να χρησιμοποιηθεί για τον αποτελεσματικό έλεγχο του χώρου των καταστάσεων ενός μοντέλου πρωτοκόλλου ασφαλείας , αλλά πιο ειδικότερα, για το κλάδεμα του παραγόμενου δένδρου, με σκοπό την πιο εύκολη και γρήγορη ανάλυση του χώρου των καταστάσεων. Η τεχνική αυτή του **επονομαζόμενου εισβολέα διερεύνησης μηνύματος, ΕΔΜ**, (*MI, message inspection*) μπορεί να χρησιμοποιηθεί ταυτόχρονα με τεχνικές όπως αυτή της μερικής ταξινομημένη μείωσης (partial order reduction) [30] ή της συμμετρικής μείωσης (symmetry reduction). Ο συγκεκριμένος εισβολέας βασίζεται στην χειραγώγηση της γνώσης που αποκτά ένας εισβολέας κατά την αλληλεπίδρασή του με οντότητες του υπό εξέταση πρωτοκόλλου ασφαλείας, προσθέτοντας ειδικά μεταδεδομένα για κάθε

υποκλεμμένο μήνυμα. Σε μια προκαταρκτική εκτέλεση προσομοίωσης του μοντέλου με τον εισβολέα ΕΔΜ, ο εισβολέας προσαρτεί ετικέτες (μεταδεδομένων) στα μηνύματα τα οποία υποκλέπτει, συγκεκριμένες τιμές οι οποίες βασίζονται στις παραμέτρους λειτουργίας του εκάστοτε πρωτοκόλλου. Τα μεταδεδομένα αυτά χρησιμοποιούνται για την αναγνώριση πιθανόν επιθέσεων, για τι οποίες είναι από πριν γνωστό ότι *δεν μπορούν να προκαλέσουν παραβίαση ασφαλείας*. Ο αλγόριθμος του εισβολέα ΕΔΜ, επιλέγει με βάση τα μεταδεδομένα, ποιες επιθέσεις που έχει διαθέσιμες στην βάση του, μπορούν να ακυρωθούν και να προταθεί η αφαίρεσή τους από τον αναλυτή. Έτσι ο έλεγχος μοντέλων εστιάζεται σε έναν εισβολέα ο οποίος εξαπολύει μόνον επιθέσεις που μπορούν να ουσιαστικά να προκαλέσουν προβλήματα ασφαλείας στο πρωτόκολλο. Το πιο ενδιαφέρον χαρακτηριστικό του εισβολέα ΕΔΜ, κατά την λειτουργία του, είναι τόσο η ικανότητά του να ελέγχει το πρωτόκολλο σε μια σειρά από επιθέσεις που εξ' αρχής ορίζονται από τον αναλυτή, όσο και η αποφυγή του φαινομένου ΕΧΚ, αφού περιορίζονται οι ενέργειες του εισβολέα, σε αυτές που πραγματικά χρειάζονται. Σε σύγκριση με τον γνωστό εισβολέα DY, ο εισβολέας ΕΔΜ επιτυγχάνει πολύ μικρότερο χώρο καταστάσεων, διευκολύνοντας έτσι τόσο την ανάλυσή του, όσο και την αποτελεσματικότητά του. Ο συγκεκριμένος εισβολέα εφαρμόστηκε πάνω στο γνωστό πρωτόκολλο ασφαλείας ασύμμετρης κρυπτογράφησης των Needham και Schroeder (NSPK) [74], όπου και επιτυχώς κατάφερε να βρει την παραβίαση ασφαλείας που υπάρχει, γρηγορότερα από τον εισβολέα DY.

5.2 Γενική περιγραφή του προβλήματος

Η ανάλυση των τρεχόντων κρυπτογραφικών πρωτοκόλλων, έχει δείξει ότι ακόμα και με την χρησιμοποίηση τέλειων κρυπτογραφικών μηχανισμών (όπου έχουν αποδειχθεί ασφαλής), το πρωτόκολλο μπορεί να υποπέσει σε λάθη, προκαλώντας παραβιάσεις ασφαλείας προς τις οντότητές του. Στην σχετική βιβλιογραφία [32][74][81], υπάρχουν παραδείγματα πρωτοκόλλων ασφαλείας τα οποία έχουν δημοσιευθεί με λάθη, τα οποία παρέμειναν άγνωστα για αρκετά χρόνια. Το γεγονός αυτό έρχεται να ενισχύσει την άποψη για απαραίτητη τυπική ανάλυση και εξαντλητικό έλεγχο των πρωτοκόλλων αυτών [45], με σκοπό την

επαλήθευση των ιδιοτήτων ασφαλείας που προσφέρει. Όπως αναφέρθηκε και σε προηγούμενα κεφάλαια, ο αυτόματος έλεγχος μοντέλων, αποτελεί μια ευρέως διαδεδομένη τεχνική η οποία έχει αποδείξει την αποτελεσματικότητά της με την ανακάλυψη αγνώστων λαθών ασφαλείας, σε πρωτόκολλα τα οποία υποτίθεται ότι πληρούσαν τις προϋποθέσεις τους. Παρόλα αυτά όμως το φαινόμενα της έκρηξης του χώρου των καταστάσεων συνεχίζει να εμποδίζει την ανάλυσή τους, ειδικότερα όταν το πρωτόκολλο βασίζεται σε πολύπλοκους μηχανισμούς αυθεντικοποίησης και παραμέτρους λειτουργίας.

Σε γενικούς ελεγκτές μοντέλων [31][29], η έκρηξη του χώρου των καταστάσεων εισέρχεται ως ένα φαινόμενο που οφείλεται στην ασύγχρονη υλοποίηση παράλληλων προς εκτέλεση διεργασιών. Ειδικότερα στην χρήση των ελεγκτών μοντέλων για την επαλήθευση ιδιοτήτων ασφαλείας, ο εισβολέας που χρησιμοποιείται αποτελεί τον κύριο παράγοντα μεγιστοποίησης του χώρου των καταστάσεων. Ο έλεγχος μοντέλων για τις ιδιότητες ασφαλείας μυστικότητας (secrecy) ή αυθεντικοποίησης (authentication) βασίζεται κυρίως στις υποθέσεις που λαμβάνονται εξ' αρχής για την κυριαρχία του εισβολέα πάνω στο επικοινωνιακό μέσο των συμμετεχόντων στο πρωτόκολλο. Οι υποθέσεις αυτές βασίζονται κυρίως στις προδιαγραφές λειτουργίας για τις οποίες υλοποιήθηκε το πρωτόκολλο, προσπαθώντας με αυτό τον τρόπο να καταγραφούν όλες οι τυχόν παραβιάσεις ασφαλείας που μπορεί να προκαλέσει ένας κακόβουλος χρήστης. Τέτοιες υποθέσεις στον έλεγχο μοντέλων των πρωτοκόλλων ασφαλείας αναπαρίστανται από τον γενικό μοντέλο εισβολέα των DY [34] όπου: ο εισβολέας μπορεί να υποκλέψει όλα τα μηνύματα που ανταλλάσσονται μεταξύ των οντοτήτων του πρωτοκόλλου, αντικαθιστώντας τα με μηνύματα τα οποία , βάση ορισμένων κανόνων-ενεργειών, μπορεί να παράγει ο ίδιος, βασιζόμενος στην αρχική γνώση του. Τα μηνύματα αυτά μπορούν να αποσταλούν σε όλες τις οντότητες, σε διαφορετικές ή ίδιες συνόδους του πρωτοκόλλου. Τα νέα μηνύματα που μπορεί να κατασκευάσει ένας εισβολέας DY μπορεί να βασίζονται στις επόμενες τέσσερις (4) βασικές ενέργειες: a) *κρυπτογράφηση (encryption),* b) *αποκρυπτογράφηση (decryption),* c) *συναλύσωση (concatenation),* δ) *προβολή (projection)*. Επίσης ο εισβολέας DY περιλαμβάνει επιπρόσθετες αρχικές προϋποθέσεις για τον περιορισμό της δύναμής του, όπως την μη δυνατή αποκρυπτογράφηση ενός μηνύματος, εάν

προηγουμένως δεν κατέχει το σωστό κλειδί, καθώς και την ικανότητα του εισβολέα αν αποτρέψει ένα μήνυμα να φτάσει στον αρχικό προορισμό του.

Βάση των παραπάνω υποθέσεων (που αρχικά έχουν οριστεί στο [34]), οποιαδήποτε απόπειρα απαρίθμησης όλων των πιθανών επιθέσεων και ενεργειών σε όλα τα βήματα ενός πρωτοκόλλου ασφαλείας, περιλαμβάνοντας την αλληλεπίδραση με όλες τις οντότητες σε πεπερασμένο αριθμό συνόδων του πρωτοκόλλου, οδηγεί αναπόφευκτα στην έκρηξη του χώρου των καταστάσεων. Δεδομένου της ικανότητας του εισβολέα DY να προσπαθεί να συνθέσει κλειδιά (μέσω της ενέργειας συναλύσωσης), δοκιμάζοντας την αποτελεσματικότητά τους στα υποκλεμμένα μηνύματα, ο εισβολέας μπορεί να επέλθει σε μια άπειρη αναδρομική εκτέλεση ενεργειών δημιουργίας καινούργιων διεφθαρμένων μηνυμάτων (επίσης άπειρων σε αριθμό), με ολέθριες συνέπειες για τον αναλυτή.

Στην περίπτωση του ελέγχου μοντέλων σήμερα, ο αναλυτής τείνει να οριοθετεί τον αριθμό των διεφθαρμένων μηνυμάτων που μπορεί να παράγει ένας εισβολέας τύπου DY, με σκοπό το πεπερασμένο όριο του χώρου των καταστάσεων του μοντέλου. Παρόλα αυτά όμως, η αναγκαία πολυπλοκότητα για τον εισβολέα, αναγκάζει τον ελεγκτή μοντέλων να αποθηκεύει όλες τις πληροφορίες για κάθε κατάσταση που παράγεται στο υπό εξέταση πρωτόκολλο, που έχει να κάνει με την τρέχουσα γνώση όλων των οντοτήτων-συμπεριλαμβανομένου και του εισβολέα-, καταναλώνοντας τους διαθέσιμους πόρους (υπολογιστικούς ή μνήμης) που αποδίδονται προς την ανάλυση του μοντέλου.

Στην σχετική βιβλιογραφία έχουν προταθεί τεχνικές για την μείωση του παραγόμενου χώρου καταστάσεων, όπως αυτή της συμμετρικής μείωσης, όπως αναφέρθηκε και στο 2ο κεφάλαιο. Για τα πρωτόκολλα ασφαλείας, μια συμμετρική μείωση των καταστάσεων είναι και η επόμενη περίπτωση: Έστω δυο συμμετέχουσες στο πρωτόκολλο οντότητες *A* και *B* οι οποίοι λειτουργού ως εναρκτήριες οντότητες για διαφορετική όμως σύνοδο του πρωτοκόλλου. Η κατάσταση στην οποία ο *A* έχει εκκινήσει την δικιά του σύνοδο πρωτοκόλλου και ο *B* παραμένει στάσιμος, παρουσιάζει συμμετρία με την κατάσταση όπου ο *B* έχει εκκινήσει την δικιά του σύνοδο και ο *A* παραμένει στάσιμος από την πλευρά του. Η τεχνική αυτή της συμμετρικής μείωσης, χωρίζει τον παραγόμενο χώρο των καταστάσεων σε ποικίλες τάξεις ομάδων, οι οποίες και εξερευνούν

ξεχωριστά για κάθε κατάσταση του μοντέλου. Η πρώτη απόπειρα υλοποίησης της τεχνικής αυτής για την επαλήθευση πρωτοκόλλων ασφαλείας, υλοποιήθηκε στο εργαλείο BRUTUS [28]. Στο [30], οι ίδιοι συγγραφείς περιγράφουν τις περιπλοκές και τα προβλήματα που εμφανίζονται με την χρήση της τεχνικής μερικής ταξινομημένης μείωσης, εξαιτίας της πολύπλοκης φύσης της γνώσης του εισβολέα. Η τεχνική αυτή, αποφεύγει την δημιουργία καταστάσεων οι οποίες δεν επηρεάζονται από την διαρκεί εκτέλεση όλων των παράλληλων διεργασιών του μοντέλου. Αποτελέσματα από πειράματα ελέγχου μοντέλων με την τεχνική αυτή μείωσαν τον χώρο των καταστάσεων σε μεγάλο βαθμό ειδικότερα σε περιπτώσεις όπου είχαμε αλληλεπίδραση πολλών οντοτήτων σε πολλές συνόδους πρωτοκόλλων. Άλλες τεχνικές [39][90] εξερευνούν συγκεκριμένες ιδιότητες οι οποίες αναγνωρίζονται ως χαρακτηριστικά των υπό εξέταση πρωτοκόλλων ασφαλείας. Επίσης στο [88] οι συγγραφείς προτείνουν μια τεχνική η οποία βασίζεται σε μια προσέγγιση *διαίρει και βασίλευε*, για την μείωση της αναγκαίας μνήμης προς την διεξαγωγή της ανάλυσης.

Ένα μεγάλο μέρος ερευνητικών προσπαθειών για την αποφυγή του φαινομένου ΕΧΚ υιοθετεί την συμβολική αναπαράσταση του χώρου των καταστάσεων, με σκοπό την μη-αναγκαία απαρίθμηση όλων των παραγομένων μηνυμάτων του εισβολέα. Γενικά, όμως όλες οι τεχνικές περιορίζουν τον αναλυτή στην χρησιμοποίηση του εργαλείου εκείνου στο οποίο υλοποιήθηκε η τεχνική. Παρόλα αυτά όμως, γίνεται αποδεκτό από τους αναλυτές, ότι κάθε προτεινόμενη τεχνική, βοηθά στην επίλυση του φαινομένου της ΕΧΚ και άρα προς την εύκολη ανάλυση ενός μέρους των μοντέλων πρωτοκόλλων ασφαλείας. Ο συνδυασμός αυτών, αντίθετα μπορεί να επιτρέψει την εκμετάλλευση των πλεονεκτημάτων, που προσφέρουν οι παραπάνω προσεγγίσεις για τους την μείωση του φόρτου μιας συνολικής εργασίας ελέγχου μοντέλων, για τον αναλυτή.

Στο παρόν κεφάλαιο, δίνεται η περιγραφή του μοντέλου εισβολέα διερεύνησης μηνύματος (Message Inspection Intruder), ο οποίος αποτελεί έναν περιορισμένου τύπου ενδιαμέσου-οντοτήτων DY εισβολέα, βασισμένος στην ιδέα της βελτίωσης της ποιότητας της γνώσης του εισβολέα με συγκεκριμένα μεταδεδομένα μηνυμάτων. Η προτεινόμενη προσέγγιση, δίνει την δυνατότητα στον αναλυτή να καθοδηγήσει ο ίδιος στην αναζήτηση στον υπό έλεγχο

παραγόμενο χώρο καταστάσεων, αφού ο εισβολέας αποφεύγει να εκτελέσει επιθέσεις που (από τον αλγόριθμο του, της διερεύνησης του μηνύματος) γνωρίζει ότι δεν πρόκειται να πετύχουν. Αντίθετα, θα δυσκολέψουν την όλη διαδικασία της ανάλυσης, εκτινάσσοντας των χώρο των καταστάσεων σε μεγάλα μεγέθη. Η περιγραφόμενη δύο-σταδίων προσέγγιση δεν περιορίζει την γενική αποτελεσματικότητα και επίδραση του ελεγκτή μοντέλων, μιας και το εργαλείο συνεχίζει να αναζητεί τον χώρο των καταστάσεων, πέρα από τις επιθέσεις ασφαλείας που προκαλεί ο εισβολέας (όπως αδιέξοδα, μη-εκτελέσιμος κώδικας).

Στις επόμενες παραγράφους του κεφαλαίου αυτού θα παρουσιαστεί η βασική θεωρία και ο συμβολισμούς που δημιουργήθηκε, απαραίτητος για την περιγραφή του εισβολέα ΕΔΜ. Παρουσιάζεται η δομή του εισβολέα, ο αλγόριθμος στον οποίο βασίζεται καθώς και οι κανόνες απόφασης που διέπουν την όλη προσέγγιση διερεύνησης μηνύματος. Στην συνέχεια υλοποιείται ο εισβολέας ΕΔΜ μαζί με τον εισβολέα DY, και εφαρμόζονται ξεχωριστά στο πρωτόκολλο ασφαλείας NSPK. Παρέχεται μια ανάλυση ευαισθησίας της όλης προσέγγισης, καθοδηγώντας λεπτομερώς των αναλυτή που επιθυμεί να χρησιμοποιήσει την συγκεκριμένη προσέγγιση. Στα αποτελέσματα που παίρνουμε από τον ελεγκτή μοντέλων SPIN, παρουσιάζεται η μεγάλη διαφορά στον χώρο των καταστάσεων που παράγεται μέσω του εισβολέα ΕΔΜ σε σύγκριση με τον εισβολέα DY. Επίσης, ο εισβολέας ΕΔΜ καταφέρνει και εντοπίζει σε μικρότερο βάθος (του χώρου καταστάσεων) το λάθος που υπάρχει στο πρωτόκολλο NSPK από ότι στο βάθος που το εντοπίζει ο εισβολέας DY. Επιπρόσθετα, η προτεινόμενη τεχνική επιτρέπει την χρήση της σε συνδυασμό με άλλες τεχνικές μείωσης του χώρου των καταστάσεων, μειώνοντας ακόμα περισσότερο των παραγόμενο χώρο. Τέλος αποτελέσματα παρουσιάζονται και από την παραμετροποίηση του εισβολέα ΕΔΜ, ολοκληρώνοντας με αυτό τον τρόπο την θετική συνεισφορά του στο συγκεκριμένο πρόβλημα. Τέλος, παρουσιάζονται παρόμοιες ερευνητικές προσπάθειες στο χώρο, συγκρίνονται διεξοδικά με την προτεινόμενη προσέγγιση του εισβολέα ΕΔΜ. Επίσης στον επίλογο του κεφαλαίου συζητούνται μελλοντικές επεκτάσεις του εισβολέα καθώς και πλεονεκτήματα ή μειονεκτήματα χρήσης του.

5.3 Θεωρία του Εισβολέα Διερεύνησης Μηνύματος

Και σε αυτό το κεφάλαιο είναι αναγκαίο να ορίσουμε τους συμβολισμούς που χρησιμοποιούνται για την περιγραφή του εισβολέα ΕΔΜ. Περιγράφεται οι χρησιμοποιούμενοι όροι που θα χρησιμοποιηθούν και οι οποίοι θεωρούνται επέκταση του συμβολισμού του εισβολέα ΕΠΕ, όπως παρουσιάστηκε στο κεφάλαιο 4.

Ένα ατομικό μήνυμα (atomic message) μπορεί να προέρχεται από τα παρακάτω σύνολα:

- *Κλειδιά* (Keys), με όρους που αναπαριστούν τα κλειδιά που χρησιμοποιούνται για την κρυπτογράφηση των μηνυμάτων, έτσι ώστε κάθε κλειδί $k \in Keys$ θα έχει και ένα αντίστροφο κλειδί $k^{-1} \in Keys$, στην ασύμμετρη κρυπτογραφία. Για τις περιπτώσεις τις συμμετρικής κρυπτογραφίας το κλειδί της κρυπτογράφησης και αποκρυπτογράφησης είναι το ίδιο, όπου $k = k^{-1}$
- *Πράκτορες (Agents)*, το οποίο αποτελεί ένα σύνολο από τα ονόματα (ταυτότητες) των έντιμων συμμετεχόντων στο πρωτόκολλο.
- *Τυχαίοι Αριθμοί (Nonces)*, οι αποτελούν ένα μη πεπερασμένο σύνολο από τυχαίους αριθμούς. Τα μέλη του συνόλου αυτού χρησιμοποιούνται ως *χρονοσφραγίδες* (timestaps), όπου επισυνάπτονται στα μηνύματα του πρωτοκόλλου με σκοπό την ένδειξη της χρονική δημιουργίας του κάθε μηνύματος.
- *Δεδομένα (Data)*, τα οποία αποτελούν ένα σύνολο από συμβολοσειρές (strings), που ανταλλάσσονται μεταξύ των συμμετεχόντων στο πρωτόκολλο. Από την πλευρά του εισβολέα, δεδομένα μπορούν να παραχθούν χωρίς όμως απαραίτητα αυτά, να έχουν κάποιο νόημα.

Θα χρησιμοποιήσουμε τον συμβολισμό I ,*Intruder*, για να αναφερθούμε στον εισβολέα, όπου $I \notin Agents$. Επίσης ορίζουμε την δυαδική σχέση: $is\_key\_of = \{(k, id): k \in Keys, id \in Agents \cup \{I\}, \text{“}key\ k\ is\ used\ by\ the\ participant\ id\text{”}\}$, έτσι ώστε $|is\_key\_of\ (k)| = 1$ στην περίπτωση της ασύμμετρης ή $|is\_key\_of\ (k)| = 2$ στην περίπτωση της συμμετρικής κρυπτογράφησης, αντίστοιχα. Το σύνολο *Msgs* των ανταλλασσόμενων μηνυμάτων ορίζεται επαγωγικά πάνω στην ασύνδετη σχέση $AMsgs = Keys \cup Agents \cup \{I\} \cup Nonces \cup Data$,η οποία αναπαριστά το σύνολο

των ατομικών μηνυμάτων ($Set_i \cap Set_j = \varnothing$ για κάθε δύο Set_i, Set_j από τα ενοποιημένα σύνολα). Θα έχουμε:

- Εάν το $\alpha \in AMsgs$ τότε $\alpha \in Msgs$.
- Εάν $msg_x \in Msgs$ και $msg_y \in Msgs$ τότε $msg_x \cdot msg_y \in Msgs$, όπου η $\cdot$ αναπαριστά την συναλύσωση μηνυμάτων (concatenation)
- Εάν $msg \in Msgs$ και $k \in Keys$ τότε $\{msg\}_k \in Msgs$.

Κάθε ag $\in$ *Agents* μπορεί να επιχειρήσει να εκτελέσει το πρωτόκολλο για ένα περιορισμένο αριθμό συνόδων, έστω $\#Ses_{\text{ag}}$, όπου κάθε τέτοια απόπειρα αποτελεί μια ξεχωριστή σύνοδο πρωτοκόλλου *noSes*, όπου $1 \leq noSes \leq \#Ses_{\text{ag}}$. Σε μια σύνοδο πρωτοκόλλου, ο πράκτορας ag παίζει είτε το ρόλο της εναρκτήριας οντότητας (Initiator) είτε της οντότητας του ανταποκριτή (Responder). Ορίζουμε ως $sent_n^{\text{ag}_{noSes}}$ την πεπερασμένη μήκους συναλύσωσης μηνυμάτων που αποστάλθηκαν από τον ag $\in$ *Agents* κατά την διάρκεια της συνόδου *noSes*:

$$sent_n^{\text{ag}_{noSes}} = (sent_{n-1}^{\text{ag}_{noSes}} \cdot msg_n),$$

με τον πρώτο όρο να αποτελεί την *κενή ακολουθία* (null sequence), η οποία θα είναι $sent_0^{\text{ag}_{noSes}} = ()$. Η ακολουθία $sent_n^{\text{ag}_{noSes}}$ αναπαριστά το ιστορικό του πράκτορα ag (history) για την σύνοδο *noSes*, μετά την αποστολή του msg_n. Ορίζουμε ως $rcvd_n^{\text{ag}_{noSes}}$ την πεπερασμένη μήκους συναλύσωσης μηνυμάτων που ελήφθησαν από τον ag $\in$ *Agents* κατά την διάρκεια της συνόδου *noSes*. Σε ένα οποιοδήποτε χρονικό στιγμιότυπο, ορίζεται ως *γνώση του συμμετέχοντα* (*participant's knowledge*) για την συγκεκριμένη σύνοδο του πρωτοκόλλου ως:

$$\text{ag}_{\text{knowledge}} = \bigcup_{\text{ag}_j} \{rcvd_{\max(i)}^{ag_j}\} \cup \text{ag}_{in\_knowledge},$$

για κάθε $1 \leq j \leq \#Ses_{\text{ag}}$, όπου $\text{ag}_{in\_knowledge}$ αναπαριστά την *αρχική βάση γνώσης* του ag (κλειδιά, πράκτορες, ταυτότητες κα) και $i > 0$ αναπαριστώντας τους όρους των συναλυσώσεων ληφθέντων ακολουθιών.

Στην περίπτωση του εισβολέα ΕΔΜ, μια σύνοδος για έναν συμμετέχοντα `ag` $\in$ *Agents* ορίζεται ως η ακόλουθη 6-πλειάδα:

$$\langle \texttt{ag}, noSes, c\text{-}\texttt{ag}, \texttt{ag}_{knowledge}, \texttt{ag}_{history}^{noSes}, P \rangle$$

όπου $1 \leq j \leq \#Ses_{\text{ag}}$, με το P να αποτελεί μια *περιγραφή διεργασίας* (process description) το οποίο και ουσιαστικά περιέχει μια ακολουθία ενεργειών που

πρέπει να εκτελεστούν. Θεωρούμε τις αυτούσιες ενέργειες-εντολές αποστολή, **send** και λήψη, **receive** για την αποστολή και λήψη μηνυμάτων από/προς τους συμμετέχοντες του πρωτοκόλλου. Οι υποθέσεις που συζητήθηκαν σε προηγούμενες παραγράφους για τον εισβολέα των Dolev και Yao, υπονοούν ότι για μια συγκεκριμένη χρονική στιγμή, η συνολική γνώση του εισβολέα για την παρούσα σύνοδο πρωτοκόλλου θα είναι:

$$I_{knowledge} = \bigcup_{\mathrm{ag} \in Agents} \bigcup_{noSes=1}^{\#Ses_{\mathrm{ag}}} \{sent^{\mathrm{ag}_{noSes}}_{\max(i)}\} \cup I_{in\_knowledge}$$

για κάθε $1 \leq j \leq \#Ses_{\mathrm{ag}}$, $\mathrm{ag} \in Agents \cup \{I\}$, όπου ο συμβολισμός $I_{in\_knowledge}$ αναπαριστά την αρχική βάση-γνώσης για τον εισβολέα, και όπου $i \geq 1$ να αναπαριστά τους όρους που υπέκλεψε από την εφαρμογή του πρωτοκόλλου. Με τον όρο $I_{in\_knowledge}$ ορίζεται η αρχική γνώση του εισβολέα ΕΔΜ. Για την πιο δυνατή -από πλευράς γνώσης- του εισβολέα ΕΔΜ θα έχουμε :

$$I_{in\_knowledge} = Agents \cup \{k \in is\_key\_of^{-1}(I)\},$$

όπου ο εισβολέας γνωρίζει τα ονόματα των έντιμων συμμετεχόντων στο πρωτόκολλο, και τα δημόσια κλειδιά που χρησιμοποιεί, στην περίπτωση που συμμετέχει σε έντιμες συνόδους του πρωτοκόλλου, είτε ως εναρκτήρια οντότητα είτε ως οντότητα ανταποκριτής. Εάν δεν βρίσκεται παρών σε τέτοιες συνόδους τότε θα ισχύει $is\_key\_of^{-1}(I) = \varnothing$.

Το *μοντέλο του πρωτοκόλλου* (protocol model) ορίζεται ως μια ασύγχρονη σύνθεση μοντέλων για κάθε σύνοδο του πρωτοκόλλου, συμπεριλαμβανομένου και του μοντέλου του εισβολέας που παρεμβάλλεται πάντα της επικοινωνίας των έντιμων οντοτήτων. Η συμπεριφορά του εισβολέα θα εξαρτάται πάντα από τις ορισμένες και διαθέσιμες τακτικές επιθέσεων που θα έχει στην δομή του. Οι τακτικές επιθέσεων επιλέγονται μη ντετερμινιστικά και εκτελούνται σαν μια μοναδική νηματοειδής ενέργεια. Κάθε πιθανή *εκτέλεση* του μοντέλου ανταποκρίνεται σε μια πεπερασμένη εναλλακτική *ακολουθία καθολικών καταστάσεων* (global states) και ενεργειών, όπου:

$\tau = s_0\, \alpha_1\, s_1\, \alpha_2 \ldots s_n$, για $n \in N$ έτσι ώστε $s_{i-1} \xrightarrow{a_i} s_i$ για $0 < i \leq n$, και για την σχέση μετάβασης $\rightarrow$ που ορίζεται ως :

$$\rightarrow \subseteq S \times PS \times A \times Msgs \times S,$$

όπου *S* είναι το σύνολο των καθολικών καταστάσεων, *PS* είναι το σύνολο των εκτελούμενων συνόδων του πρωτοκόλλου και με *A* να αναπαρίσταται το σύνολο τον ονομάτων-ενεργειών. Μια σημαντική τεχνική που έχουμε εισάγει στο προτεινόμενο μοντέλο του εισβολέα, είναι η ικανότητά του να συνδυάζει τις διαθέσιμες τακτικές επιθέσεων με σκοπό την δημιουργία πακέτων επιθέσεων, πιο πολύπλοκων απαιτήσεων.

Στο γενικό μοντέλο εισβολέα του DY [34], οι ενέργειες των επιθέσεων περιλαμβάνει την αποστολή διεφθαρμένων μηνυμάτων, που δημιουργούνται με την εφαρμογή deduction κανόνων που φαίνονται παρακάτω, για όλα τα μηνύματα που βρίσκονται στην γνώση $I_{knowledge}$. Οι κανόνες αυτοί είναι οι ακόλουθοι:

- *Συναλύσωση μηνύματος (message concatenation)*:

$$\frac{msg_x \in I_{knowledge},\ msg_y \in I_{knowledge}}{msg_x \cdot msg_y,\ msg_y \cdot msg_x \in I_{knowledge}}$$

- *Προβολή μηνύματος (message projection)*:

$$\frac{msg_x \cdot msg_y \in I_{knowledge}}{msg_x, msg_y \in I_{knowledge}}$$

- *Κρυπτογράφηση μηνύματος (message encryption)*:

$$\{\bullet\}_k \in I_{knowledge}, k \in Keys \Rightarrow$$

$$\frac{msg_x \in I_{knowledge}, k \in I_{knowledge}}{\{msg_x\}_k \in I_{knowledge}} \vee \frac{msg_x \in I_{knowledge}, msg_y \in I_{knowledge}}{\{msg_x\}_{msg_y}, \{msg_y\}_{msg_x} \in I_{knowledge}}$$

- *Αποκρυπτογράφηση μηνύματος (message decryption)*:

$$\frac{\{msg_x\}_k \in I_{knowledge}, \dfrac{k \in I_{knowledge}}{k^{-1} \in I_{knowledge}}}{msg_x \in I_{knowledge}}$$

Στις υπάρχουσες προσεγγίσεις που βασίζονται τον εισβολέα του DY, υλοποιούνται οι παραπάνω κανόνες παραγωγής, με βάση πάντα την αναπαράσταση αφαίρεσης που χρησιμοποιείται για τα μηνύματα που επεξεργάζεται ο εισβολέας. Κάτω τις υποθέσεις a) ότι η μέθοδος κρυπτογράφησης που χρησιμοποιείται είναι απαραβίαστη και b) ότι είναι πιθανό να αποτραπεί ένα αυθεντικό μήνυμα να φτάσει στον αρχικό προορισμό

του, το μοντέλο του εισβολέα εκτελεί μη ντετερμινιστικά ενέργειες επιθέσεων οι οποίες εκτελούνται σε μια μόνο νηματοειδής εκτέλεση.

5.4 Ο Εισβολέας Διερεύνησης Μηνύματος (ΕΔΜ)

Ο σκοπός του μοντέλου ΕΔΜ είναι να περιλάβει όλες εκείνες τις ιδιότητες ασφαλείας οι οποίες μπορούν να χαρακτηριστούν (και να υλοποιηθούν) ως παραβιάσεις της γενικές ασφάλειας (safety) που εμποδίζει την σωστή λειτουργία του μοντέλου. Επίσης, είναι πολύ σημαντικό το μοντέλο του εισβολέα αν έχει σχεδιαστεί με αυτό τον τρόπο, ώστε ο εκάστοτε αναλυτής να μπορεί να επεκτείνει την συμπεριφορά του ΕΔΜ, με σκοπό να περιλάβει στην ανάλυση –με βάση τα [3][40][53] - τον αυτόματο έλεγχο μοντέλων πέραν των ιδιοτήτων ασφαλείας, όπως για παράδειγμα ιδιότητες βιωσιμότητας ή δικαιοσύνης (μη-αποποίησης της ευθύνης). Στον αυτόματο ελεγκτή μοντέλων SPIN, η παραβίαση της ασφάλειας ενός πρωτοκόλλου ασφαλείας μπορεί να εντοπιστεί ως την προσεγγισιμότητα μιας μη αποδεκτής κατάστασης τερματισμού (reachability of invalid end-state) ή ως μιας προτάσεως επιβεβαίωσης (assertion violation). Οι ιδιότητες που σχετίζουν βιωσιμότητα χαρακτηριστικών του πρωτοκόλλου, μπορούν να ορισθούν με διάφορους τρόπους, με την χρήση της γραμμικής χρονικής λογικής (LTL).

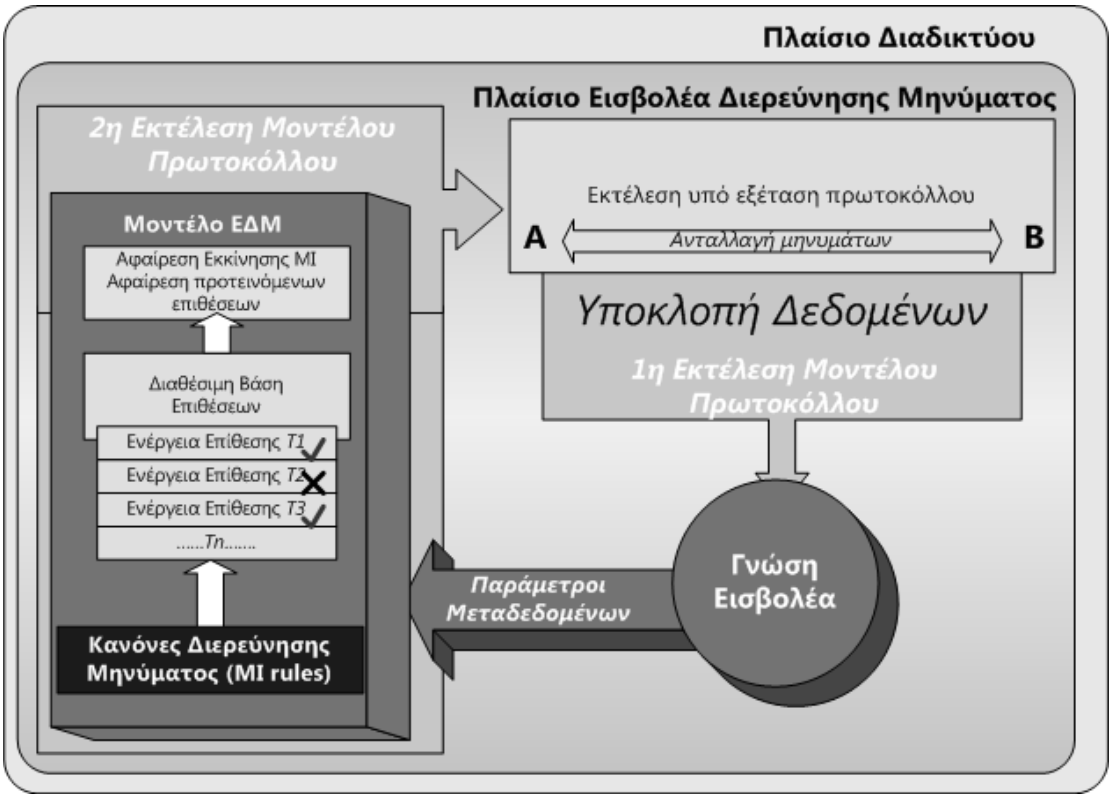
Σε παρούσα φάση, το μοντέλο ΕΔΜ συγκρίνει μεταδεδομένα που προσαρτεί σε υποκλεμμένα μηνύματα του πρωτοκόλλου. Τα μεταδεδομένα αυτά βασίζονται σε χαρακτηριστικά των μηνυμάτων, όπως πληροφορίες της χρησιμοποιούμενης κρυπτογράφησης, χρονοσφραγίδες των μηνυμάτων και βάρος (μέγεθος) των μηνυμάτων. Σκοπός των πληροφοριών αυτών είναι η αξιολόγησή και η σύγκριση μεταξύ τους, με στόχο την απορροή κάποιων συμπερασμάτων για επιθέσεις του εισβολέα οι οποίες είναι ανώφελο να πραγματοποιηθούν. Όπως ο εισβολέας ΕΠΕ, έτσι και ο εισβολέας ΕΔΜ, έχει μια ανοικτή-προς-κλείσιμο βάση επιθέσεων όπου στην μορφή του ΕΠΕ, εκτελούσε όλες τις επιθέσεις που βρίσκονταν στο σώμα του υλοποιημένες. Στην περίπτωση του εισβολέα ΕΔΜ, υλοποιείται ο *αλγόριθμος διερεύνησης μηνύματος (Message Inspection Algorithm)* ο οποίος λαμβάνει αποφάσεις με βάση τα προαναφερθείσα μεταδεδομένα για το ποιες επιθέσεις μπορούν να

αποφευχθούν και επομένως να αφαιρεθούν από τον εισβολέα, προς διευκόλυνση της ανάλυσης. Οι επιθέσεις που έχουν υλοποιηθεί στην βάση αυτή του εισβολέα ΕΔΜ είναι οι επαναλήψεις μηνυμάτων (message replays), η παραβίαση ακεραιότητας μηνύματος (message integrity violation), η επίθεση παράλληλης συνόδου (parallel sessions attacks) και οι επιθέσεις παραβίασης τύπου μηνύματος (type-flaws attacks), οι οποίες και περιγράφθηκαν φορμαλιστικά στο προηγούμενο κεφάλαιο και παρουσιάζονται στο [7] και [10].

Το μοντέλο του ΕΔΜ μπορεί να θεωρηθεί σαν μια προσέγγιση βελτίωσης, η οποία βασίζεται στην συμβολική αναπαράσταση η οποία αποφεύγει την επακριβώς απαρίθμηση των μηνυμάτων που δημιουργούνται στην γνώση του εισβολέα $I_{knowledge}$. Προς αντίθεση της χρήσης των κανόνων παραγωγή του εισβολέα DY για την παραγωγή όλων των πιθανών συνδυασμών δημιουργίας διεφθαρμένων μηνυμάτων, το μοντέλο ΕΔΜ καταγράφει τα υποκλεμμένα μηνύματα σε μια εκτέλεση προσομοίωσης και την ίδια ώρα, δημιουργεί διακριτές τιμές μεταδεδομένων για κάθε καταγραμμένο μήνυμα. Με αυτό τον τρόπο ο εισβολέας εκμεταλλεύεται μόνο τα μεταδεδομένα των μηνυμάτων που αρχικά δημιουργήθηκαν (με βάση χαρακτηριστικά πρωτοκόλλου που έχει ορίσει ο αναλυτής) και όχι ολόκληρο το κάθε ένα μήνυμα. Οι *κανόνες διερεύνησης μηνύματος (MI rules)*, οι οποίοι θα οριστούν στις επόμενες παραγράφους, αποφασίζουν ποιες θα είναι αυτές οι ενέργειες επιθέσεων οι οποίες μπορούν να προκαλέσουν προβλήματα στις οντότητες του πρωτοκόλλου, και άρα πρέπει να συμπεριληφθούν στην όλη ανάλυση, και ποιες όχι. Είναι γνωστό για παράδειγμα ότι μια επίθεση κρυπτογράφησης (encryption scheme attack) θα αποτύχει εάν ο εισβολέας δεν έχει στη βάση γνώσης του το σωστό κλειδί αποκρυπτογράφησης. Συνεπώς, τα κρυπτογραφημένα μηνύματα μπορούν να χειραγωγηθούν με διαφορετικό τρόπο από ότι τα μηνύματα απλού κειμένου (plain text messages) ή τα μηνύματα τα οποία είναι μερικώς κρυπτογραφημένα. Μια προφανής βελτίωση του εισβολέα, θα ήταν η αφαίρεση της ικανότητας από την βάση επιθέσεών του, της παραπάνω επίθεσης (ή του κανόνα αποκρυπτογράφησης μηνύματος στον DY εισβολέα), αφού είναι γνωστό ότι δεν πρόκειται να πετύχει, εάν ο εισβολέας δεν κατέχει το σωστό κλειδί. Έτσι με αυτό τον τρόπο ο αναλυτής μπορεί να χρησιμοποιήσει το μοντέλο ΕΔΜ με σκοπό να μειώσει τις

καταστάσεις εκείνες που δεν θα είναι δυνατό να παρουσιάσουν κάποια παραβίαση της ασφάλειας, με βάση τους καθορισμένους κανόνες διερεύνησης.

Στην εικόνα 5.4.1 παρουσιάζεται το γενικό σχήμα με βάση του οποίου το μοντέλο εισβολέα ΕΔΜ συμπεριφέρεται ως μια ενδιάμεση οντότητα (man-in-the-middle entity), η οποία κυριαρχεί πάνω στο επικοινωνιακό μέσο που χρησιμοποιούν οι έντιμες οντότητες *A* και *B* του πρωτοκόλλου. Κάθε μήνυμα αξιολογείται από συγκεκριμένους μηχανισμούς χαρακτηρισμών μηνυμάτων οι οποίες θα ονομάζονται *συναρτήσεις μεταδεδομένων (metadata functions)*. Κάθε μήνυμα αντιστοιχεί σύνολο χαρακτηριστικών, όπου ο αριθμός τους είναι όσες είναι και οι συναρτήσεις των μεταδεδομένων. Στη συνέχεια ο εισβολέας συμβουλεύεται την βάση των επιθέσεών του που έχει ήδη διαθέσιμες, με σκοπό να αποφασίσει ποιες από αυτές μπορούν να απενεργοποιηθούν από τον αναλυτή, χωρίς όμως να ακυρώσει τις επιθέσεις που μπορεί να επιτύχουν να αποκαλύψουν ένα λάθος ασφαλείας. Ο αναλυτής στη συνέχεια προχωρά σε έναν δεύτερο έλεγχο επαλήθευσης αυτή τη φορά του πρωτοκόλλου με το αλλαγμένου μοντέλου του ΕΔΜ.



Εικόνα 5.4.1 Το μοντέλο του Εισβολέα Διερεύνησης Μηνύματος ΕΔΜ

Στη γενική περίπτωση χρήσης του μοντέλου ΕΔΜ, ο εισβολέας έχει την ικανότητα να ξεκινήσει μόνος του, (μία ή περισσότερες τυχαίες) καινούργιες συνόδους του υπό εξέταση πρωτοκόλλου, με έναν από τους έντιμες οντότητες. Ο εισβολέας επαναχρησιμοποιεί με βάση τις ενέργειες των επιθέσεων που έχει στην διάθεσή του, προηγούμενα υποκλεμμένα μηνύματα, με σκοπό την επιτυχή εφαρμογή των επιθέσεών του έναντι τον συμμετεχόντων στο πρωτόκολλο. Ένα σημαντικό γεγονός που πρέπει να αναφερθεί είναι ότι το μοντέλο ΕΔΜ δεν εξαπολύει απευθείας την επίθεση παραβίαση της ακεραιότητας σε κάθε μήνυμα που λαμβάνει, όπως πραγματοποιούν σε αυτή την περίπτωση τόσο το μοντέλο του εισβολέα DY (με βάση τους κανόνες παραγωγής του) όσο και το προηγούμενο μοντέλο στο οποίο αναφερθήκαμε του εισβολέα ΕΠΕ (με βάση της συναλύσωση μηνυμάτων). Αντίθετα, δοκιμάζει την συγκεκριμένη τακτική επίθεσης σε συγκεκριμένα μηνύματα τα οποία μπορούν να διαβαστούν από τον εισβολέα. Έτσι το μοντέλο περιορίζει σημαντικά την κληρονομούμενη πολυπλοκότητα που θα εισήγαγε μια τέτοια ενέργεια, αφού θα αναγκάζονταν να παραχθεί όλοι οι δυνατοί συνδυασμοί συναλυσώσεων όλων των μηνυμάτων, αναλύοντάς τα μέσα στη γνώση $I_{knowledge}$ του εισβολέα.

5.4.1 Μεταδεδομένα Μηνύματος (Message metadata)

Ας θεωρήσουμε ένα πρωτόκολλο $\overline{\overline{\Pr}}$ μεταξύ συμμετεχόντων οντοτήτων A, B, . . . , Z ϵ Agents και το z να αναπαριστά τον αριθμό των βημάτων του πρωτοκόλλου. Εκτελούμε μια προσομοίωση του $\overline{\overline{\Pr}}$ για έναν πεπερασμένο αριθμό συνόδων του πρωτοκόλλου, που συμβολίζεται με n. Χρησιμοποιούμε τα μηνύματα του επόμενου πίνακα, με σκοπό την εξαγωγή μεταδεδομένων για την γνώση του εισβολέα $I_{knowledge}$,

$$
\begin{array}{ll}
\sigma\acute{\upsilon}\nu o\delta o\iota \rightarrow & \begin{array}{cccc} 1^{st} & 2^{nd} & . & n^{th} \end{array} \\
\downarrow \beta\acute{\eta}\mu\alpha\tau\alpha & \begin{array}{c} 1^{st} \\ 2^{nd} \\ . \\ z^{th} \end{array} \overbrace{\begin{bmatrix} msg_{1,1} & . & . & msg_{1,n} \\ msg_{2,1} & . & . & . \\ . & . & . & . \\ msg_{z,1} & . & . & msg_{z,n} \end{bmatrix}}^{\#Ses_{\mathrm{ag}}}
\end{array}
$$

$$
\bigcup_{\mathrm{ag} \in Agents} \; \bigcup_{noSes=1} \{ sent^{\mathrm{ag}}_{\max(i)} \}
$$

με $msg_{a,b}$ να αναπαριστά ένα οποιοδήποτε μήνυμα το οποίο στάλθηκε στο *a* βήμα του πρωτοκόλλου στη σύνοδο *b* ενός συμμετέχοντα `ag` ϵ `Agents`. Το μοντέλο του εισβολέα αποθηκεύει τα μεταδεδομένα αυτά για κάθε μήνυμα που τοποθετείται στον παραπάνω πίνακα. Επιπλέον τιμές των μεταδεδομένων, (πέρα από το πότε και σε ποια σύνοδο ελήφθησαν τα μηνύματα) καθορίζονται από παραμετρικές συναρτήσεις τύπου *p(a,b)* για το *a* βήμα του πρωτοκόλλου στη σύνοδο *b*, ως ακολούθως:

Ορισμός 1. *Η συνάρτηση μεταδεδομένων p(a,b) αποτελεί μια K^{th} παραμετρική συνάρτηση μεταδεδομένων με K υποσυναρτήσεις,*

$$p(a,b) = \begin{cases} p(msg_{a,b})^1 \\ p(msg_{a,b})^2 \\ \quad . \quad , \ K \geq 1 \\ \quad . \\ p(msg_{a,b})^K \end{cases}$$

όπου η τιμή του $p(msg_{a,b})^{mtd}$, $1 \leq mtd \leq K$ εξαρτάται από το χαρακτηριστικό των μεταδεδομένων mtd που εκφράζεται (για παράδειγμα κρυπτογράφηση, μέγεθος) για το συγκεκριμένο μήνυμα $msg_{a,b} \in Msgs$ το οποίο στάλθηκε στο βήμα πρωτοκόλλου a στην σύνοδο b.

Βασιζόμενοι στην υλοποιημένη συνάρτηση διερεύνησης μηνύματος, η τιμή του $p(msg_{a,b})^{mtd}$ μπορεί να αναπαριστά για παράδειγμα την τιμή του μεγέθους ενός μοναδικού μηνύματος ή μια λογική τιμή, για τον αν το συγκεκριμένο μήνυμα είναι αναγνώσιμο ή όχι. Οι επόμενοι ορισμοί θα συνθέσουν τον μηχανισμό της διερεύνησης μηνύματος για τον εισβολέα ΕΔΜ, για την περίπτωση χρήσης που επιλέχθηκε να εφαρμοστεί ο εισβολέας του πρωτοκόλλου NSPK.

Ορισμός 1.1. *Η υποσυνάρτηση $p(msg_{a,b})^{Encryption}$ του p(a,b) αναπαριστά την* ***αναγνωσιμότητα*** *του υποκλεμμένου μηνύματος. Η εικόνα του $p(msg_{a,b})^{Encryption}$ θα είναι μια από τις τιμές του συνόλου E={0, 1, 2}, όπου κάθε μια τιμή αναπαριστά μια διακριτή περίπτωση τύπου κρυπτογραφίας: το 0 χρησιμοποιείται για το μη κρυπτογραφημένο μήνυμα (απλό κείμενο), το 1 για το μερικώς κρυπτογραφημένο μήνυμα και το 2 για το πλήρως κρυπτογραφημένο μήνυμα.*

$$p(a,b) = p(msg_{a,b})^{Encryption} = \begin{cases} 0 & \text{όταν } msg_{a,b} = msg_u \\ 1 & \text{όταν } msg_{a,b} = msg_y \cdot \{msg_u\}_k \cdot msg_z \\ 2 & \text{όταν } msg_{a,b} = \{msg_u\}_k \end{cases},$$

$$\forall (a,b) \in [1..z] \times [1..n]$$

για κάποια $msg_u \in Msgs$, $k \in Keys$ και $msg_y \cdot msg_z \neq ( )$, όπου τουλάχιστον ένα από τα συναλυσωμένα μηνύματα δεν είναι κενό.

Οι παραπάνω υποσυναρτήσεις ωθούν το μοντέλο ΕΔΜ να ενεργεί ως μια μηχανή αποφάσεων, η οποία ομαδοποιεί τις διαθέσιμες ενέργειες των επιθέσεων σε τρεις διαφορετικές κατηγορίες λειτουργίας με τις ακόλουθες συμβολικές τιμές, 0 για την μη ύπαρξη κρυπτογράφησης στο μήνυμα που ελήφθη, 1 για την μερική κρυπτογράφηση και 2 για το πλήρως κρυπτογραφημένο μήνυμα. Με αυτό τον τρόπο, ο εισβολέα ΕΔΜ υλοποιεί την επιπλέον δυνατότητά του να επιλέξει ενέργειες επιθέσεων, για τις οποίες –βασιζόμενος σε γνωστές αρχές ασφαλείας- γνωρίζει ότι δεν θα πετύχουν τον στόχο τους. Για παράδειγμα, αναφέρεται η επίθεση σε ένα μήνυμα με τιμή αναγνωσιμότητας 2 (πλήρως κρυπτογραφημένο). Με βάση τις υποθέσεις που έχουμε ορίσει για τον εισβολέα, ο ίδιος δεν θα μπορέσει σε καμιά περίπτωση να διαβάσει τα περιεχόμενα του μηνύματος, εάν αυτός πρώτα δεν κατέχει το αντίστοιχο κλειδί της αποκρυπτογράφησης μέσα στην γνώση του $I_{knowledge}$. Σε αυτή και μόνο την περίπτωση ο εισβολέας θα αποπειραθεί να αποκρυπτογραφήσει το μήνυμα (και όχι σε όλες τις περιπτώσει με όλα τα δυνατά κλειδιά, όπως το μοντέλο εισβολέα του DY). Έτσι, κατά τον έλεγχο μοντέλων μιας σειράς από ανούσιες επιθέσεις, ο *αλγόριθμος διερεύνησης μηνύματος (MI algorithm)* θα αναλάβει να ενημερώσει τον αναλυτή για την πιθανότητα διόρθωσης του μοντέλου του εισβολέα του, με την αφαίρεση αχρείαστων επιθέσεων από την δομή του. Με αυτό τον τρόπο το μοντέλο του εισβολέα απλοποιείται και εφαρμόζει τις μόνο τις εναπομείναντες απαραίτητες επιθέσεις του. Κάθε επίθεση που θα έχει διαθέσιμη ο εισβολέας ΕΔΜ, θα ανήκει σε μια γενικότερη κατηγορία επιθέσεων, οι οποίες και παρουσιάστηκαν στο κεφάλαιο 4 αυτής της διατριβής, ως συνακολουθίες ενεργειών αποστολής (send) και λήψης (receive) μηνυμάτων.

Ορισμός 1.2. *Η υποσυνάρτηση $p(msg_{a,b})^{Size}$ για το p(a,b) αναπαριστά το* ***μέγεθος*** *κάθε υποκλεμμένου μηνύματος σε δυφία (bits). Η εικόνα αυτής της συνάρτησης θα περιλαμβάνει συμβολικές τιμές από το σύνολο S={s: s ϵ ℵ and s>0} με φυσικούς αριθμούς να αναπαριστούν το άθροισμα τιμών επιμέρους τμημάτων για το κάθε μήνυμα του πρωτοκόλλου.*

$$p(a,b) = p(msg_{a,b})^{Size} = \begin{cases} size(msg_{a,b}) & \text{για } size \subseteq Msgs \times S \\ 0 & \text{Εάν γγι } msg_{a,b} \text{ ισχύει sent (null)} \end{cases},$$

$$\forall (a,b) \in [1..z] \times [1..n]$$

Η συγκεκριμένη υποσυνάρτηση ωθεί το μοντέλο ΕΔΜ να ανιχνεύει το υποκλεμμένο μήνυμα σαν μια ποσότητα πληροφορίας, εκτιμώντας συμβολικά το μέγεθός του, το οποίο θα βασίζεται από τα επιμέρους μέρη που αποτελείται το μήνυμα (για παράδειγμα μήνυμα που περιέχει μια ταυτότητα-ID μιας οντότητας και ένα 32-διφύων μέγεθος τυχαίο αριθμού). Όταν ο εισβολέας ανιχνεύσει οι τιμές των μεταδεδομένων της υποσυνάρτησης μεγέθους για δυο υποκλεμμένα μηνύματα από διαφορετικές συνόδους του πρωτοκόλλου και σε διαφορετικό βήμα αυτού, ότι είναι ίδιες, τότε με βάση το [43] είναι πιθανή η επιτυχία μιας επίθεσης παραβίασης τύπου του μηνύματος (type flaw attack), ανεξάρτητα από το εάν η συγκεκριμένη επίθεση πετύχει αργότερα ή όχι. Επίσης, εάν το πρωτόκολλο διακοπεί από κάποιο λάθος (σε περίπτωση που το μοντέλο του πρωτοκόλλου καλύπτει το συγκεκριμένο ενδεχόμενο), ή η ενδιάμεση οντότητα του εισβολέα σταματήσει να υποκλέπτει τα μηνύματα εξαιτίας λήξης του χρόνου απάντησης (timeout) για μια έντιμη οντότητα, τότε η τιμή της υποσυνάρτησης για το μέγεθος του μηνύματος (και για τις επόμενες τιμές) θα είναι μηδενικό (0), μέχρι το τέλος της παρούσας συνόδου.

Οι στήλες του παραπάνω πίνακα που παρουσιάστηκε στην αρχή της παραγράφου αναπαριστά όπως ειπώθηκε τα διακριτά βήματα και τις καταγεγραμμένες συνόδους του πρωτοκόλλου στην διάρκεια της πρώτης φάσης της προσομοίωσης, για την ανάλυση με το μοντέλο του ΕΔΜ. Οι στήλες αυτές αυξάνονται ακολουθιακά με προσαύξηση της μονάδας δημιουργώντας έτσι όρους θετικών ακέραιων αριθμών, που αναπαρίστανται με b_m ϵ ℵ και αρχικό όρο b_0 = 1. Οι διαφορετικοί όροι μπορούν να θεωρηθούν ως χρονοσφραγίδες των μηνυμάτων, μιας και αν γνωρίσουμε το διακριτό βήμα και την διακριτή τιμή της συνόδου του πρωτοκόλλου, μπορούμε χρονικά μα ταξινομήσουμε τα

μηνύματα (και τα μεταδεδομένα τους). Η συγκρισιμότητα αυτή του χρονικού σημείου λήψης (δημιουργίας) μηνύματος (μεταδεδομένων), είναι χρήσιμη για την εφαρμογή κάποιον επιθέσεων οι οποίες και εξαρτώνται στην διαθεσιμότητα ενός υποκλεμμένου μηνύματος –όπου μέρος αυτού είναι και μια χρονοσφραγίδα- σχετιζόμενο με την χρονοσφραγίδα εκείνη του τελευταίου μηνύματος που υποκλαπεί στην ίδια σύνοδο του πρωτοκόλλου. Για παράδειγμα, μια επίθεση πλαστοπροσωπίας (impersonation attack) μεταξύ δύο παράλληλων συνόδων του πρωτοκόλλου, βάση του [48] και του [99], δεν μπορούν να επαναχρησιμοποιήσουν μέρη μηνυμάτων με τιμές χρονοσφραγίδων μεγαλύτερες από την τιμή της χρονοσφραγίδας του τελευταίου υποκλεμμένου μηνύματος στην επιτιθέμενη σύνοδο του πρωτοκόλλου. Εάν είναι απαραίτητο, το μοντέλο του ΕΔΜ μπορεί να ενοποιήσει επιπρόσθετες υποσυναρτήσεις μεταδεδομένων, πέρα από τις προηγούμενες που αναφέρθηκαν σε αυτή την παράγραφο.

Μετά από τον ορισμό όλων υποσυναρτήσεων για αυτή την έκδοση του μοντέλου ΕΔΜ, ορίζουμε την *Πίνακα Γνώσης του Εισβολέα (Intruder Knowledge Table, IKT),* με συμβολισμό *[Ikt]* ως εξής:

Ορισμός 2. *Σε έναν μοντέλο ΕΔΜ ορίζουμε τον Πίνακα Γνώσης του Εισβολέα [IKT], ο οποίος και παράγεται από τις τιμές των παραμετρικών υποσυναρτήσεων μεταδεδομένων p(a,b) για όλα τα υποκλεμμένα μηνύματα* $msg_{a,b}$*, με* $(a,b) \in [1..z] \times [1..n]$*, ως:*

$$[Ikt] = \begin{bmatrix} p(1,1) & . & . & p(1,n) \\ p(2,1) & . & . & . \\ . & . & . & . \\ p(z,1) & . & . & p(z,n) \end{bmatrix},$$

$$p(a,b)=\begin{cases} p(msg_{a,b})^{Size}=\begin{cases} size(msg_{a,b}) & \text{για } size \subseteq Msgs \times S \\ 0 & \text{Εάν } msg_{a,b} \text{ ισχύει sent (null)} \end{cases} \\ p(msg_{a,b})^{Encryption}=\begin{cases} 0 & \text{όταν } msg_{a,b}=msg_u \\ 1 & \text{όταν } msg_{a,b}=msg_y \cdot \{msg_u\}_k \cdot msg_z \\ 2 & \text{όταν } msg_{a,b}=\{msg_u\}_k \end{cases} \\ \text{προσθήκη άλλων…} \end{cases}$$

$$\forall (a,b) \in [1..z] \times [1..n]$$

για κάποιες συμβολικές τιμές του S={s: s ϵ ℵ and s>0} με φυσικούς αριθμούς, και κάποια μηνύματα $msg_u \in Msgs$, $k \in Keys$ *και* $msg_y \cdot msg_z \neq ( )$. *Οι ιδιότητες που θα ισχύουν για τον πίνακα [Ikt] θα είναι:*

- *if* $msg_{a,b}$ *is never sent (null) then p(a,b)=0, που σημαίνει ότι ο εισβολέας δεν έχει υποκλέψει κανένα μήνυμα το οποίο στάλθηκε στο βήμα α και στη σύνοδο b του πρωτοκόλλου*
- *if p(a,b)=0 then p(a+φ, b)=0* $\forall \varphi \in \aleph$: $a+\varphi \leq z$, *που σημαίνει ότι εάν ο εισβολέας δεν λάβει ένα μήνυμα πρωτοκόλλου σε ένα βήμα α, τότε θεωρεί ότι και στα επόμενα βήματα μέχρι το τελευταίο μήνυμα της συνόδου, δεν λάβει επακόλουθο μήνυμα*

Οι ιδιότητες του πίνακα *[Ikt]* θέτουν ικανή την χειραγώγηση των δημιουργημένων μεταδεδομένων, για την απορροή βελτιώσεων για το συγκεκριμένο πρωτόκολλο ασφαλείας που εξετάζεται, όπως η μείωση του παραγομένου χώρου καταστάσεων μέσω της απλοποίησης του χρησιμοποιούμενου μοντέλου εισβολέα.

Για ένα πρωτόκολλο $\overline{\Pr}$ και ένα υποκλεμμένο μήνυμα στο βήμα του πρωτοκόλλου *α*, στη σύνοδο *b* ο εισβολέας ΕΔΜ καλείται να συμπληρώσει όλο τον πίνακα *[Ikt]* με τιμές μεταδεδομένων *p(a, b)*. Εάν $a < z$, τότε όλες οι εισαγόμενες τιμές $p(a+\varphi, b)$ με $\varphi \in \aleph$: $a+\varphi \leq z$ κρατούν την αρχική τους τιμή, η οποία είναι μηδενική (0), μέχρις ότου ο εισβολέας υποκλέψει το αμέσως επόμενο μήνυμα. Εάν για κάποιο λόγο, η σύνοδος του πρωτοκόλλου σταματήσει, τότε οι τιμές $p(a+\varphi, b)$ θα παραμείνουν μηδενικές.

Ορισμός 3. *Για να συγκρίνουμε δύο διαφορετικούς όρους του Πίνακα Γνώσης του Εισβολέα [Ikt], έστω p(a, b) και p(c, d), έτσι ώστε* $a \neq c \vee b \neq d$*, ορίζουμε τον επόμενο τελεστή (*$\cong$*):*

$$p(a, b) \cong p(c, d), \text{ if}$$

$$p(msg_{a,b})^1 = p(msg_{c,d})^1 \vee p(msg_{a,b})^2 = p(msg_{c,d})^2 \vee \ldots \vee p(msg_{a,b})^K = p(msg_{c,d})^K$$

5.4.2 Το μοντέλο εισβολέα ΕΔΜ στη πράξη

Στη πρώτη αυτή φάση της προσομοίωσης με τον εισβολέα ΕΔΜ, η γνώση του εισβολέα ενημερώνεται με τα νέα δεδομένα του πίνακα *[Ikt]*. Έτσι η γνώση του ΕΔΜ μεταλλάσσεται σε:

$$I_{knowledge} = \bigcup_{\text{ag} \in Agents} \bigcup_{noSes=1}^{\#Ses_{\text{ag}}} \{sent^{\text{ag}_{noSes}}_{\max(i)}\} \cup I_{in\_knowledge} \cup \{[Ikt]\}.$$

Καμία επίθεση δεν εξαπολύεται κατά την διάρκεια της προσομοίωσης, φέρνοντας τον εισβολέα σε μια θέση παθητικού ωτακουστή. Η προσομοίωση αυτή αντίθετα εφαρμόζει τον αλγόριθμο διερεύνησης μηνύματος στην ανανεωμένη γνώση του εισβολέα $I_{knowledge}$ ενεργοποιώντας εσωτερικές του διεργασίες (χωρίς την αλληλεπίδραση άλλων οντοτήτων) για την επεξεργασία των ακολουθιών των μηνυμάτων $\bigcup_{\text{ag} \in Agents} \bigcup_{noSes=1}^{\#Ses_{\text{ag}}} \{sent^{\text{ag}_{noSes}}_{\max(i)}\}$ για όλες τις οντότητες $\text{ag} \in Agents$, με βάση τον πίνακα *[Ikt]*. Το αποτέλεσμα αυτής της προσομοίωσης εξάγει χρήσιμες πληροφορίες αφαίρεσης των επιθέσεων που μπορούν με ασφαλή τρόπο να αφαιρεθούν από την βάση επιθέσεων του εισβολέα ΕΔΜ.

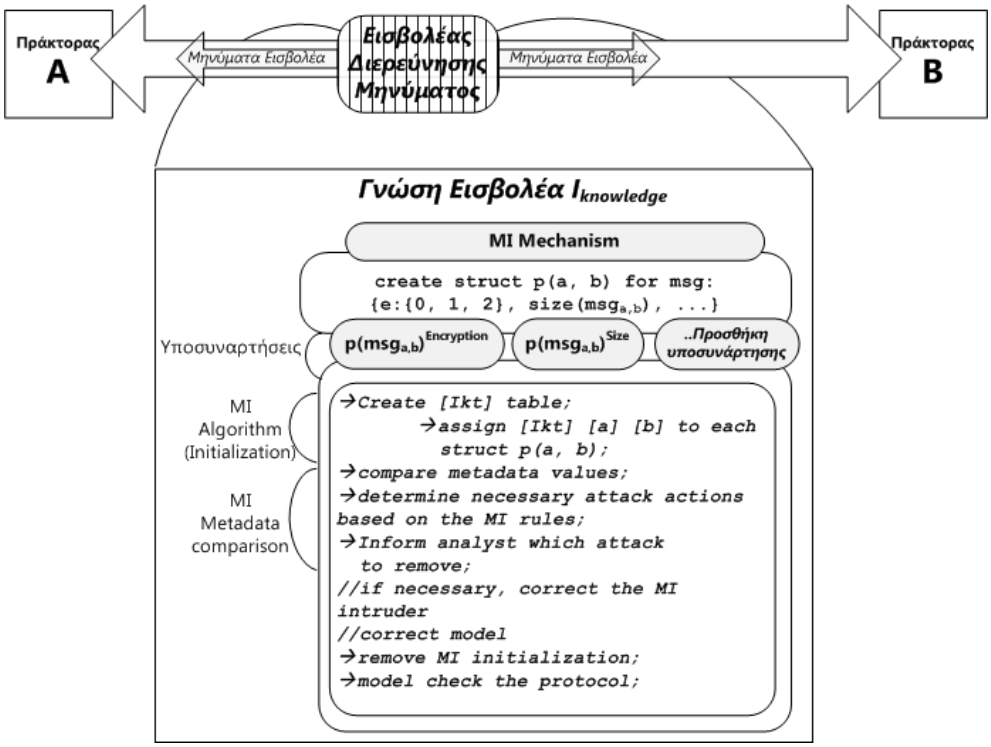
Η εικόνα 2 εισάγει τις δύο ξεχωριστές φάσεις, του *αλγόριθμου διερεύνησης*. Θεωρούμε δύο πράκτορες *A (Initiator)*, *B (Responder)* ε *Agents* οι οποίοι ανταλλάσσουν μηνύματα με βάση τις περιγραφές διεργασιών τους P_I και P_R για ένα πρωτόκολλο έστω $\overline{\overline{\Pr}}$. Ο εισβολέας συμπεριφέρεται σαν μια ενδιάμεση οντότητα η οποία και μπορεί να αιχμαλωτίσει όλα τα μηνύματα του πρωτοκόλλου. Για κάθε ένα τέτοιο μήνυμα το μοντέλο ΕΔΜ δημιουργεί μια δομή (*structure*) *p*(*a*, *b*) η οποία και θα περιέχει τις τιμές για όλες τις ορισμένες υποσυναρτήσεις δεδομένων που έχουν εξετάσει το συγκεκριμένο μήνυμα. Η δομή αυτή κατά την διάρκεια του ελέγχου μοντέλων με το εργαλείο SPIN, μπορεί εύκολα να ορισθεί με γλώσσας εκτέλεσης προδιαγραφών, όπως στη PROMELA

[ΣΠΙΝ]. Ο αναγνώστης μπορεί να αναφερθεί στο παράρτημα Α, για περισσότερες πληροφορίες που αφορούν την γλώσσα PROMELA που χρησιμοποιήθηκε για την δημιουργία του μοντέλου ΕΔΜ [70]. Η δομή αυτή του μηνύματος, θα ανταποκρίνεται στον πίνακα *[Ikt]* για το a^{th} βήμα της συνόδου *b* του πρωτοκόλλου, όπως φαίνεται στην εικόνα 2. Οι τιμές των μεταδεδομένων στις $p(a, b)$ δομές που δημιουργεί ο εισβολέας χρησιμοποιούνται για την σύγκριση μεταξύ τους. Ουσιαστικά συγκρίνονται με μεγαλύτερη λεπτομέρεια τεχνικά χαρακτηριστικά των μηνυμάτων και όχι τα μηνύματα ως ολότητες. Κάτι τέτοιο αποδεικνύεται εξαιρετικά χρήσιμο στην διάρκεια του ελέγχου μοντέλων, μιας και ένα μήνυμα είναι σύνηθες να αποτελείται από πολλά διακριτά και κρυπτογραφημένα μέρη πληροφορίας. Με βάση το αποτέλεσμα των συγκρίσεων αυτών, ο εισβολέας βασιζόμενος στους *κανόνες διερεύνησης μηνύματος (MI rules)*, μπορεί να προτείνει την αφαίρεση κάποιον από τις επιθέσεις του που έχει διαθέσιμες στη βάση του.

Έστω ένα πρωτόκολλο $\overline{\overline{\text{Pr}}}$ το οποίο για την ολοκλήρωσή του απαιτεί τέσσερα (4) διακριτά βήματα, και το οποίο εκτελείται ταυτόχρονα για δύο ξεχωριστές συνόδους. Στην φάση *εκκίνησης της διερεύνησης μηνύματος (MI initialization phase)* ο εισβολέας καταγράφει όλα τα υποκλεμμένα μηνύματα για τις δύο συνόδους, συμπληρώνοντας με αυτό τον τρόπο τις δύο (2) πρώτες στήλες του πίνακα γνώσης *[Ikt]*. Ο αριθμός των πεδίων στις δημιουργημένες δομές *p(a,b)* θα είναι ο αριθμός των ορισμένων υποσυναρτήσεων μεταδεδομένων που θα υπάρχουν υλοποιημένες τον εισβολέα ΕΔΜ. Όταν ο εισβολέας υποκλέψει ένα μήνυμα, ανανεώνει τον πίνακα *[Ikt]* έτσι ώστε αργότερα να χρησιμοποιηθεί αποτελεσματικά για τις συγκρίσεις των μεταδεδομένων. Οι συγκρίσεις αυτές θα λάβουν χώρα με την χρήση του τελεστή $\cong$ όπως περιγράφθηκε στον ορισμό 3, όπου τιμές μεταδεδομένων μηνυμάτων διαφορετικού βήματος του πρωτοκόλλου και ίδιας ή διαφορετικές συνόδου συγκρίνονται μεταξύ τους.

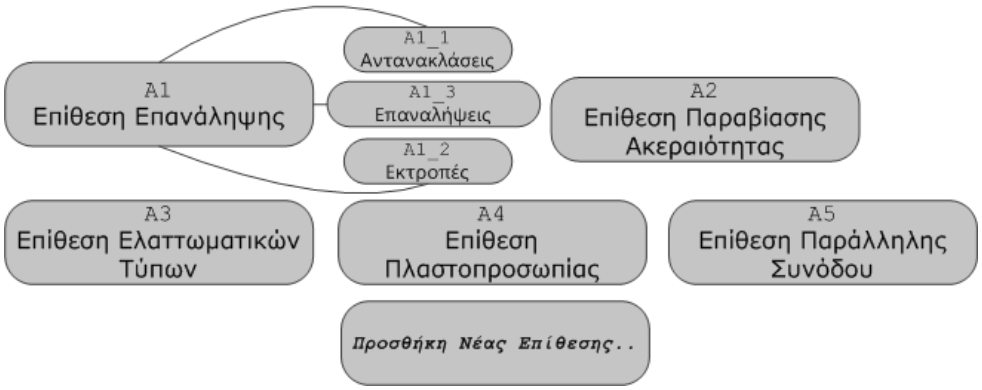
Το αποτέλεσμα κάθε σύγκρισης αποφασίζει με βάση τους κανόνες, ποιες ενέργειες επιθέσεων είναι ανώφελο να γίνουν, προτείνοντας με αυτό τον τρόπο τον αναλυτή να τις αφαιρέσει από την βάση του εισβολέα. Οι ενέργειες επιθέσεων οι οποίες δεν μπορούν να συνεισφέρουν στον αυτόματο έλεγχο μοντέλων όλων των βημάτων του πρωτοκόλλου, για πεπερασμένο αριθμό

συνόδων, προτείνονται στο αποτέλεσμα της προσομοίωσης, στην πρώτη αυτή φάση της όλης ανάλυσης. Στη συνέχεια ο αναλυτής αφαιρεί όχι μόνο τις προτεινόμενες από τον αλγόριθμο επιθέσεις αλλά και τον ίδιο τον αλγόριθμο που συμπληρώνει τον πίνακα *[Ikt]* και συγκρίνει τα μεταδεδομένα αυτού. Έπειτα συνεχίζει την ανάλυση του μοντέλου του πρωτοκόλλου με το ανανεωμένο μοντέλο του εισβολέα ΕΔΜ, στην δεύτερη φάση, την φάση της εξαντλητικής επαλήθευσης, προσδοκώντας σε έναν μειωμένο και εύκολο να αναλυθεί παραγόμενο χώρο καταστάσεων.



Εικόνα 5.4.2 Ο αλγόριθμος διερεύνησης μηνύματος και οι φάσεις του κατά την λειτουργία του εισβολέα ΕΔΜ

Η εικόνα 5.4.2, περιγράφει την ανοικτή-προς-κλείσιμο βάση επιθέσεων που έχει διαθέσιμες ο εισβολέας ΕΔΜ, στην παρούσα του έκδοση. Οι ενέργειες των επιθέσεων αυτών, όπως φαίνονται στην εικόνα 5.4.3, έχουν υλοποιηθεί μέσα στο μοντέλο του εισβολέα ΕΔΜ, τα οποία αρχικά έχουν περιγραφθεί και ταξινομηθεί στα [7][10].



Εικόνα 5.4.3 Ενέργειες επιθέσεων διαθέσιμες για το μοντέλο εισβολέα ΕΔΜ

Ο πίνακας 5.4.1 εισάγει τις περιγραφές των ενεργειών επιθέσεων (A1 μέχρι και A5) που έχει διαθέσιμες το μοντέλο ΕΔΜ στη βάση του. Οι περιγραφές αυτές αποτελούν ακολουθίες *αποστολής (send)* και *λήψης (receive),* καθώς και τον συσχετισμό τον επιθέσεων με τα στοιχεία του πίνακα γνώσης του εισβολέα *[Ikt]*.

Πίνακας 5.4.1Ενέργειες επιθέσεων για το μοντέλο ΕΔΜ και αντιστοίχησή τους με τα μεταδεδομένα του πίνακα γνώσης του εισβολέα *[Ikt]*

Ενέργεια Επίθεσης	Περιγραφή Επίθεσης
A1	*Επέλεξε ένα υποκλεμμένο μήνυμα και στείλε το στον αποστολέα του (A1_2) ή στον αρχικό προορισμό του (A1_3) ή σε κάποια άλλη οντότητα η οποία δεν είναι ούτε ο αρχικός αποστολέας ούτε παραλήπτης του μηνύματος (A1_1)*
A2	*Αντικατέστησε ένα υποκλεμμένο μήνυμα με ένα άλλο μήνυμα ή με ένα διεφθαρμένο μήνυμα μέσω ενέργειας συναλύσωσης (concatenation) προερχόμενο από την βάση γνώσης του εισβολέα* $I_{knowledge}$
A3	*Αντικατέστησε ολόκληρο (ή μέρος) υποκλεμμένου μηνύματος το οποίο αντιστοιχεί σε ένα p(a,b) με ένα προηγούμενο υποκλεμμένο μήνυμα (ή μέρος)*
A4	*Πλαστοπροσωπία ενός πράκτορα* $ag \in Agents$ *χρησιμοποιώντας ένα προηγούμενο υποκλεμμένο μήνυμα που αντιστοιχεί σε ένα p(a,b) με a=1*
A5	*Εκκίνηση νέας συνόδου πρωτοκόλλου ή χειραγώγηση υπάρχουσας χρησιμοποιώντας μήνυμα προηγούμενης συνόδου*

Οι ενέργειες επιθέσεων της ομάδας A1 αντιπροσωπεύουν την αποστολή ενός υποκλεμμένου μηνύματος ή (εάν συνδυαστεί με άλλη σύνοδο του πρωτοκόλλου) ένα διεφθαρμένο μήνυμα, είτε προς τον αρχικό προοριζόμενο παραλήπτη ή στον ίδιο τον αποστολέα ή σε κάποιον τρίτο διαφορετικό από αυτούς. Οι τιμές των μεταδεδομένων $p(a, b)$ δεν επηρεάζουν την γενική εκτέλεση της συγκεκριμένης επίθεσης. Παρόλα αυτά, υιοθετούμε την υπόθεση ότι εάν αποσταλεί ένα διεφθαρμένο μήνυμα από τον εισβολέα που δεν πληρεί την γενική 'μορφολογία' του προς παραλαβή μηνύματος, ο παραλήπτης θα το

απορρίψει εισερχόμενος σε μια κατάσταση λάθους-τερματισμού (fail-stop state), χωρίς να ολοκληρώσει την σύνοδο του πρωτοκόλλου. Η υπόθεση αυτή αναπαριστά την αναμενόμενη συμπεριφορά ενός μοντέλου-αντίγραφου του πρωτοκόλλου και καθ' επέκταση των συμμετεχόντων αυτού.

Η ενέργεια επίθεσης A2, όταν είναι εφικτή, αλλάζει ένα υποκλεμμένο μήνυμα με την αντικατάστασή του με ένα άλλο μήνυμα η μέρος αυτού, προερχόμενο από την γνώση του εισβολέα $I_{knowledge}$.. Αυτό είναι δυνατό μόνο όταν η τιμή του $p(msg_{\alpha,b})^{Encryption}$ είναι 0 ή 1. Εάν η τιμή είναι $p(msg_{\alpha,b})^{Encryption}$=2 και ο εισβολέας δεν έχει στην γνώση του το $I_{knowledge}$ το σωστό κλειδί αποκρυπτογράφησης (μετά από αναζήτηση που διεξάγει), συμπεραίνει ότι τα περιεχόμενα του μηνύματος με βάση τις αρχικές προδιαγραφές μοντελοποίησης που ακολουθήθηκαν, δεν μπορούν να διαβαστούν (un-breakability)και έτσι η επίθεση A2, μπορεί να αφαιρεθεί από την βάση επιθέσεων του εισβολέα.

Η ενέργεια επίθεσης A3, αντικαθιστά ένα μήνυμα (ή μέρος) με ένα άλλο μήνυμα που βρίσκεται στην γνώση $I_{knowledge}$. Το παραγόμενο διεφθαρμένο μήνυμα μπορεί να γίνει αποδεκτό από την οντότητα-θύμα, μόνο εάν το μέγεθος του διεφθαρμένου, είναι το ίδιο με το μέγεθος του αναμενόμενου προς λήψη μηνύματος, αλλά και της ίδιας μορφής. Σε αυτή την περίπτωση και μόνο η οντότητα-θύμα δεν θα πέσει σε κατάσταση λάθους-τερματισμού (πριν από την επαλήθευση των εμπεριεχομένων πληροφοριών του μηνύματος). Κάτι τέτοιο μπορεί να ελέχθη από συγκεκριμένες ενέργειες συγκρίσεων των μεταδεδομένων που δημιούργησε ο εισβολέας για τα μηνύματα που βρίσκονται στη $I_{knowledge}$. Οι επιθέσεις παραβίασης τύπου μηνυμάτων (Type Flaws) με μερικώς διεφθαρμένα μηνύματα, είναι δυνατόν να συμβούν μόνο όταν η τιμή του $p(msg_{\alpha,b})^{Encryption}$ δεν είναι ίση με 2, που σημαίνει ότι το μήνυμα πρέπει να είναι μερικώς αναγνωστέο. Εναλλακτικά, με βάση το [43], μια *TFLAW* επίθεση είναι δυνατόν να συμβεί, σε μια σύνοδο όπου ένας έντιμος πράκτορας πέφτει στην λανθασμένη διαδικασία διερμηνείας ενός μηνύματος , σε κάποιο βήμα του ίδιου πρωτοκόλλου. Επίσης, το συμβάν της επίθεσης παραβίασης τύπου μηνύματος μπορεί να ενεργοποιηθεί και σε περίπτωση που το μήνυμα είναι πλήρως κρυπτογραφημένο. Σε τέτοια περίπτωση, ο αναλυτής παραμετροποιεί τις ορισμένες συνθήκες σύγκρισης των κανόνων διερεύνησης του μηνύματος, οι οποίες ορίζονται παρακάτω.

Η ενέργεια επίθεσης A4, εκκινεί μια καινούργια σύνοδο του υπό εξέταση πρωτοκόλλου με την επαναχρησιμοποίηση προηγούμενων υποκλεμμένων μηνυμάτων με κάποια τιμή μεταδεδομένου $p(a,b)$ όπου a =1. Τέλος η επίθεση A5, εκκινεί επίσης μια καινούργια σύνοδο πρωτοκόλλου, ή χειραγωγεί μια τρέχουσα σύνοδο, με την επαναχρησιμοποίηση μηνυμάτων που υπέκλεψε ο εισβολέας από άλλη τρέχουσα σύνοδο. Και οι δύο αυτές οι επιθέσεις, δεν επηρεάζονται από την μορφή κρυπτογράφησης μηνύματος, και άρα δεν αναλύονται οι τιμές των μεταδεδομένων που έχουν να κάνουν με υποκλεμμένα μηνύματα κρυπτογραφημένα μερικώς ή όχι.

Πίνακας 5.4.2 Κανόνες για τον έλεγχο αποτελεσματικότητας ενεργειών επιθέσεων για το μοντέλο ΕΔΜ

Μεταδεδομένα	**Συνθήκες Ενεργοποίησης**		**Ενέργειες Επιθέσεων**
Αναγνωσιμότητα $p(msg_{a,b})^{Encryption}$	$p(msg_{a,b})^{Encryption}= 2$		A1, A4, A5
	$p(msg_{a,b})^{Encryption}= 1$		A1, A2, A4, A5
	$p(msg_{a,b})^{Encryption}\neq 2$ και $\exists m \in Msgs: exists(m, msg_{a,b}) =$ `true` και $\exists amsg \in AMsgs \cap I_{knowledge}: p(amsg)^{Size}=p(m)^{Size}$		A3
	$p(msg_{a,b})^{Encryption}= 0$		A1, A2, A4, A5
Μέγεθος $s_1 = p(msg_{a,b})^{Size}$ και $s_2 = p(msg_{c,d})^{Size}$	$s_1 = s_2$ και $a < c$	$b = d$	A3

Ο πίνακας 5.4.2 παρουσιάζει τους κανόνες διερεύνησης μηνύματος (MI rules) που ισχύουν στην παρούσα έκδοση του εισβολέα ΕΔΜ. Οι συνθήκες ενεργοποίησης ορίζονται για τις συγκρίσεις που επενεργεί ο εισβολέας στα δημιουργημένα μεταδεδομένα, με σκοπό να αποφασίσει για το ποιες επιθέσεις μπορούν να είναι αποτελεσματικές στην ανάλυση και ποιες όχι. Ενέργειες επιθέσεων που σε όλα τα βήματα του πρωτοκόλλου δεν πληρούν τις απαραίτητες συνθήκες ενεργοποίησής τους, προτείνονται από τον αλγόριθμο MI προς αφαίρεση, επηρεάζοντας με αυτό τον τρόπο θετικά στον παραγόμενο χώρο καταστάσεων του μοντέλου.

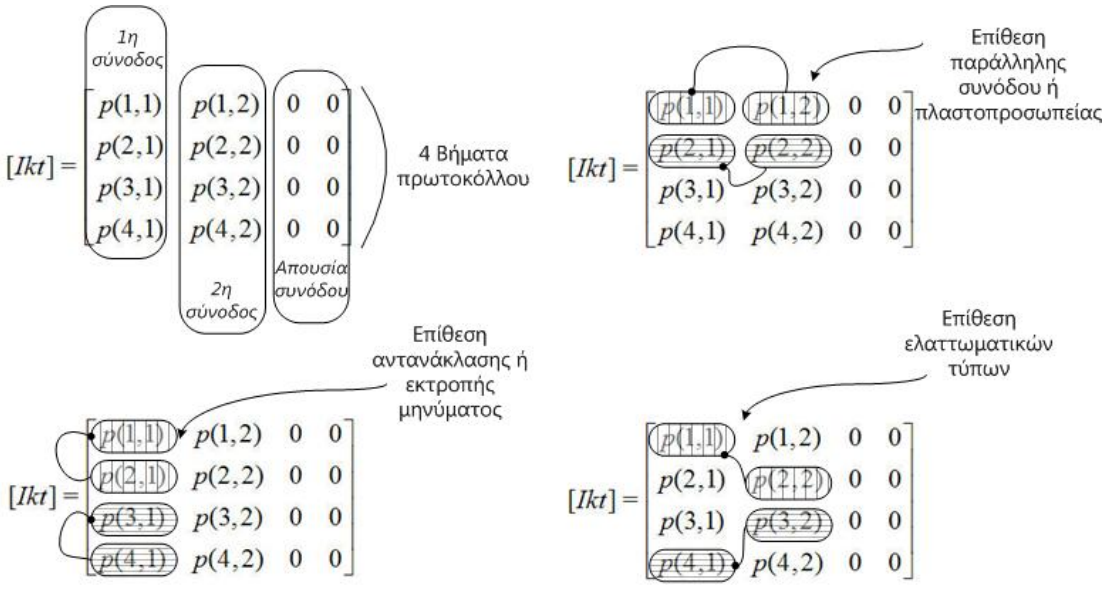
Η υποσυνάρτηση μεταδεδομένων $p(msg_{a,b})^{Encryption}$ παίζει έναν σημαντικό ρόλο σε αυτή την ανάλυση, αφού οι τιμές τις είναι αυτές που καθορίζουν εάν ένα υποκλεμμένο μήνυμα $msg_{a,b}$ μπορεί να διαβαστεί ή όχι. Όταν το $msg_{a,b}$ είναι πλήρως κρυπτογραφημένο ($p(msg_{a,b})^{Encryption}$=2), το μοντέλο του εισβολέα ελέγχει την γνώση του $I_{knowledge}$ εάν κατέχει το σωστό κλειδί

αποκρυπτογράφησης. Εάν βρεθεί το κλειδί τότε το μήνυμα μαρκάρεται ως μη κρυπτογραφημένο ανανεώνοντας την τιμή του μεταδεδομένου με $p(msg_{a,b})^{Encryption}$=0, καταγράφοντας την τελική αυτή τιμή στο *p(a,b)*. Εάν είναι δυνατή η ανάγνωση μόνο ενός μέρους του μηνύματος $msg_{a,b}$ τότε η τιμή θα είναι $p(msg_{a,b})^{Encryption}$=1, δηλαδή το $msg_{a,b}$ θα είναι μερικώς κρυπτογραφημένο. Οι συνθήκες αυτές θα αποτελούν την αφορμή ενεργοποίησης των ενεργειών επιθέσεων A1, A2, A4 και A5. Επιπρόσθετα, η πιθανότητα να αντικατασταθεί ένα μέρος του μηνύματος, έστω *m*, με ένα ατομικό μήνυμα *amsg* από την γνώση $I_{knowledge}$ απαιτεί ίσες τιμές μεταδεδομένων για το $p(amsg)^{Size}$ και $p(m)^{Size}$. Η συνθήκη ενεργοποίησης αυτή ουσιαστικά είναι προαπαιτούμενη για να θέσει μια έντιμη οντότητα υπό την επήρεια της επίθεσης A3, ωθώντας τον να εκλάβει το μήνυμα που δέχεται από τον εισβολέα με διαφορετικό τρόπο από τον αναμενόμενο.

Στις περισσότερες περιπτώσεις, στα πρωτόκολλα που εξετάζονται, η τιμή του $p(msg_{a,b})^{Encryption}$ θα είναι ίση με 2, κάτι το οποίο αυτόματα θα αποτρέψει την επιτυχία της επίθεσης A2, δηλαδή της παραβίασης της ακεραιότητας του μηνύματος. Όπως έχει προαναφερθεί, στις επιθέσεις παραβίασης των τύπων του μηνύματος (Type flaw attacks) όπου το υποκλεμμένο μήνυμα αντικαθίσταται στην ολότητά του, δεν είναι απαραίτητη η επιπλέον πληροφορία για την κρυπτογράφηση που μπορεί να υπόκειται ή όχι το συγκεκριμένο μήνυμα. Εάν το αναμενόμενο μήνυμα έχει το ίδιο μέγεθος με ένα υποκλεμμένο μήνυμα που βρίσκεται στην γνώση του εισβολέα από ένα προηγούμενο βήμα (ή σύνοδο) του πρωτοκόλλου, τότε είναι δυνατή από τον εισβολέα να εξαπολύσει μια επίθεση A3. Στην τελευταία στήλη του πίνακα 5.4.2, παρέχονται οι απαραίτητες πληροφορίες για τις συνθήκες ενεργοποίησης των διαθέσιμων ενεργειών επιθέσεων του εισβολέα. Για μια πιο πλήρη περιγραφή των υλοποιημένων ενεργειών επιθέσεων στου πίνακα 5.4.2, ο αναγνώστης μπορεί να αποταθεί στο προηγούμενο κεφάλαιο, της περιγραφής του εισβολέα ΕΠΕ ή στο [6].

Για τις ενέργειες επιθέσεων που αναφέρονται στον πίνακα 5.4.1, το μοντέλο ΕΔΜ συγκρίνει τα μεταδεδομένα που έχει δημιουργήσει κατά την υποκλοπή των μηνυμάτων που ανταλλάσουν οι συμμετέχουσες οντότητες του πρωτοκόλλου, προσπαθώντας να διερευνήσει την αποτελεσματικότητα ή όχι

των διαθέσιμων ενεργειών επιθέσεων. Παραδείγματα σύγκρισης των για ένα υποτιθέμενο πρωτόκολλο τεσσάρων (4) διακριτών βημάτων απεικονίζονται στην εικόνα 5.4.4, για τις περισσότερες ενέργειες επιθέσεων του εισβολέα ΕΔΜ, στην παρούσα έκδοση. Όταν οι τιμές $p(1,1) \cong p(1,2)$ και την ίδια στιγμή ισχύει μια από τις συνθήκες ενεργοποίησης του πίνακα 5.4.2, όπου ενεργοποιούνται οι επιθέσεις Α4 και/ή Α5, τότε το μοντέλο ΕΔΜ μπορεί να εκκινήσει μια καινούργια σύνοδο του πρωτοκόλλου, με σκοπό την προσπάθεια επιτυχίας μιας επίθεσης πλαστοπροσωπίας ή μιας επίθεσης παράλληλης συνόδου. Όταν οι τιμές $p(2,1) \cong p(2,2)$ και ισχύουν οι συνθήκες του πίνακα 5.4.2 που ενεργοποιούν την επίθεση Α5, τότε είναι πιθανό ο εισβολέας να χειραγωγήσει παράλληλες συνόδους του πρωτοκόλλου, αποτρέποντας εάν μπορέσει τις ιδιότητες ορθότητας του πρωτοκόλλου. Κοιτώντας την πιο γενική περίπτωση σύγκρισης των μεταδεδομένων, όπου θα ισχύει $p(a,b) \cong p(c,b)$ μαζί με την απαραίτητη συνθήκη του πίνακα 2 αναφερόμενη στην επίθεση Α1, ο εισβολέας ΕΔΜ θα μπορεί (για οποιαδήποτε βήμα *a,c* στις συνόδους *b,d*) να εξαπολύσει μια από τις επιθέσεις επανάληψης μηνύματος. Τέλος όταν $p(a,b) \cong p(c,d)$ για δύο μηνύματα διαφορετικών συνόδων του πρωτοκόλλου, και το τελευταίο υποκλεμμένο μήνυμα είναι (μερικώς) αναγνώσιμο, με προϋπόθεση ενεργοποίησης των συνθηκών του πίνακα 5.4.2 για την επίθεση Α3, ο εισβολέας τίθεται ικανός να εκτελέσει μια επίθεση παραβίασης τύπων του μηνύματος.



Εικόνα 5.4.4 Συγκρίσεις μεταδεδομένων για τιμές του πίνακα [*Ikt*] για την ανίχνευση πιθανών ενεργειών επιθέσεων σε κάθε βήμα του πρωτοκόλλου

Στις παραπάνω καταφατικές περιπτώσεις ισότητας σύγκρισης των μεταδεδομένων, ο εισβολέας ΕΔΜ θα είναι σε θέση να πυροδοτήσει τις επιθέσεις που έχουν περάσει τις συνθήκες ενεργοποίησης του πίνακα 5.4.2. Ο εισβολέας μπορεί (πιθανόν) να έχει διαφθείρει ένα υποκλεμμένο μήνυμα $msg \in Msgs$ βασισμένο στη γνώση $I_{knowledge}$, δημιουργώντας ένα καινούργιο μήνυμα $msg' \in Msgs$. Η επόμενη ενέργεια του εισβολέα θα είναι είτε η εκτέλεση της ενέργειας $send(I, v, msg')$ ή $send(I, v, \{msg'\}_{k'})$ για ένα $k' \in I_{knowledge}$ έτσι ώστε $v \in is\_key\_of(k')$, που σημαίνει ότι ο v θα είναι ο κάτοχος του k'. Η ενέργεια επίθεσης αυτή θα επιτύχει εάν παρουσιαστεί μια καθολική κατάσταση στον παραγόμενο χώρο των καταστάσεων κατά την διάρκεια της επαλήθευσης του συνολικού μοντέλου, μετά από το συμβάν καταστάσεων για την ενέργεια $receive(v, I, msg')$ ή $receive(v, I, \{msg'\}_{k'})$, όπου θα υπάρξει ένα ατομικό μήνυμα $amsg$, έτσι ώστε:

$$exists(amsg, rcvd^{v_{noSes}}_{\max(i)}) = \texttt{true}, 1 \leq noSes \leq \#Ses_v,$$

και για δύο σύνολα Set_e και Set_f της ασύνδετης ένωσης $Amsgs$, $\text{amsg} \in \text{Set}_\text{e} \cap \text{Set}_\text{f}$ όπου $i \geq 1$ αναπαριστά τους όρους της συναλύσωσης των ακολουθιών των μηνυμάτων που λαμβάνονται από τον πράκτορα v στην διάρκεια της συνόδου $noSes$. Έτσι, ένα ατομικό μήνυμα το οποίο προορίζονταν αρχικά να έχει μια μορφή συγκεκριμένου τύπου (για παράδειγμα ένας τυχαίος αριθμός *Nonce*) μεταφράζεται σαν να έχει λάβει ένα μήνυμα άλλου τύπου (όπως ένα αριθμητικό κλειδί ή δεδομένα), που σημαίνει ότι μια παραβίαση τύπου μηνύματος είναι εφικτή, ακόμη και στην περίπτωση που κάτι τέτοιο δεν θα οδηγήσει σε άμεση παραβίαση της συνολικής ασφάλειας του πρωτοκόλλου. Κατά την διάρκεια του ελέγχου μοντέλων, το μοντέλο ΕΔΜ εφαρμόζει όλες τις επιθέσεις που υπάρχουν υλοποιημένες στο σώμα του, σε όλα τα βήματα του πρωτοκόλλου, μετά από την αφαίρεση των όποιων αχρείαστων επιθέσεων μετά το πέρας της πρώτης φάσης, της προκαταρκτικής προσομοίωσης με τον εισβολέα ΕΔΜ. Οι έντιμοι πράκτορες $\texttt{ag} \in Agents$ είτε με την αποδοχή είτε με την απόρριψη διεφθαρμένων μηνυμάτων βασισμένα στην υλοποιήσιμη λογική του πρωτοκόλλου.

Πέρα από τις δύο υποσυναρτήσεις που έχουν υλοποιηθεί στην παρούσα φάση του μοντέλου ΕΔΜ, είναι δυνατό να επεκταθούν τα μεταδεδομένα με άλλου τύπου πληροφορίες που μπορούν φανούν χρήσιμες για την αποτροπή ή

όχι περαιτέρω επιθέσεων, όπως για παράδειγμα χρονοσφραγίδες των υποκλεμμένων μηνυμάτων ή καταγραφή του αποστολέα του μηνύματος, για έλεγχο διαφύλαξης της ανωνυμίας της κάθε οντότητας. Κάτι τέτοιο θα επέτρεπε την βελτίωση της συμπεριφοράς του μοντέλου ΕΔΜ [21][90][100], βασιζόμενοι σε γνωστές αρχές ιδιοτήτων ασφαλείας .

5.5 Έλεγχος μοντέλων με τον εισβολέα ΕΔΜ

Η ενότητα αυτή παρουσιάζει την πρακτική εφαρμογή της όλης θεωρίας που παρουσιάστηκε για τον εισβολέα ΕΔΜ με τον αυτόματο ελεγκτή μοντέλων SPIN. Στόχοι της όλης ανάλυσης μετά την επιλογή ενός πρωτοκόλλου ασφαλείας (NSPK) είναι η παρουσίαση συγκριτικών αποτελεσμάτων του γνωστού εισβολέα Dolev-Yao και του μοντέλου εισβολέα ΕΔΜ, για τον παραγόμενο χώρο καταστάσεων, κατά την διάρκεια της εξαντλητικής επαλήθευσης. Επιπλέον πρέπει, να αποδειχθεί, παρόλο την μείωση του χώρου των καταστάσεων, και η αποτελεσματικότητα του εισβολέα στον να βρίσκει λάθη σε πρωτόκολλα ασφαλείας. Για την όλη ανάλυση επιλέχθηκε το γνωστό πρωτόκολλο ασύμμετρης κρυπτογράφησης των Needham και Schroeder (Needham-Schroeder Public Key protocol, NSPK) [74]. Έτσι σκοπεύουμε:

- Στην παροχή ενδεικτικών αποτελεσμάτων για τον χώρο των καταστάσεων που μπορεί να παραχθεί από τα δύο παραπάνω μοντέλα εισβολέων, σε συνδυασμό με τεχνικές όπως αυτή της μερικής αναδιατεταγμένης μείωσης (partial order reduction) του χώρου των καταστάσεων ή άλλες προσεγγίσεις που μπορούν να ακολουθηθούν με τον αυτόματο ελεγκτή μοντέλων SPIN
- Στην παροχή ενδεικτικών αποτελεσμάτων από ποικίλες εκδόσεις του μοντέλου ΕΔΜ αφαιρώντας σταδιακά επιθέσεις από την δομή του, έτσι ώστε να καταγραφεί η μεταβολή που θα έχουμε στον παραγόμενο χώρο των καταστάσεων κατά την διάρκεια της επαλήθευσης

Σε πρώτη φάση περιγράφουμε τα αποτελέσματα που πάρθηκαν από τα πειράματα που διενεργήθηκαν με τον συνδυασμό των διαφόρων τεχνικών μείωσης του χώρου των καταστάσεων. Μια σημαντική παρατήρηση είναι ότι τα αποτελέσματα αυτά δεν μπορούν να συγκριθούν μεταξύ τους, μιας και καθένα

από αυτά αφορούν διαφορετικές χρησιμοποιούμενες τεχνικές. Ο αναλυτής έχει με βάση αυτά, την ευχέρεια επιλογής της τεχνικής εκείνης, όπου με δεδομένου της φύσης του συστήματος που σκοπεύει να ελέγξει, μπορεί να επιλέξει την κατάλληλη τεχνική μείωσης.

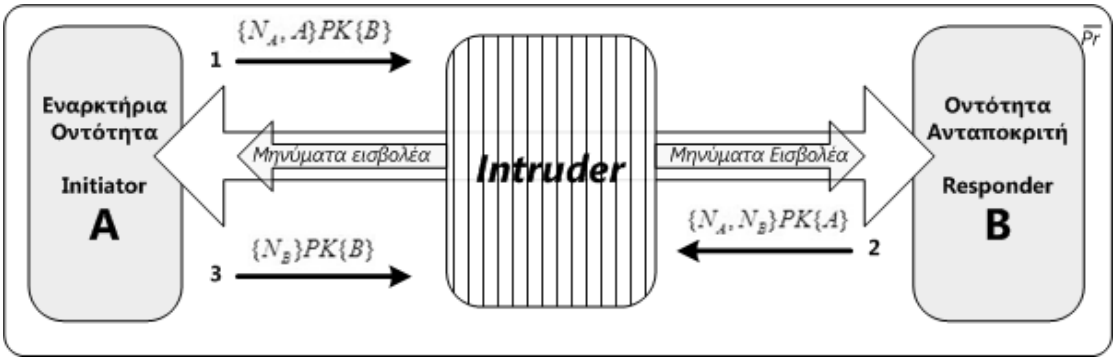
Ο τελικός παραγόμενος χώρος των καταστάσεων και οι βελτιώσεις που εμπεριέχονται στο μοντέλο του αυτόματου ελεγκτή μοντέλων, εξαρτώνται από την παραλληλία των διεργασιών που έχουν οριστεί μέσα στο μοντέλο του πρωτοκόλλου, καθώς και της μοντελοποιούμενης κρυπτογράφησης των μηνυμάτων που ανταλλάσσονται. Ο συνδυασμός της μεθόδου του μοντέλου ΕΔΜ μπορεί να μην είναι δυνατή παρόλα αυτά, με άλλες τεχνικές όπου δεν συγκαταλέγονται στην υπάρχουσα έκδοση του ελεγκτή μοντέλων SPIN. Για τον λόγο αυτό, όλα τα αποτελέσματα με τις τεχνικές που χρησιμοποιήθηκαν, παρουσιάζουν έναν επεξηγηματικό χαρακτήρα [98], δηλαδή αποτελούν μελέτες που δίνουν την ευκαιρία για την διερεύνησης γεγονότων τύπου αιτίας-επίδρασης (*cause-and effect*). Το σύνολο των αποτελεσμάτων αυτών, ειδικά κατά την επαλήθευση πρωτοκόλλων ασφαλείας, επιλέγεται προς χάριν τεκμηρίωσης της προτεινόμενης μεθόδου, να διεξαχθεί σε ένα γνωστό πρωτόκολλο ασφαλείας (NSPK), όπου παρόμοιες ερευνητικές απόπειρες, έχουν επαληθεύσει το λάθος ασφαλείας που περιέχει .

5.5.1 Το πρωτόκολλο ασύμμετρης κρυπτογράφησης των Needham και Schroeder

Το πρωτόκολλο ασφαλείας ασύμμετρης κρυπτογράφησης που προτάθηκε από τους Needham και Schroeder, NSPK, στο [74], στοχεύει στην εγκαθίδρυση μιας ασφαλούς συνεδρίας, αμοιβαίας αυθεντικοποίησης, μεταξύ δύο οντοτήτων-πρακτόρων, έστω μιας εναρκτήριας οντότητας (Initiator) και μιας οντότητας ανταποκριτή (Responder). Από την ονομασία του πρωτοκόλλου γίνεται κατανοητή ότι θα γίνεται χρήση κρυπτογράφησης δημόσιου κλειδιού προς την σωστή διατήρηση των ιδιοτήτων αυθεντικοποίησης.

Η απλή έκδοση του πρωτοκόλλου ασφαλείας NSPK, όπως φαίνεται στο σχήμα 5.5.1, περιλαμβάνει τρία (3) βασικά βήματα, όπου σε καθένα οι συμμετέχοντες στο πρωτόκολλο A (Initiator) και B (Responder) ανταλλάσσουν μηνύματα τα οποία περιέχουν τις ταυτότητές τους (identities) τυχαίους

παραγόμενους αριθμούς (Nonces N_A, N_B), κρυπτογραφημένα με τα δημόσια κλειδιά τους *PK{A}* και *PK{B}*.



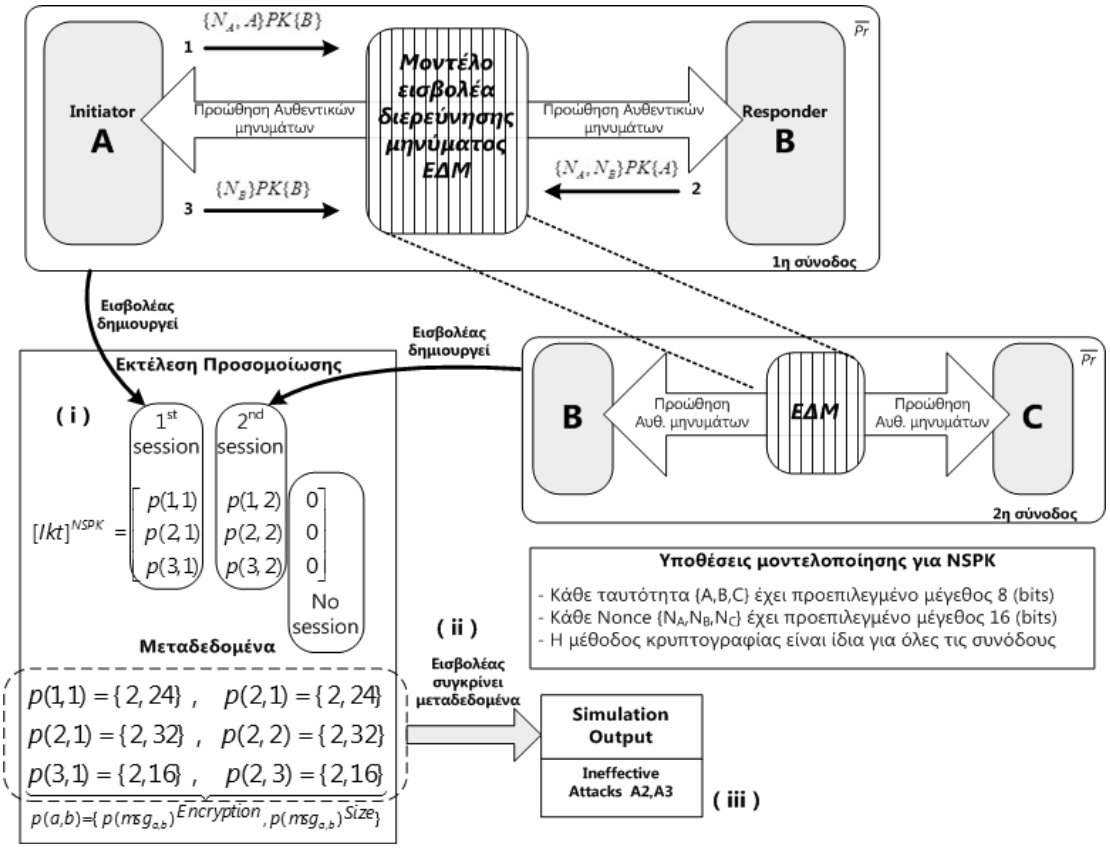
Εικόνα 5.5.1 Τα βήματα πρωτοκόλλου για το Needham Schroeder

Μαζί με τους πράκτορες Α και Β που φαίνονται στην εικόνα 5.5.1, το ολοκληρωμένο μοντέλο θα περιλαμβάνει τον εισβολέα ΕΔΜ, ο οποίος και θα είναι ο απόλυτος κυρίαρχος του επικοινωνιακού καναλιού μεταξύ των έντιμων οντοτήτων, ως ενδιάμεση οντότητα (man-in-the-middle entity). Τα αποτελέσματα της όλης ανάλυσης, όπως αναφέρθηκε και παραπάνω, αφορούν την διεξαγωγή δύο πειραμάτων αυτόματου ελέγχου μοντέλων για το πρωτόκολλο NSPK. Το ένα θα αφορά το γενικό μοντέλο εισβολέα των Dolev και Yao, αποτελούμενο από τους κανόνες παραγωγής μηνυμάτων όπως περιγράφθηκαν φορμαλιστικά σε προηγούμενη παράγραφο αυτού του κεφαλαίου, και του προτεινόμενου μοντέλου ΕΔΜ. Και στα δύο μοντέλα των εισβολέων δίνεται η δυνατότητα εκκίνησης καινούργιων συνόδων του υπό εξέταση πρωτοκόλλου, με την προϋπόθεση ότι οι εισβολείς έχουν προηγουμένως υποκλέψει μηνύματα του ίδιου του πρωτοκόλλου, στην παρούσα ή σε διαφορετική σύνοδο που αυτό εξετάζεται. Οι έντιμες οντότητες Α και Β μοντελοποιούνται ως διεργασίες λανθασμένου-τερματισμού (fail-stop) έτσι ώστε σε περίπτωση όπου ληφθεί ένα μη αναμενόμενο –με βάση τις προδιαγραφές του πρωτοκόλλου- μήνυμα, η οντότητα να απορρίπτει την όλη σύνοδο του πρωτοκόλλου, ακόμα και σε περίπτωση όπου δεν έχουμε παραβίαση των όποιων ιδιοτήτων ασφαλείας. Οι ιδιότητες ασφαλείας που εγγυάται το συγκεκριμένο πρωτόκολλο του NSPK ορίζονται στο περιβάλλον του αυτόματου ελεγκτή μοντέλων SPIN ως μη αποδεκτές καταστάσεις τερματισμού (invalid end-states).

5.5.2 Έλεγχος μοντέλων του NSPK με το μοντέλο ΕΔΜ

Στην πρώτη φάση της προσομοίωσης του μοντέλου του NSPK μαζί με τον εισβολέα ΕΔΜ, για δύο συνόδους του πρωτοκόλλου, όπως φαίνεται και στο σχήμα 5.5.2, το μοντέλο του εισβολέα ανιχνεύει δύο επιθέσεις, τις Α2 και Α3, οι οποίες μπορούν μετά από την ανάλυση που διεξάγει να αφαιρεθούν από το σώμα του. Ειδικότερα, ο εισβολέας ΕΔΜ τίθεται παράλληλα ως η κυρίαρχη ενδιάμεση οντότητα τόσο για την σύνοδο μεταξύ των οντοτήτων Α και Β αλλά και της συνόδου μεταξύ των Β και C.

Με την υποκλοπή ενός μηνύματος του NSPK, έστω $msg_{a,b}$, το μοντέλο του ΕΔΜ δημιουργεί τις απαραίτητες τιμές των μεταδεδομένων $p(msg_{a,b})^{Encryption}$ και $p(msg_{a,b})^{Size}$, τα οποία και καταγράφονται στον πίνακα [*Ikt*] (Εικόνα 5.5.2i). Από την στιγμή που ο εισβολέας ΕΔΜ προωθεί τα υποκλεμμένα μηνύματα στους αρχικούς προορισμούς τους, μετά την ανάλυση που υπόκεινται αυτά από τις υποσυναρτήσεις των μεταδεδομένων, και οι δυο οι σύνοδοι του πρωτοκόλλου ολοκληρώνονται με επιτυχία. Σε αυτό το σημείο γίνεται αντιληπτό, ότι όλα τα μηνύματα του πρωτοκόλλου είναι πλήρως κρυπτογραφημένα. Επιπρόσθετα αφού το αντίστοιχο κλειδί αποκρυπτογράφησης δεν θα βρεθεί ποτέ στην γνώση $I_{knowledge}$, του εισβολέα, θα περιμένουμε ότι για όλα τα μηνύματα που ανταλλάσσονται, οι τιμές των μεταδεδομένων για την υποσυνάρτηση αναγνωσιμότητας αυτών θα είναι $p(msg_{a,b})^{Encryption}$ =2. Παρατηρείται επίσης ότι το μέγεθος των μηνυμάτων που υπολογίζεται (μετά από την λήψη συμβολικών υποθέσεων για τα επιμέρους μεγέθη πληροφορίας που εμπεριέχουν αυτά) για την τιμή $p(msg_{a,b})^{Size}$, όπως φαίνεται στην εικόνα 6, αναπαριστούν το άθροισμα των μεγεθών των διακριτών πληροφοριών που βρίσκονται στο σώμα του μηνύματος (τυχαίοι αριθμοί και ταυτότητες οντοτήτων).



Εικόνα 5.5.2 Προκαταρκτική εκτέλεση προσομοίωσης βάση του μοντέλου ΕΔΜ: ο εισβολέας (i) δημιουργεί τον πίνακα [*Ikt*], (ii) συγκρίνει τα μεταδεδομένα και (iii) προτείνει την αφαίρεση των ενεργειών επιθέσεων A2 και A3

Το μοντέλο του ΕΔΜ επιχειρεί τις απαιτούμενες συγκρίσεις των δημιουργημένων μεταδεδομένων (Εικόνα 5.5.2ii) όπως περιγράφθηκε και στην προηγούμενη παράγραφο, με βάση τους κανόνες διερεύνησης μηνυμάτων που ορίστηκαν στον πίνακα 2. Στο τέλος, το μοντέλο του εισβολέα επιστρέφει τις όποιες αποφάσεις του, με συγκεκριμένο μήνυμα στο αποτέλεσμα της προσομοίωσης (Εικόνα 6iii). Από τη στιγμή που $p(msg_{a,b})^{Encryption} = 2$ σε όλα τα βήματα του πρωτοκόλλου η επίθεση παραβίασης της ακεραιότητας του μηνύματος (A2) προτείνεται να αφαιρεθεί. Επίσης εξαιτίας του $p(msg_{a,b})^{Encryption} = 2$ για όλα τα ανταλλασσόμενα μηνύματα και την ίδια στιγμή, δεν υπάρχουν μηνύματα με το ίδιο μέγεθος στην ίδια σύνοδο του πρωτοκόλλου, το μοντέλο του ΕΔΜ προτείνει την αφαίρεση της επίθεσης παραβίασης τύπων του μηνύματος (A3).

Η εικόνα 5.5.3 παρέχει τα αποτελέσματα της επαλήθευσης του NSPK κατά την διάρκεια του αυτόματου ελέγχου μοντέλων με το εργαλείο SPIN, όπως αυτά παράχθηκαν από το χώρο καταστάσεων που συνηγόρησε το όλο μοντέλο. Σε

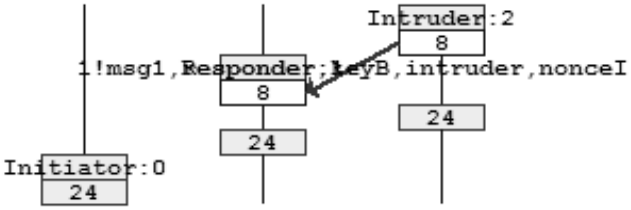
αυτή την επαλήθευση, χρησιμοποιείται πρώτα το μοντέλο ΕΔΜ μαζί με την τεχνική της αναδιατεταγμένης μείωσης του χώρου των καταστάσεων προσπαθώντας παράλληλα να αιχμαλωτίσει πιθανές παραβιάσεις ασφαλείας. Η επαλήθευση, εντοπίζει σε ένα βάθος 25 (του παραγόμενου δένδρου) μια κατάσταση μη αποδεκτού τερματισμού (invalid end-state) με βάση τις ιδιότητες ασφαλείας που ορίστηκαν εξ' αρχής στο μοντέλο. Ακολουθώντας το ίχνος του αντι-παραδείγματος για να οδηγηθούμε στο λάθος που βρήκε ο ελεγκτής μοντέλων, δημιουργείται το διάγραμμα ακολουθίας μηνυμάτων MSC (Message Sequence Chart) της εικόνας 5.5.4. Η προσέγγιση της μη αποδεκτής κατάστασης (και ουσιαστικά της παραβίασης ασφαλείας του πρωτοκόλλου), αντιστοιχεί στην περίπτωση εκείνη όπου ο πράκτορας Β που ενεργεί ως οντότητα ανταποκριτή, δέχεται ένα διεφθαρμένο μήνυμα του εισβολέα για το NSPK, το οποίο και τον αναγκάζει να εκκινήσει μια καινούργια σύνοδο του πρωτοκόλλου (επίθεση πλαστοπροσωπίας).

```
pan: invalid end state (at depth 25)
pan: wrote pan_in.trail

(Spin Version 5.1.6 -- 9 May 2008)
Warning: Search not completed
        + Partial Order Reduction

Full statespace search for:
        never claim             - (not selected)
        assertion violations    - (disabled by -A flag)
        cycle checks            - (disabled by -DSAFETY)
        invalid end states      +

State-vector 162 byte, depth reached 24, errors: 1
     264 states, stored
     883 states, matched
    1147 transitions (= stored+matched)
       11 atomic steps
hash conflicts:     1493 (resolved)

    2.326       memory usage (Mbyte)

unreached in proctype Agent_A
        line 33, "pan_in", state 8, "-end-"
        (1 of 12 states)
unreached in proctype Agent_B
        (0 of 10 states)
unreached in proctype MI_intruder
        line 54, "pan_in", state 8, "-end-"
        (1 of 86 states)
```

Εικόνα 5.5.3 Αποτελέσματα επαλήθευσης για το NSPK και το μοντέλο ΕΔΜ με την μη αποδεκτή κατάσταση τερματισμού να εντοπίζεται στο βάθος 25

Το χρησιμοποιούμενο μήνυμα δημιουργείται από το μοντέλο ΕΔΜ, θέτοντας τον εισβολέα να παίζει τον ρόλο της εναρκτήριας οντότητας, με την προϋπόθεση ότι η μορφή του μηνύματος που αποστέλλεται να είναι σε

συμφωνία με την γενική μορφή μηνύματος του πρωτοκόλλου NSPK. Η επίθεση αυτή της μη αποδεκτής αυθεντικοποίησης επίθεσης εντοπίζεται και από τον εισβολέα DY με την σημαντική διαφορά, ότι το λάθος βρίσκεται σε βάθος 48 [12].



Εικόνα 5.5.4 Καθοδηγούμενη προσομοίωση από τον ελεγκτή μοντέλων SPIN αντικατοπτρίζοντας την ανιχνεύσιμη επίθεση πλαστοπροσωπίας για το NSPK

5.5.3 Μείωση του χώρου των καταστάσεων με το μοντέλο ΕΔΜ

Για την εφικτή σύγκριση μεταξύ των δύο χρησιμοποιηθέντων μοντέλων εισβολέων, του ΕΔΜ και του DY, για το πρωτόκολλο NSPK, κρίνεται απαραίτητο [98], να περιγραφθούν οι λειτουργίες του ελεγκτή μοντέλων SPIN και των τεχνικών του, προσπαθώντας να μελετήσουμε τον συνδυασμό των παραπάνω μοντέλων, σε συνδυασμό με γνωστές τεχνικές μείωσης του χώρου των καταστάσεων. Η παρακάτω περιγραφή παρέχει την δυνατότητα να ορισθεί επακριβώς το εύρος, οι συνθήκες και η αποτελεσματικότητα του πειράματος, δίνοντας στον αναλυτή την δυνατότητα να καταλάβει την σχέση αιτίας-αποτελέσματος (*cause-effect*) της όλης ερευνητικής πρότασης, το οποίο αποτελεί και πρωταρχική προτεραιότητα της όλης μελέτης. Ο φορμαλιστικός ορισμός του πειράματος με βάση το [98], δίνεται στον πίνακα 5.5.1.

Πίνακας 5.5.1: Ορισμός του Πειράματος

Ανάλυση *(Analyze)*	Του γενικού εισβολέα DY έναντι του εισβολέα διερεύνησης μηνύματος
Για τον σκοπό *(For the purpose of)*	Εξερεύνησης συσχετίσεων αιτίας αποτελέσματος (cause and effect relationships)
Εξετάζοντας *(With respect to)*	Το μέγεθος των παραγόμενων γράφων (συνολικός αριθμός καταστάσεων) και της χρησιμοποιούμενης μνήμης
Από την πλευρά *(From the point of view of)*	Του αναλυτή στον αυτόματο έλεγχο μοντέλων
Για την μελέτη *(In the context of)*	Του πρωτοκόλλου NSPK με τα μοντέλα DY και ΕΔΜ που υλοποιήθηκαν στο SPIN

Οι ερευνητικές ερωτήσεις και οι σχετιζόμενες μετρικές που χρησιμοποιήθηκαν για την διεκπεραίωση αυτής της μελέτης είναι:

Ερώτηση 1 (Q1): Τι αντίκτυπο έχει στον παραγόμενο χώρο καταστάσεων ενός μοντέλου πρωτοκόλλου ασφαλείας, το χρησιμοποιούμενο μοντέλο εισβολέα ;

Χρησιμοποιούμενες Μετρικές: Συγκρίνουμε τον χώρο των καταστάσεων (συνολικός αριθμός των καταστάσεων και χρησιμοποιούμενη μνήμη) που παράγονται από τα δύο μοντέλα εισβολέων, για τις παρακάτω περιπτώσεις.

- Αρχικά, συγκρίνουμε το μέγεθος των πλήρων γράφων προσέγγισης, δηλαδή θέτουμε στον ελεγκτή μοντέλων να παράγει όλες τις πιθανές καταστάσεις του πρωτοκόλλου με τα μοντέλα των εισβολέων, ζητώντας του να αγνοήσει καταστάσεις παραβιάσεων της ασφάλειας. Με την προϋπόθεση αυτή, δίνονται πιο αντιπροσωπευτικά αποτελέσματα για την βελτίωση (ή όχι) του εισβολέα ΕΔΜ, μιας και η κατάσταση παραβίασης ασφαλείας που εντοπίζει το μοντέλο του ΕΔΜ, βρίσκεται σε μικτό βάθος, καθιστώντας αδύνατη την παραγωγή του συνολικού χώρου των καταστάσεων.
- Συγκρίνεται επίσης ο χώρος των καταστάσεων με την ανιχνευτική ιδιότητα του ελεγκτή μοντέλων να αιχμαλωτίζει μη αποδεκτές καταστάσεις τερματισμού. Αποδεικνύεται και περιγράφεται η μείωση του χώρου των καταστάσεων που επιτυγχάνεται με το μοντέλο του ΕΔΜ σε σχέση με αυτό του εισβολέα DY.

Ερώτηση 2 (Q2): Σε τι βαθμό επηρεάζει το μοντέλο ΕΔΜ των χώρο των καταστάσεων σε σχέση με τις διαφορετικές ενέργειες επιθέσεων που διαθέτει ;

Χρησιμοποιούμενες Μετρικές: Γίνεται σύγκριση των πλήρη παραγόμενων γράφων κατά την διάρκεια του αυτόματου ελέγχου μοντέλων για 14 διαφορετικές εκδοχές του μοντέλου ΕΔΜ.

Η προεπιλεγμένες λειτουργίες του αυτόματου ελεγκτή μοντέλων SPIN περιλαμβάνει την τεχνική της μερικής διατεταγμένης μείωσης του χώρου των καταστάσεων (πίνακας 5.5.2). Παρόλα αυτά, στοχεύοντας στην κατανόηση σχέσεων αιτίας-αποτελέσματος όσον αφορά τα χρησιμοποιούμενα μοντέλα εισβολέων στην ανάλυση του NSPK, διεξάγονται αναλύσεις του χώρου των καταστάσεων με διάφορες τεχνικές μείωσης του χώρου των καταστάσεων που έχουν καταγραφθεί στην βιβλιογραφία σήμερα. Εκτός όμως από την διαδικασία χειραγώγησης του χώρου των καταστάσεων, σημαντικό ρόλο παίζει και η μέθοδος αναζήτησης σε αυτόν για οποιεσδήποτε παραβιάσεις έχει ορίσει ο

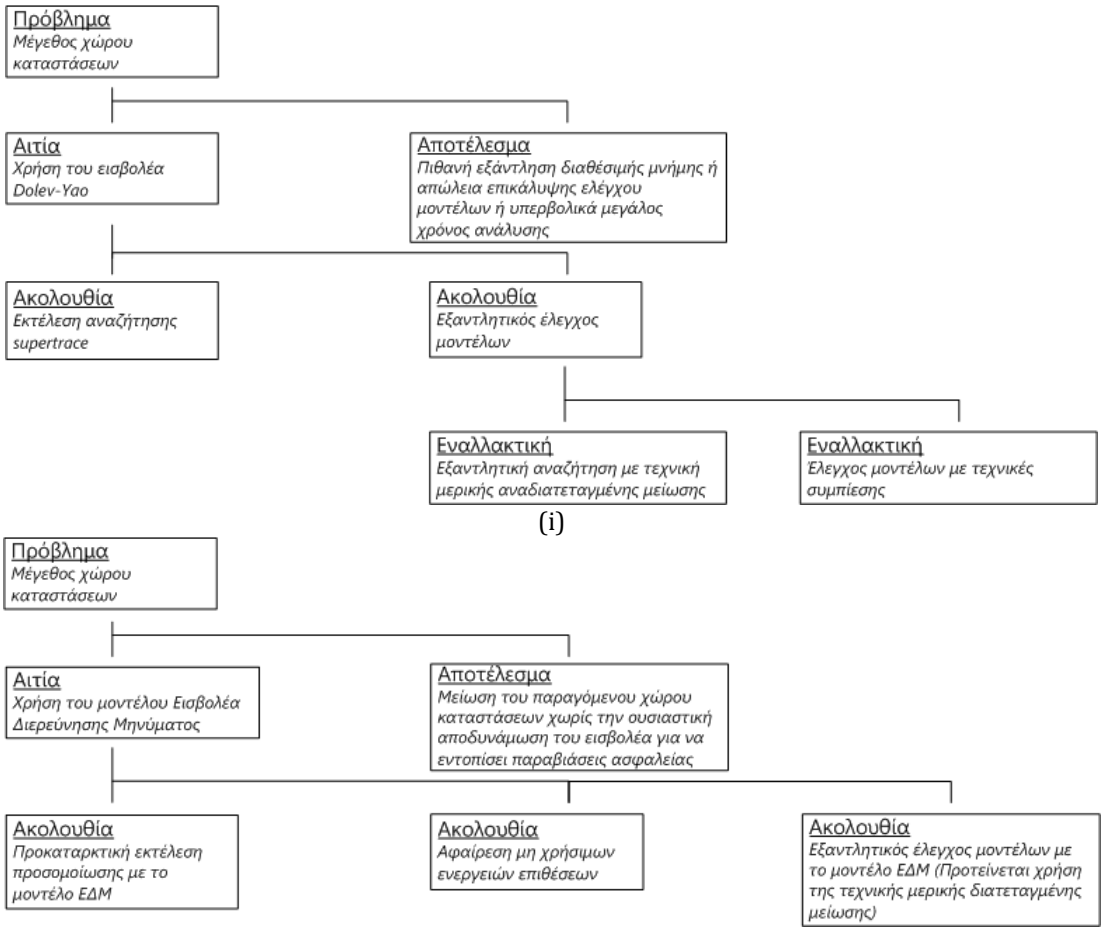
αναλυτής. Η καθεμία από τις τεχνικές αυτές περιγράφεται στον πίνακα 5.5.2. Η αναζήτηση *Supertrace* τείνει να εντοπίζει λάθη (εάν αυτά υπάρχουν) με γρήγορο τρόπο, αλλά όχι με τον πλέον παραγωγικό τρόπο, ειδικά όταν πρόκειται να εξεταστεί ένα μοντέλο, για το οποίο ο αναλυτής δεν μπορεί να κάνει καμιά εκτίμηση του προς παραγωγή χώρο των καταστάσεων. Η αναζήτηση *Hash-Compact* παρουσιάζει μεγάλη ακρίβεια κατά την διεξαγωγή της, ειδικά σε έναν εκτιμώμενο από πριν χώρο καταστάσεων.

Από τα μεγαλύτερα πλεονεκτήματα κατά την αναζήτηση στον χώρο των καταστάσεων, είναι η υλοποίηση της επιλογής αναζήτησης με την τεχνική Breadth-First (Breadth First Search), η οποία είναι η καλύτερη δυνατή για τον εντοπισμό μη αποδεκτών καταστάσεων τερματισμού και ειδικότερα για γενικά λάθη παραβιάσεων ασφαλείας. Ουσιαστικά εντοπίζει το μικρότερο εκείνο μονοπάτι –σε αντίθεση με την τεχνική DFS (Depth First Search) που επιστρέφει το μακρύτερο μονοπάτι - που οδηγεί σε λάθος, δεδομένου της αρχική κατάστασης (ρίζα του δένδρου) του μοντέλου. Η επιλογή συμπίεσης *collapse* από τη μεριά της επιτυγχάνει μείωση των συνολικών απαιτήσεων μνήμης για την διεξαγωγή της ανάλυσης στον αυτόματο έλεγχο μοντέλων, για εξαντλητικές αναζητήσεις με βάση το επισυναπτόμενο κόστος των απαιτήσεων εκτέλεσης του συστήματος που εξετάζεται.

Πίνακας 5.5.2 Μείωση του χώρου των καταστάσεων και τεχνικές εξερεύνησης στο SPIN

Λειτουργίες στο SPIN	**Περιγραφή**
Εξαντλητική αναζήτηση με μερική αναδιατεταγμένη μείωση (P.O.R.)	Βάση προεπιλογής το SPIN [46], δημιουργεί τον συνολικό χώρο των καταστάσεων και τις σχέσεις των μεταβάσεων με την μέθοδο on-the-fly εφαρμόζοντας μια DFS αναζήτηση στο χώρο. Η τεχνική της αναδιατεταγμένης μείωσης που βρίσκεται προεπιλεγμένα ενεργοποιημένη, αποφεύγει την δημιουργία καταστάσεων οι οποίες δεν μπορούν να επηρεαστούν από την παράλληλη εκτέλεση των διεργασιών του μοντέλου. Βασίζεται στις εξαρτήσεις που μπορούν να συμβούν μεταξύ συγκεκριμένων διεργασιών, προσπαθώντας να εντοπίσει τις καταστάσεις εκείνες οι οποίες είναι ανεξάρτητες.
Αναζήτηση Supertrace	Στο SPIN, για την ενεργοποίηση του γρήγορου ελέγχου των καταστάσεων, οι καταστάσεις αποθηκεύονται σε ένα πίνακα κατακερματισμού (hash table). Η αναζήτηση Supertrace ή αλλιώς bit-state-hashing αποτελεί επιλογή η οποία μειώνει την χρησιμοποιούμενη μνήμη καταναλώνοντας έναν μικρό αριθμό δυφίων για την αποθήκευση μιας κατάστασης. Με την προϋπόθεση ότι η κανονική αποθήκευση όλων των καταστάσεων είναι αδύνατη, εξαιτίας του περιορισμών που τίθενται από την διαθέσιμη μνήμη, η επιλογή αναζήτησης αυτή, μπορεί να θεωρηθεί αρκετά χρήσιμη, μιας και παρουσιάζει χαμηλή πιθανότητα επικάλυψης των καταστάσεων, μιας και όταν ανιχνευθεί μια σύγκρουση κατακερματισμού (hash collision) θα σημαίνει ότι η κατάσταση αυτή υπάρχει στον πίνακα (και άρα έχει

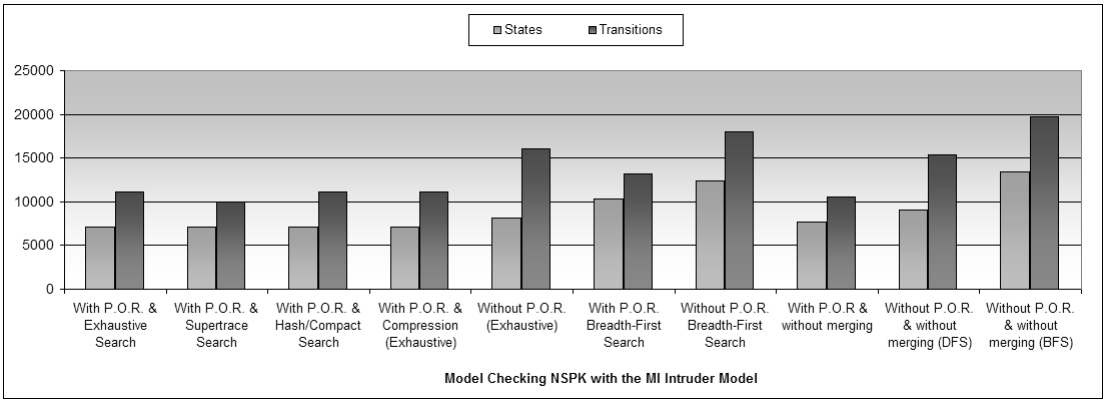
	γίνει έλεγχος αυτής), οπότε και απορρίπτεται.
Αναζήτηση Hash-Compact	Στην μέθοδο αναζήτησης Hash-Compact, μια συνάρτηση κατακερματισμού χρησιμοποιείται με σκοπό την συμπίεση της περιγραφής της κάθε κατάστασης ανεξάρτητα των πραγματικών δυφίων που χρειάζονται σε 64. Οι ακολουθίες αυτές αποθηκεύονται σε έναν πίνακα αναζήτησης όπου με αυτόν τον τρόπο το SPIN προσομοιώνει έναν πίνακα κατακερματισμού με μέγεθος μνήμης πολύ μικρότερο σε σχέση με αυτόν που θα χρειαζόταν στην πραγματικότητα.
Συμπίεση (Collapse)	Η συμπίεση (collapse) λειτουργεί στο SPIN με την ακόλουθο τρόπο: το SPIN αναγνωρίζει και αποθηκεύει τις παραμέτρους των καταστάσεων για κάθε διεργασία του μοντέλου και αντί να αποθηκεύει την πλήρη περιγραφή της κάθε κατάστασης στο καθολικό διάνυσμα κατάστασης, χρησιμοποιεί την ακολουθία των αναγνωριστικών (identifiers) για τις εμπλεκόμενες διεργασίες αυτές. Αποτελεί μια χωρίς απώλειες τεχνική συμπίεσης η οποία εγγυάται την εξαντλητική επικάλυψη όλων των παραγόμενων καταστάσεων.
Αναζήτηση Breadth-First	Η επιλογή αναζήτησης BFS στον on-the-fly παραγόμενο χώρο των καταστάσεων για λάθη.
Συγχώνευση καταστάσεων	Η συγχώνευση των καταστάσεων αποτελεί μια ειδική περίπτωση της αναδιατεταγμένης μερικής μείωσης. Συγχωνεύει redundant interleavings καταστάσεις των διεργασιών όπου αυτό είναι δυνατό, χωρίς όμως να αποπειράται προσπάθειας βελτίωσης όταν non-interleaved ακολουθίες καταστάσεων μπορούν να συγχωνευτούν σε μία.



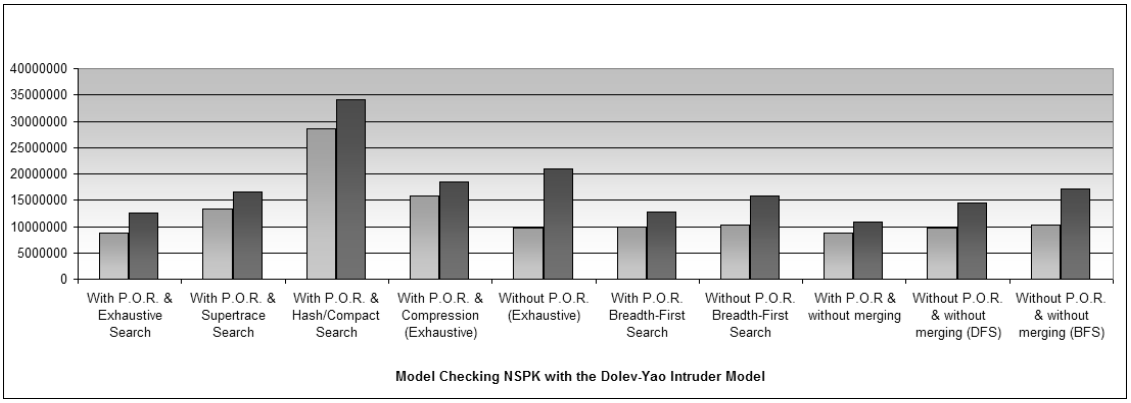
(ii)

Εικόνα 5.5.5 Διαγράμματα γεγονότων αιτίας-αποτελέσματος (cause-effect) για (i)το γενικό εισβολέα Dolev – Yao και (ii) τον εισβολέα διερεύνησης μηνύματος

Η εικόνα 5.5.5i δείχνει την τυπική ακολουθία των επιμέρους περιπτώσεων που πρέπει να μελετηθούν κατά την διάρκεια του ελέγχου μοντέλων ενός πρωτοκόλλου ασφαλείας χρησιμοποιώντας το μοντέλο εισβολέα DY και ΕΔΜ (Εικόνα 5.5.5 ii). Η διάκριση αυτών των βημάτων κρίνεται απαραίτητη για την περίπτωση όπου ο συνολικός παραγόμενος χώρος των καταστάσεων που αναμένεται να προκύψει στην διάρκεια της επαλήθευσης είναι άγνωστος. Επίσης εισάγεται τα προς εξερεύνηση γεγονότα αιτίας-αποτελέσματος που σχετίζονται με το πρόβλημα που μελετάται στον έλεγχο μοντέλων, άμεσα συνδεδεμένη με τον παραγόμενο χώρο των καταστάσεων. Τα αποτελέσματα που παρουσιάζονται παρακάτω επιβεβαιώνουν την εξάντληση όλης της διαθέσιμης μνήμης, ακόμα και για ένα μικρής πολυπλοκότητας πρωτόκολλο, όπως η περίπτωση του NSPK που εξετάζεται (3 βήματα πρωτοκόλλου για δύο παράλληλες συνόδους). Τέλος τα αποτελέσματα, επιδεικνύουν την σημαντική διαφορά του εισβολέα ΕΔΜ με το γενικό μοντέλο εισβολέα DY, όπου και παρατηρείται η μεγάλη μείωση στον χώρο των καταστάσεων, σε κάθε περίπτωση.



(i)



(ii)

Εικόνα 5.5.6 Μεγέθη των πλήρως παραγομένων χώρων καταστάσεων για τονNSPK (i) με το μοντέλο ΕΔΜ και (ii) με το μοντέλο εισβολέα DY

Ειδικότερα, η εικόνα 5.5.6 παρέχει αποτελέσματα από την διαδικασία της επαλήθευσης των ιδιοτήτων ασφαλείας για τον μεν μοντέλο του ΕΔΜ 5.5.6i και για το μοντέλο του DY 5.5.6ii. Οι παραγωγές των χώρων των καταστάσεων και για τις δύο περιπτώσεις που εξετάζονται, αφορούν στην καταμέτρηση και επαλήθευση όλων των πιθανών καταστάσεων που μπορεί να οδηγήσουν τα υλοποιημένα μοντέλα. Καταγράφεται έτσι, ο συνολικός αριθμός των μοναδικών καταστάσεων όπως τις εντόπισε ο αυτόματος ελεγκτής μοντέλων SPIN, στον δημιουργημένο πίνακα κατακερματισμού που χρησιμοποιήθηκε για την αποθήκευσή τους, με σκοπό τον γρήγορο έλεγχό τους. Πέρα αυτών, το SPIN παρήγαγε τον συνολικό άθροισμα των μοναδικών αλλά και των επαναεπισκεπτόμενων καταστάσεων, όπου ουσιαστικά αντικατοπτρίζουν τις μεταβάσεις που πραγματοποιήθηκαν στο μοντέλο. Η τεράστια συνεισφορά στον χώρο των καταστάσεων, όπως απεικονίζεται και στα διαγράμματα, με οποιαδήποτε συνδυασμό των προαναφερόμενων τεχνικών, μεταξύ του μοντέλου εισβολέα DY και του μοντέλου ΕΔΜ, υπολογίζεται σε διαφορά της τάξης του 10^3.

Η απαιτούμενη μνήμη για την ολοκλήρωση των παραπάνω πειραμάτων, για το βελτιωμένο μοντέλο του ΕΔΜ, για τις διάφορες εκδόσεις του, υπολογίστηκε μεταξύ 2.9 MB και 33MB, εκτός της περίπτωσης συμπίεσης όπου σε συνδυασμό με την τεχνικής της μερικής αναδιατεταγμένης μείωσης του χώρου των καταστάσεων, η απαιτούμενη μνήμη ξεπέρασε τα 260MB. Ένα τέτοιο φαινόμενο, ειδικά για την περίπτωση της συμπίεσης, είναι και το αναμενόμενο μιας και βάση της θεωρίας του [39], σε αρκετές περιπτώσεις στην

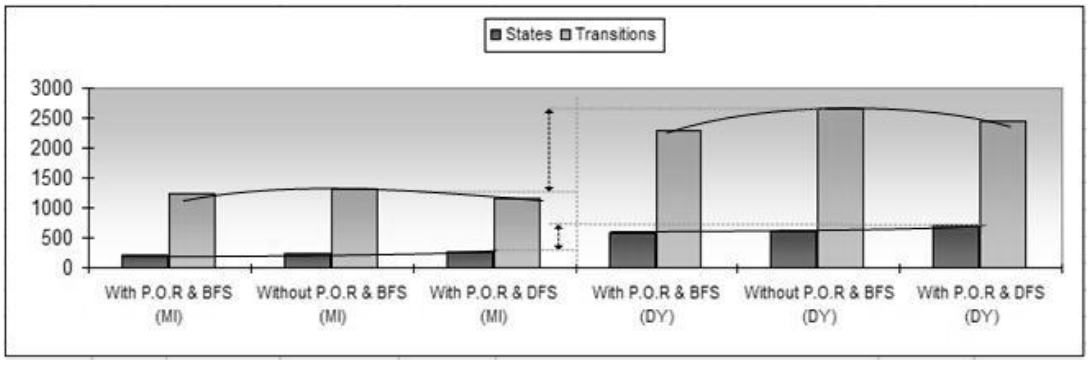
φάση του ελέγχου μοντέλων και των τεχνικών μείωσης του χώρου των καταστάσεων, οι τεχνικές συμπίεσης λειτουργούν με αρνητικές επιπτώσεις όσον αφορά την απαραίτητη μνήμη που χρειάζονται. Τα πλεονεκτήματα της τεχνικής συμπίεσης του χώρου των καταστάσεων απλά διαπιστώνονται με πειραματικές απόπειρες στον εκάστοτε χώρο που μελετάται. Από την άλλη μεριά, για την πλήρη παραγωγή του συνολικού χώρου των καταστάσεων για το μοντέλο εισβολέα DY, το μοντέλο κατανάλωσε όλη την διαθέσιμη μνήμη του τερματικού , διαθέσιμη αποκλειστικά για τον ελεγκτή μοντέλων στο όριο των 1.024 MB. Η τεχνική της συμπίεσης σε αυτή την φάση βρέθηκε να καταναλώνει μνήμη μεγέθους 363MB.

Οι συνολικοί αριθμοί που απεικονίζονται στην εικόνα 5.4.10, όπως αναφέρθηκε κάνουν αισθητό το γεγονός διαφοράς των χώρων των καταστάσεων των δύο εισβολέων, με τον εισβολέα ΕΔΜ να επιδεικνύει καλύτερη συμπεριφορά και περιορισμό του χώρου των καταστάσεων (10^3 φορές). Στην περίπτωση χρήσης των μοντέλων με την τεχνική της συμπίεσης το χάσμα των δύο χώρων μεγαλώνει, σε βαθμό όπου ο χώρος των καταστάσεων για το μοντέλο εισβολέα DY να είναι $4 \cdot 10^3$ φορές μεγαλύτερος από αυτόν του μοντέλου ΕΔΜ.

Στην εικόνα 5.5.6ii, οι επιλεγμένες τεχνικές παραγωγής του χώρου των καταστάσεων χωρίς την συμπίεση αυτού, παρουσιάζουν μικρότερο αριθμό αποθηκευμένων καταστάσεων (μοναδικών) όταν συγκρίνονταν με τις περιπτώσεις χρήσης συμπίεσης, τόσο για την περίπτωση του εισβολέα DY όσο και για τον ΕΔΜ. Κάτι τέτοιο επεξηγείται από την υλοποίηση και των δύο εισβολέων ως διεργασίες συμμετρικής φύσεως, αφού το σύνολο των κανόνων επίθεσης για τον μεν DY βασίζεται στους κανόνες παραγωγής των μηνυμάτων του, ενώ για τον ΕΔΜ στις τακτικές των ενεργειών επιθέσεων (ακολουθίες από εντολές send και receive). Κατά την ανάλυση των διεργασιών αυτών (και των εντολών τους), και με την προϋπόθεση χρήσης των πινάκων κατακερματισμού για την αποθήκευση των καταστάσεων, παρατηρείται ένας μεγάλος αριθμός από συγκρούσεις κατακερματισμού (hash collisions) κάτι που αυξάνει τις απαιτήσεις μνήμης για την διεξαγωγή της ανάλυσης, αλλά μειώνει την συνολική χρονική τους διάρκεια. Σαν παράδειγμα, αναφέρεται η περίπτωση του εισβολέα DY ο οποίος με την εφαρμογή του σε συνδυασμό με την τεχνική P.O.R., μια

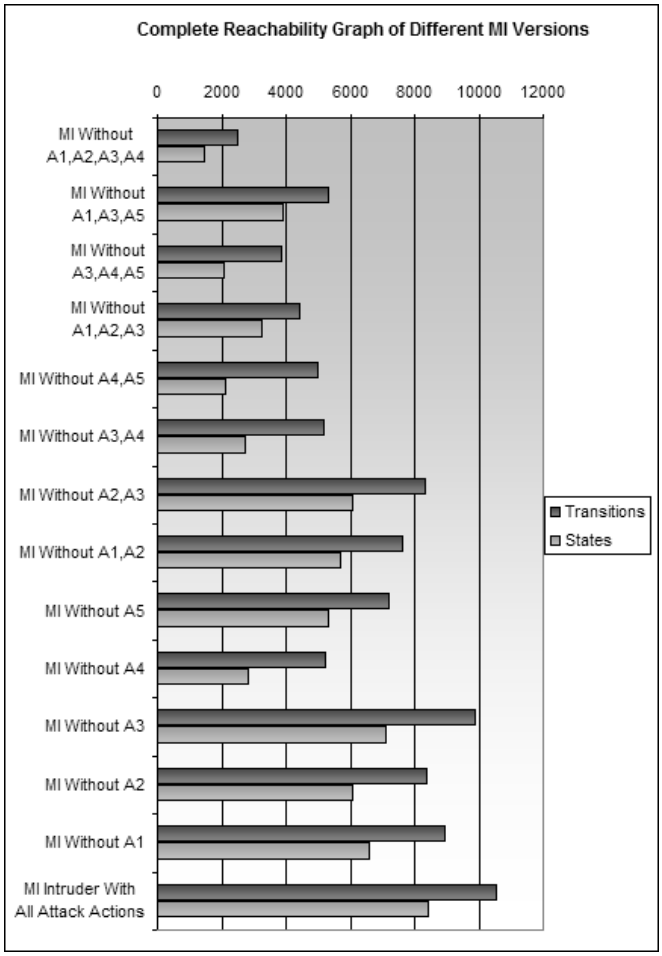
τυπική αναζήτηση για συγκρούσεις κατακερματισμού (δηλαδή επαναεπισκεπτόμενες καταστάσεις) επέστρεψε ένα αριθμό $4{,}3 \cdot 10^6$, σε έναν πίνακα συνολικού χώρου $8 \cdot 10^6$ μοναδικών καταστάσεων. Το αποτέλεσμα αυτό επιβεβαιώνει την ακραία γενικότητα δράσεων ενός εισβολέα DY στην περίπτωση του ελέγχου μοντέλων πρωτοκόλλων ασφαλείας, που έγκειται στο γεγονός έντονης επικάλυψης των καταστάσεων που δημιουργούν οι εντολές του, με βάση τους κανόνες παραγωγής καινούργιων (διεφθαρμένων ή όχι) μηνυμάτων. Σε πλήρη αντίθεση, το μοντέλο του εισβολέα του ΕΔΜ, έχει τέτοια υλοποίηση, βασιζόμενη στην ολοκληρωμένη βάση των επιθέσεων που περιέχει, όπου και κατά την ανάλυσή του, σε μια αναζήτηση για συγκρούσεις κατακερματισμού, ο μεγαλύτερος αριθμός που επέστρεψε, για όλα τα πειράματα που διεξήχθησαν με τις ποικίλες τεχνικές μείωσης, δεν ξεπέρασε τις 43.

Σε περιπτώσεις όπου το μοντέλο του πρωτοκόλλου απαιτεί περισσότερα βήματα προς ολοκλήρωση, η χρήση του εισβολέα DY είναι πιθανόν να απαιτήσει τεχνική συμπίεσης για να διεξαχθεί επιτυχώς, αναζητώντας λάθη στον χώρο με την τεχνικής αναζήτησης Supertrace. Το γεγονός αυτό όμως αφήνει ανοιχτό το ενδεχόμενο για την ελλιπή επικάλυψη όλων των καταστάσεων (coverage loss), πράγμα που αντιβαίνει με τους προαπαιτούμενους ελέγχους για τις όποιες προϋποθέσεις ασφαλείας έχει ορίσει ο αναλυτής. Στο ίδιο μήκος από την άλλη, το μοντέλο του ΕΔΜ αποδίδει έναν αρκετά εκμεταλλεύσιμο χώρο καταστάσεων, όπου σε συνδυασμό με τεχνικές προόδου (progress techniques) μπορούν να προβλέψουν την εύκολη και γρήγορη ανάλυσή του.



Εικόνα 5.5.7 Μέγεθος των χώρων καταστάσεων για το πρωτόκολλο NSPK για την ανίχνευση του λάθους, με τα μοντέλα εισβολέων ΕΔΜ και DY, για διάφορες τεχνικές αναζήτησης

Η εικόνα 5.5.7, παρουσιάζει αποτελέσματα για τον χώρο των καταστάσεων που παράγεται, με την προϋπόθεση ότι ο ελεγκτής μοντέλων προσπαθεί να αναγνωρίσει λάθη ή παραβιάσεις ασφαλείας που μπορεί να προκαλέσει ο εκάστοτε εισβολέας. Το μοντέλο του NSPK με τον εισβολέα του DY, στην περίπτωση εντοπισμού της επίθεσης πλαστοπροσωπίας που επιτυγχάνεται στο πρωτόκολλο, παρήγαγε 2,5 φορές μεγαλύτερο χώρο μοναδικών καταστάσεων σε σύγκριση με το μοντέλο του ΕΔΜ. Η βελτίωση αυτή βασίζεται στο βάθος που επιτυγχάνει ο ελεγκτής μοντέλων να εντοπίσει το λάθος με βάση τις πλεονάζουσες καταστάσεις που εισάγει ο κάθε εισβολέας, αλλά και των τεχνικών αναζήτησης του λάθους, που χρησιμοποιούνται μέσα από το εργαλείο SPIN. Στην περίπτωση της αναζήτησης BFS, το λάθος εντοπίζεται σε μικρότερο βάθος αλλά και σε γρηγορότερο χρόνο σε σύγκριση με την αναζήτηση DFS. Ενώ στην εικόνα 5.5.3, (μοντέλο ΕΔΜ με αναζήτηση λάθους DFS) το λάθος εντοπίστηκε σε βάθος 25, με $1{,}4 \cdot 10^3$ συγκρούσεις κατακερματισμού στον πίνακα κατακερματισμού, όπου και αποθηκεύθηκαν 264 μοναδικές καταστάσεις. Με την χρήση της BFS αναζήτησης, το λάθος εντοπίστηκε στο βάθος 5, χωρίς να έχουν ανιχνευθεί συγκρούσεις κατακερματισμού, με τον πίνακα κατακερματισμού να αποθηκεύει 112 μοναδικές καταστάσεις. Από την άλλη μεριά, κατά την διάρκεια επαλήθευσης του πρωτοκόλλου NSPK με το μοντέλο εισβολέα DY, με την αναζήτηση DFS το λάθος εντοπίζεται σε βάθος 48, προκαλώντας όμως $7{,}3 \cdot 10^4$ συγκρούσεις κατακερματισμού, με τον πίνακα κατακερματισμού να αποθηκεύει 698 μοναδικές καταστάσεις.



Εικόνα 5.5.8 Πλήρης διαγράμματα υπολογισμού του χώρου των καταστάσεων για διαφορετικές εκδόσεις του μοντέλου ΕΔΜ

Στο δεύτερο μέρος αυτής της μελέτης, η εικόνα 5.5.8 παρουσιάζει την επίδραση του μοντέλου ΕΔΜ για διαφορετικούς συνδυασμούς επιθέσεων, πάνω στο χώρο των καταστάσεων για το πρωτόκολλο NSPK. Αναφέρονται ο συνολικός αριθμός των μοναδικών καταστάσεων που κατέγραψε ο αυτόματος ελεγκτής μοντέλων SPIN, όπου οι συγκρούσεις κατακερματισμού που παρατηρήθηκαν είναι αμελητέες σε σχέση με τον παραγόμενο χώρο. Με βάση τα αποτελέσματα αυτά, ο αναλυτής μπορεί να εκτιμήσει και να προβλέψει έμμεσα την επιβάρυνση του χώρου των καταστάσεων από την επιλογή ή όχι μιας επίθεσης ορισμένη στην διεργασία του εισβολέα ΕΔΜ. Επιπρόσθετα, με βάση των χώρο των καταστάσεων, μερικών από των συνδυασμένων επιθέσεων, εκδόσεων του μοντέλου ΕΔΜ, ο αναλυτής μπορεί να επαναπροσδιορίσει τους κανόνες διερεύνησης (MI rules) που ορίστηκαν στο κεφάλαιο αυτό (πίνακας 5.4.2), με βάση την απτή πολυπλοκότητα του πρωτοκόλλου ασφαλείας που επιθυμεί να ελέγξει. Αξίζει, παρόλα αυτά να σημειωθεί ότι τα αναφερόμενα

αποτελέσματα, εξαρτώνται από την παράλληλε εκτέλεση των συμμετεχόντων οντοτήτων στο πρωτόκολλο, καθώς και από τις τεχνικές προδιαγραφές που εισάγει το κάθε επικοινωνιακό σύστημα (λ.χ. ορισμός τρίτης έμπιστης οντότητας).

Καθώς επεξηγήθηκε η επίδραση ενός μοντέλου εισβολέα στις απαιτήσεις μνήμης κατά την διάρκεια του ελέγχου μοντέλων, ένα ανοιχτό πρόβλημα στην όλη ανάλυση του χώρου των καταστάσεων, είναι η εξάρτηση του εισβολέα από τις εκάστοτε απαιτήσεις του πρωτοκόλλου που εξετάζεται. Η υλοποίηση του τόσο του εισβολέα ΕΔΜ όσο και του DY για το πρωτόκολλο NPSK, οι εισβολείς τέθηκαν ενήμεροι για τις προδιαγραφές του πρωτοκόλλου, όπως για παράδειγμα για τον τύπο των μηνυμάτων που θα τους μεταφέρονταν μέσα από το μοντελοποιούμενο κανάλι επικοινωνίας. Το πρόβλημα της γενίκευσης της υλοποίησης ενός εισβολέα έγκειται στην ικανότητα κωδικοποίησης με αφαίρεση, διεργασιών του εισβολέα από τον κάθε αναλυτή, αλλά και την ευκολία που παρέχεται από τον ελεγκτή μοντέλων που χρησιμοποιείται. Στο παρόν κεφάλαιο, μελετήθηκαν 14 διαφορετικές υλοποιήσεις (εκδόσεις) του εισβολέα ΕΔΜ, αλλά και του εισβολέα DY, προσπαθώντας να αντικατοπτριστεί οι επιδράσεις των μοντέλων αυτών, στον χώρο των καταστάσεων που πρέπει να ελεγχθεί σε συνδυασμό με ήδη υπάρχουσες τεχνικές μείωσης του χώρου, που προσφέρει το εργαλείο SPIN. Παρόλα αυτά, παρουσιάζεται ο τρόπος με τον οποίο ο έλεγχος μοντέλων, μπορεί βάση των παραπάνω, να προσπεράσει προβλήματα (τύπου ΕΧΚ) χωρίς να αποδυναμώνει τον εισβολέα που χρησιμοποιεί για τον έλεγχο πρωτοκόλλων ασφαλείας. Στην επόμενη ενότητα γίνεται η προσπάθεια καθοδήγησης του ειδικού-αναλυτή για την σωστή επαλήθευση των πρωτοκόλλων ασφαλείας με το μοντέλο ΕΔΜ.

5.5.4 Οδηγίες προς τον ειδικό ελέγχου μοντέλων πρωτοκόλλων ασφαλείας

Για τον αναλυτή που επιθυμεί την διεξαγωγή ελέγχου μοντέλων σε πρωτόκολλα ασφαλείας, μια αποτελεσματική διαδικασία εφαρμογής του προτεινόμενου μοντέλου ΕΔΜ σε μεγάλα-πολύπλοκα συστήματα, αποτελούν τα επόμενα βήματα:

- Μια προκαταρκτική εκτέλεση προσομοίωσης με το μοντέλο εισβολέα ΕΔΜ θα παρέχει σημαντικές πληροφορίες για τις οποίες αναγκαίες ή μη βελτιώσεις που μπορεί να εφαρμοστούν στο μοντέλο του ΕΔΜ
- Από την στιγμή που, ο προς παραγωγή χώρος των καταστάσεων, είναι άγνωστος, δεν υπάρχει η ανάγκη για εφαρμογή των βελτιώσεων από την πρώτη διαδικασία επαλήθευσης του συνολικού μοντέλου του πρωτοκόλλου με τον ΕΔΜ. Προτείνεται, η πρώτη απόπειρα παραγωγής του χώρου των καταστάσεων να λαμβάνει χώρα με την επιλογή αναζήτησης Supertrace σε συνδυασμό με την τεχνική P.O.R.

Για την επαλήθευση της απουσίας μη αποδεκτών καταστάσεων τερματισμού (παραβιάσεων ασφαλείας) σε ένα πρωτόκολλο που παράγει μεγάλο χώρο καταστάσεων, ο αναλυτής μπορεί να επιλέξει μεταξύ των ακόλουθων επιλογών:

- Για την εφαρμογή εναλλακτικής μεθόδου συμπίεσης του χώρου των καταστάσεων, (αναζήτηση hash-compact ή συμπίεση collapse). Παρόλο που αυτή η επιλογή επιφέρει επιπρόσθετο φόρτο, είτε με την μορφή επιπλέον χρονικής διάρκειας για τον αυτόματο έλεγχο μοντέλων ή με την μορφή επαναλαμβανόμενων δοκιμών για την καλύτερη παραμετροποίηση του παραγόμενου χώρου καταστάσεων.
- Για την εφαρμογή του μοντέλου ΕΔΜ και την αφαίρεση η πρόσθεση επιπλέον επιθέσεων στο σώμα του εισβολέα, ο αναλυτής υποχρεώνεται χειρωνακτικά να επέμβει στον κώδικα της PROMELA.

Η παρούσα έκδοση του μοντέλου ΕΔΜ μπορεί να παράγει χώρο καταστάσεων στον οποίο είναι επιτυχής όλες οι τεχνικές εξαντλητικής αναζήτησης, που διαθέτει το εργαλείο SPIN. Εάν το μέγεθος του χώρου καταστάσεων εκτιμηθεί σε τεράστια μεγέθη, τότε ο αναλυτής καλείται να πειραματιστεί με διάφορες τεχνικές συμπίεσης (hash-compact ή collapse) σε συνδυασμό και με την χρήση διαφορετικών εκδόσεων του μοντέλου ΕΔΜ, αλλά και με την χρήση άλλων τεχνικών μειώσεων. Νέες βελτιώσεις για το μοντέλο ΕΔΜ αλλά και για τους κανόνες διερεύνησης μηνύματος που βασίζεται η όλη προσέγγιση, μπορούν να επεκταθούν για πιο αποτελεσματική αλλά και ευρεία κάλυψη άλλων επιθέσεων ασφαλείας που μπορούν να υλοποιηθούν στον εισβολέα, πέρα από αυτές του πίνακα 5.4.1. Παρόλα αυτά η επιπλέον επέκταση του μοντέλου ΕΔΜ, πέραν του βασικού αλγορίθμου (MI initialization), χρειάζεται προγραμματιστικές

ικανότητες σε PROMELA από τον αναλυτή, αλλά και κατανόηση του θεωρητικού υποβάθρου που βασίζεται η δομή και λειτουργία του εισβολέα.

5.6 Σύγκριση του μοντέλου ΕΔΜ

Το προτεινόμενο μοντέλο εισβολέα ΕΔΜ του κεφαλαίου αυτού, μπορεί να συγκριθεί με εκείνες τις προσεγγίσεις που βελτιώνουν τον παραγόμενο χώρο των καταστάσεων, εξαιτίας των απεριόριστων μηνυμάτων που τείνει να δημιουργεί ένας εισβολέας και ειδικότερα εκείνοι που βασίζονται στους κανόνες παραγωγής μηνυμάτων του εισβολέα DY, χωρίς παράλληλα να στερείται η όλη ανάλυση από πιθανές επιθέσεις που μπορεί να εξαπολύσει ο εισβολέα. Τα υπάρχοντα μοντέλα εισβολέων [71][88][58][79][83] είτε βασίζονται σε απλή παραγωγή νέων διεφθαρμένων μηνυμάτων με σκοπό να προκαλέσουν προβλήματα στις έντιμες οντότητες του πρωτοκόλλου, ή συνδυάζουν τις ικανότητές τους με μια γλώσσα προδιαγραφής πλεονασμού ιδιοτήτων, προσπαθώντας να ελέγξουν τον χώρο των καταστάσεων με πιο αποδοτικό τρόπο. Η προσέγγιση του μοντέλου ΕΔΜ αποφεύγει, με βάση τις διαθέσιμες επιθέσεις που έχουν οριστεί, την ολοκληρωτική παραγωγή άπειρων μηνυμάτων από την γνώση του εισβολέα, επιτυγχάνοντας σημαντική μείωση στον παραγόμενο χώρο των καταστάσεων. Για παράδειγμα, ένα εξαιρετικά πολύπλοκης μορφής νέο διεφθαρμένο μήνυμα που μπορεί να παράγει ο εισβολέας, μπορεί να θεωρηθεί καινούργιο (και άρα και οι συνθήκες που μπορεί να δημιουργήσει άγνωστες), αλλά είναι απίθανο να προκαλέσει προβλήματα ασφαλείας σε σύγχρονα πρωτόκολλα που βασίζονται σε υπάρχοντες κρυπτογραφικούς μηχανισμούς.

Το μοντέλο ΕΔΜ, αποτελεί το πρώτο μοντέλο στην βιβλιογραφία που χρησιμοποιεί και επεξεργάζεται μεταδεδομένα μηνυμάτων, και όχι τα ίδια τα μηνύματα, κατά την πρώτη φάση της προσομοίωσης του πρωτοκόλλου μαζί με τον εισβολέα. Τα μεταδεδομένα αυτά βελτιώνουν την γνώση του εισβολέα με επιπρόσθετες πληροφορίες για τα ανταλλασσόμενα μηνύματα του υπό εξέταση πρωτοκόλλου ασφαλείας, οι οποίες μπορούν να χρησιμεύσουν στην εναλλακτική εκτέλεση ενεργειών του, μειώνοντας αποτελεσματικά τον χώρο των καταστάσεων κατά τον έλεγχο μοντέλων. Με βάση ένα σύνολο επιθέσεων,

που κατέχει ο εισβολέας στην δομή του, και ορίζοντας κανόνες διερεύνησης μηνύματος που απορρέουν από γνωστές αρχές παραβιάσεων της ασφάλειας, υποδεικνύεται στον αναλυτή ποιες επιθέσεις μπορούν να αφαιρεθούν χωρίς να θέσουν σε κίνδυνο την αποτελεσματικότητα της όλης ανάλυσης.

Ένα μειονέκτημα της όλης προσέγγισης που διεξάγεται σε δύο φάσεις ανάλυσης, είναι το αποτέλεσμα ενός μη γενικού μοντέλου εισβολέα ο οποίος περιορίζεται σε κανόνες που οριοθετούνται από την εκάστοτε μορφή και φύση των μηνυμάτων του πρωτοκόλλου ασφαλείας. Παρόλα αυτά, πιστεύεται ότι το μοντέλο ΕΔΜ μπορεί να καθοδηγήσει τον αναλυτή σε μια ανάλυση ασφαλείας που μπορεί να θεωρηθεί ως μια χαμηλού φόρτου εργασία ελέγχου μοντέλων, εύκολη να διεξαχθεί εξαιτίας της χειραγώγησης του παραγόμενου χώρου των καταστάσεων. Δεν πρέπει να παραβλεφθεί επίσης ο περιορισμός που εισάγεται στην επιλογή του εργαλείου που υλοποιείται το μοντέλο ΕΔΜ. Ο ελεγκτής μοντέλων θα πρέπει να παρέχει την δυνατότητα της προσομοίωσης του μοντέλου του πρωτοκόλλου με τον εισβολέα, αφού κάτι τέτοιο είναι απαραίτητο για την διεξαγωγή της πρώτης φάσης προσομοίωσης που απαιτείται για την ανάλυση, όπως ειπώθηκε και σε προηγούμενο κεφάλαιο (όπως το Murφ, το AVISPA). Στην παρούσα υλοποίηση [53] που διεξήχθη στο εργαλείο SPIN, σημειώνεται ότι ο αναλυτής θα πρέπει να έχει την απαραίτητη εξοικείωση με την γλώσσα προδιαγραφών PROMELA, για να μπορέσει να τροποποιήσει το συνολικό μοντέλο του εισβολέα από τυχόν προσθαφαιρέσεις των επιθέσεων ή αλλαγές στους κανόνες διερεύνησης των μηνυμάτων.

Τα κύρια πλεονεκτήματα του μοντέλου ΕΔΜ συνοψίζονται ως εξής:

- Παρέχει μια φορμαλιστική περιγραφή της όλης προσέγγισης η οποία είναι παραμετροποιημένη από τις εκάστοτε ανάγκες του αναλυτή
- Εμπεριέχει τέτοια δομή, όπου η επέκτασή του με περαιτέρω επιθέσεις ή κανόνες διερεύνησης μηνυμάτων, μπορεί να συμπεριλάβει επιπλέον επιθέσεις στον έλεγχο των πρωτοκόλλων ασφαλείας.
- Το μοντέλο μπορεί να επεκταθεί με επιπλέον υποσυναρτήσεις μεταδεδομένων εμπλουτίζοντας περισσότερο την ποιότητα γνώσης του εισβολέα
- Τέλος, πέρα από αποτελέσματα ελέγχου για μη αποδεκτές καταστάσεις τερματισμού, το μοντέλο του ΕΔΜ, μπορεί να συνδυαστεί με διάφορες

τεχνικές μείωσης του χώρου των καταστάσεων, όπως παρουσιάστηκε και παραπάνω, επιτυγχάνοντας επιπλέον συρρίκνωση του χώρου, με βάση το μοντέλο του συστήματος που ελέγχεται.

5.7 Συμπεράσματα κεφαλαίου

Το μοντέλο εισβολέα διερεύνησης μηνύματος ΕΔΜ, που παρουσιάστηκε στο κεφάλαιο αυτό, έχει ως σκοπό τον περιορισμό της συνδυαστικής πολυπλοκότητας που εισάγεται στον έλεγχο μοντέλων πρωτοκόλλων ασφαλείας, από την χρησιμοποίηση ενός μοντέλου εισβολέα βασισμένο στον DY, χωρίς όμως τον περιορισμό της συνολικής δύναμής του. Κάτι τέτοιο μπορεί και επιτυγχάνεται με την διερεύνηση των μηνυμάτων, δημιουργώντας κατάλληλα μεταδεδομένα που βελτιώνουν την γνώση και τις θέτουν περιορισμούς στις εκτελούμενες ενέργειες του εισβολέα, παραμετροποιώντας με αυτό τον τρόπο την συμπεριφορά του με σκοπό την αποφυγή φαινομένων ΕΧΚ. Ο μοναδικός περιορισμός που έγκειται στην χρήση του μοντέλου ΕΔΜ, είναι ότι το εργαλείο στο οποίο υλοποιείται θα πρέπει να περιλαμβάνει προσομοίωση του συνολικού μοντέλου προς επαλήθευση.

Διεξήχθησαν και παρουσιάστηκαν μια σειρά από πειράματα ελέγχου μοντέλων, προσπαθώντας να διερευνηθεί η όλη διαδικασία του αυτόματου ελέγχου μοντέλων, για το πρωτόκολλο ασφαλείας NSPK, σε απόλυτη σύγκριση με το γνωστό μοντέλο εισβολέα DY. Η συνολική υλοποίηση του εισβολέα ΕΔΜ παρέχεται σε κώδικα PROMELA για το NSPK στο [75], όπου και είναι δημόσια διαθέσιμος για εκπαιδευτικούς σκοπούς.

Το μοντέλο ΕΔΜ, παρέχει ένα ολοκληρωμένο πλαίσιο ελέγχου μοντέλων, υλοποιημένο στο περιβάλλον του ελεγκτή μοντέλων SPIN, όπου και παρέχει δυνατότητες επέκτασης της λειτουργικότητάς του με περαιτέρω επιθέσεις, υποσυναρτήσεις μεταδεδομένων ή κανόνες διερεύνησης μηνυμάτων. Περισσότερες μελλοντικές επεκτάσεις μπορούν να περιλάβουν ενοποίησης της γνώσης του συγκεκριμένου εισβολέα με πληροφορίες που έχουν να κάνουν με αναγνωριστικά πρότυπα των μηνυμάτων, άμεσα εξαρτώμενα από το επικοινωνιακό περιβάλλον που λειτουργεί το πρωτόκολλο ασφαλείας. Σε αυτή την περίπτωση επιπλέον κανόνες διερεύνησης μηνύματος μπορούν να

περιορίζουν ακόμα περισσότερο τους κανόνες παραγωγής διεφθαρμένων μηνυμάτων του κλασσικού εισβολέα, μειώνοντας επιπλέον τον χώρο των καταστάσεων. Άλλες προοπτικές εξέλιξης του μοντέλου ΕΔΜ, θα μπορούσε να είναι ο έλεγχός που θα διεξήγαγε για ιδιότητες ασφαλείας [88] που αφορούν γενικές ιδιότητες βιωσιμότητας του πρωτοκόλλου [23], όπως ορθός τερματισμός της συνόδου του πρωτοκόλλου, ή επικαιρότητα μηνύματος. Σε αυτή την περίπτωση το μοντέλο ΕΔΜ θα πρέπει να παρουσιάζει ενέργειες εκτελέσεων που αποσκοπούν στην εκπλήρωση υποθέσεων δικαιοσύνης που πρέπει να παρέχουν οι συμμετέχουσες οντότητες στο πρωτόκολλο, κάτι που δεν καλύπτεται σήμερα από τους υπάρχοντες εισβολείς στην βιβλιογραφία. Τέλος θα μπορούσε η όλη ανάλυση να υλοποιηθεί και να δομήσει έναν ξεχωριστό εργαλείο ελέγχου μοντέλων, όπου ανεξάρτητα από τους περιορισμούς που τίθενται από τα υπό εξέταση πρωτόκολλα ασφαλείας, θα μπορούσε να συμπεριλάβει τον εισβολέα ΕΔΜ αυτόματα στις προδιαγραφές που ορίζει ο αναλυτής για το μοντέλο του.

Κεφάλαιο 6ο

Το Μοντέλο Εισβολέα Πιθανοκρατικού Ελέγχου

6.1 Εισαγωγή

Οι μηχανισμοί αυθεντικοποίησης των πρωτοκόλλων που χρησιμοποιούνται σήμερα στις επικοινωνίες του ηλεκτρονικού εμπορίου τείνουν συχνά να είναι επιρρεπής στις γνωστές επιθέσεις άρνησης εξυπηρέτησης (DoS attacks). Οι τυπικές μέθοδοι ανάλυσης συστημάτων εστιάζουν στον εξαντλητικό έλεγχο των πρωτοκόλλων αυτών με σκοπό τον έλεγχο τυχόν παραβίασης ιδιοτήτων μυστικότητας ή αυθεντικοποίησης. Με βάση το γεγονός αυτό, επιθέσεις διαθεσιμότητας των παρεχόμενων υπηρεσιών, όπως οι επιθέσεις DoS, οι οποίες δεν συγκαταλέγονται στην κατηγορία ελέγχου ιδιοτήτων αυθεντικοποίησης ή μυστικότητας, είναι αρκετά δύσκολο και πολύπλοκο, μέχρι σήμερα να εντοπιστούν. Στο παρών κεφάλαιο παρουσιάζεται μια καινούργια μέθοδος ελέγχου επιθέσεων άρνησης εξυπηρέτησης, η οποία βασίζεται στον πιθανοκρατικό έλεγχο μοντέλων με το εργαλείο PRISM. Η συγκεκριμένη μέθοδος βασίζεται σε μια μεθοδολογία κατασκευής ενός πιθανοκρατικού εισβολέα (ΠΕ) ο οποίος μπορεί και εξαπολύει επιθέσεις τύπου DoS, έτσι ώστε κατά την εξαντλητικό έλεγχο του χώρου των καταστάσεων, να μπορεί να εντοπιστεί εάν μπορεί το υπό εξέταση πρωτόκολλο να βρεθεί σε τέτοια απειλή. Με τον ορισμό

του επακριβούς κόστους των όλων ενεργειών που εκτελούνται στο πρωτόκολλο, μπορούμε να εξάγουμε ακριβείς πιθανότητες με βάση πόσο θα είναι το κόστος για μια συγκεκριμένη οντότητα (και αν μπορεί να ανταπεξέλθει σε αυτό), σε περίπτωση μιας ενέργειας επίθεσης DoS.

Στην παρούσα φάση, αναπτύσσεται η θεωρία δημιουργίας του συγκεκριμένου εισβολέα, η οποία και αποτελείται από βασικά βήματα αποστολής, επεξεργασίας και λήψης μηνυμάτων. Μοντελοποιείται το υπό εξέταση πρωτόκολλο και ανάμεσα σε αυτό περιλαμβάνουμε έναν ενδιάμεσο ΠΕ. Το αντικατοπτριζόμενο κόστος της κάθε ενέργειας τόσο για τις έντιμες συμμετέχουσες οντότητες, όσο και για τον εισβολέα τον ίδιο, υπολογίζεται συνολικά με βάση το είδος της κρυπτογράφησης που χρησιμοποιεί το συγκεκριμένο πρωτόκολλο. Έτσι, δημιουργώντας το συνολικό μοντέλο συνδυασμένο με τις επακριβείς δομές κόστους, παράγεται το μοντέλο της Μαρκοβιανής αλυσίδας διακριτού χρόνου (DTMC) όπου μέσα στο περιβάλλον PRISM διεξάγεται η ανάλυση του πρωτοκόλλου για επιθέσεις άρνησης εξυπηρέτησης. Με καθορισμένα ερωτήματα PCTL λογικής, διερωτάται ο παραγόμενος χώρος καταστάσεων για το εάν υπάρχει περίπτωση να βρεθεί κατάσταση που ισοδυναμεί με κατάσταση επίθεσης DoS. Αν όντως υπάρξει τέτοια κατάσταση τότε το εργαλείο θα μας επιστρέψει και μια τιμή πιθανότητας με την οποία μπορεί αυτή η κατάσταση να περιέλθει. Επιπρόσθετα υπολογίζεται και ποιο θα είναι το εκτιμώμενο κόστος για έναν εισβολέα να διεξάγει μια επίθεση DoS για το συγκεκριμένο πρωτόκολλο. Κάτι τέτοιο μας βοηθά παράγουμε συγκριτικά αποτελέσματα για το πόσο δύσκολο για έναν εισβολέα θα είναι η διεξαγωγή μιας τέτοιας επίθεσης προς το υπό εξέταση πρωτόκολλο. Για την ελέγξουμε την αποτελεσματικότητα της προτεινόμενης τεχνικής, επιλέχθηκε το ειδικά σχεδιασμένο πρωτόκολλο προς αποφυγήν των επιθέσεων άρνησης εξυπηρέτησης, Host Identity Protocol (HIP). Αποτέλεσμα της όλης ανάλυσης ήταν η επιβεβαίωση για την ύπαρξη μιας DoS επίθεσης στο HIP (με την πιθανότητα να φτάνει σε τιμή 0.8), παράγοντας ακριβή αποτελέσματα για το κόστος υπολογίζεται, τόσο για τις έντιμες οντότητες όσο και για τον εισβολέα.

6.2 Πιθανοκρατικός έλεγχος μοντέλων

Οι τυπικές μέθοδοι ανάλυσης συστημάτων έχουν αποδείξει από την σχετική βιβλιογραφία την αποτελεσματικότητά τους στην μελέτη για την ύπαρξη ή μη προβλημάτων μυστικότητας ή αυθεντικοποίησης. Όπως προαναφέρθηκε, η χρήση εξειδικευμένων εισβολέων που αλληλεπιδρούν με τα υπό εξέταση πρωτόκολλα, μπορούν να βοηθήσουν στην ρεαλιστικότερη και πιο αποδοτική ανάλυση του μοντέλου για λάθη. Οι περισσότερες προσεγγίσεις δημιουργίας εισβολέων σήμερα στηρίζεται στις βασικές υποθέσεις που ορίστηκαν στο γνωστό μοντέλο εισβολέα των Dolev και Yao [34]. Οι υποθέσεις αυτές συνοψίζονται ως ακολούθως: α)Η μέθοδος της κρυπτογράφησης που χρησιμοποιείται θεωρείται απαραβίαστη, β) ο εισβολέας έχει την ικανότητα να αποτρέψει οποιοδήποτε μήνυμα του πρωτοκόλλου να φτάσει στον τελικό προορισμό του και γ) ο εισβολέας έχει την δυνατότητα να δημιουργεί δικά του μηνύματα. Παρόλα αυτά όμως, ένας εισβολέας ο οποίος βασίζεται στις προαναφερθείσες υποθέσεις έχει την δυνατότητα, με ορισμένες εξειδικευμένες ενέργειες να αποτρέψει την διαθεσιμότητα μιας έντιμης οντότητας ενός πρωτοκόλλου, με σκοπό την εύρεση ή όχι σφάλματος άρνησης εξυπηρέτησης. Για να μπορέσουμε παράλληλα να επαληθεύσουμε την ύπαρξη μιας τέτοιας κατάστασης για το πρωτόκολλο, θα πρέπει να εισάγουμε στο μοντέλο μας ανάλογες δομές ενεργειών σε σχέση με το κόστος αυτών, με σκοπό την ποσοτική εξαντλητική ανάλυση του χώρου των παραγόμενων καταστάσεων.

Το υλοποιημένο μοντέλο του πιθανοκρατικού εισβολέα ΠΕ, βασίζεται εν μέρει στον εισβολέα ΕΠΕ ως προς την δομή του, που παρουσιάστηκε στο κεφάλαιο 4 [7]. Η ανάλυση και εντοπισμός μιας επίθεσης άρνησης εξυπηρέτησης εκφράζεται ως ερώτημα πιθανοκρατικής λογικής, χαρακτηριζόμενη ως πιθανοκρατική ιδιότητα προσέγγισης μιας οποιασδήποτε κατάστασης προς επαλήθευση, με βάση τους [57] και [41], σε απόλυτα συμφωνία με το κατάλληλο μοντέλο DTMC που αναπαριστά τις οντότητες του υπό εξέταση πρωτοκόλλου. Η όλη εργασία διεξάγεται στο περιβάλλον που παρέχει το εργαλείο PRISM [91]. Με την βοήθεια αυτού, βελτιώνεται η αποτελεσματικότητα της ανάλυσης από τους σχεδιαστές πρωτοκόλλων μιας και η διεξαγωγή οποιαδήποτε ελέγχου γίνεται με αυτοματοποιημένο τρόπο.

Επιλέγοντας το πρωτόκολλο ασφαλείας HIP για την διεξαγωγή της προτεινόμενης ανάλυσης, επειδή εστιάζουμε στο κόστος των εκτελούμενων ενεργειών, θα πρέπει να έχουμε στην διάθεσή μας πραγματικές πληροφορίες μετρικές, οι οποίες και θα αποτυπώνουν τον χρόνο που είναι απαραίτητος για την όποια ενέργεια (λ.χ. κρυπτογράφηση ενός μηνύματος, αποστολή ή λήψη, αναμονή κ.α.). Μια τέτοια ανάλυση απόδοσης του πρωτοκόλλου βασισμένη στο χρόνο, παρουσιάζεται στο [24]. Με βάση τα αποτελέσματα αυτής επιλέχθηκαν οι ανάλογες τιμές κόστους για κάθε βήμα το πρωτοκόλλου, δομώντας με αυτόν τον τρόπο το απαραίτητο μοντέλο για την τελική επαλήθευση ποσοτικοποίησης που σκοπεύουμε να διεξάγουμε.

Βασιζόμενοι στο μοντέλο εισβολέα [7], υλοποιήθηκε ένας ΠΕ εισβολέας, οποίος θα είχε την δυνατότητα να προσομοιώσει ταυτόχρονα *N* 'οντότητες φαντάσματα (zombie entities) με σκοπό την παραβίαση του ενσωματωμένου DoS μηχανισμού που περιέχεται στο πρωτόκολλο HIP. Η ανάλυση επίσης βασίζεται στον ορισμό προκαθορισμένων ερωτημάτων με χρήση της πιθανοκρατικής χρονικής λογικής (Probabilistic Computational Tree Logic, PCTL), παράγοντας αποτελέσματα για όλες τις συμμετέχουσες οντότητες (συμπεριλαμβανομένου και του εισβολέα) βασισμένα στο κόστος διεξαγωγής ενεργειών.

Το κεφάλαιο αυτό περιλαμβάνει τις παρακάτω υπό-ενότητες. Στην πρώτη πραγματοποιείται μια επισκόπηση των σχετικών εργασιών που έχουν βρεθεί στην βιβλιογραφία με σκοπό την ανάδειξη των διαφορών σε σχέση με την προτεινόμενη προσέγγιση. Στη συνέχεια, παρατίθεται μια εισαγωγή στην θεωρία του πιθανοκρατικού ελέγχου μοντέλων και ορίζονται οι περιπτώσεις όπου ένα πρωτόκολλο μπορεί να περιέλθει σε κατάσταση άρνησης της εξυπηρέτησης. Κάτι τέτοιο θα βοηθήσει στον καλύτερο ορισμό των απαραίτητων ερωτημάτων που θα μας οδηγήσουν στην αποκάλυψη τέτοιων λαθών. Στη συνέχεια παρουσιάζεται ο εισβολέας που δημιουργήθηκε και το μοντέλο του HIP πάνω στο οποίο εφαρμόστηκε η όλη ανάλυση. Αποτελέσματα επαλήθευσης όλων των παραπάνω παρουσιάζονται στο τέλος αυτού του κεφαλαίου, μαζί με τα τελικά συμπεράσματα και οφέλη χρήσης της προτεινόμενης μεθόδου.

6.3 Σχετική ερευνητική βιβλιογραφία

Η σημαντικότητα της διεξαγωγής μιας ανάλυσης διαθεσιμότητας υπηρεσίας στα κρυπτογραφικά πρωτόκολλα ασφαλείας παρουσιάστηκε αρχικά στην [66]. Οι συγγραφείς εξετάζουν την επίθεση άρνησης εξυπηρέτησης στο πλαίσιο μιας παραβίασης των διαθέσιμων πόρων μιας οντότητας υλοποιώντας μια θεωρία για την μέτρηση του κόστους της έντιμης οντότητας έναντι του επιτιθέμενου. Η θεωρία αυτή παρόλο που δεν έχει υλοποιηθεί σε κάποιο εργαλείο ελεγκτή μοντέλου της ασφάλειας, παρατίθεται προς ενοποίηση με κάποιο από αυτά ως μελλοντική προοπτική. Πρόσφατη η θεωρία που προτείνεται στο [66] υλοποιήθηκε στο [92], όπου και το υπολογιστικό κόστος της όποιας ενέργειας ενός επικοινωνιακού συστήματος καθορίζεται με απόλυτα ακρίβεια. Παρόλα αυτά όμως η ανάλυση κόστος που πραγματοποιείται στο υλοποιημένο χρονικό-χρωματισμένο δίκτυο Petri πραγματοποιείται με τεχνική προσομοίωσης αδυνατώντας να επαληθεύσουν τις όποιες ποσοτικές ιδιότητες επιθυμεί ο αναλυτής. Μια ακόμη ενδιαφέρουσα προσέγγιση παρουσιάζεται στην εργασία [64] όπου στοχαστική αναλυτική προσέγγιση δημιουργίας ενός μοντέλου με σκοπό την ποσοτικοποίηση της διαθεσιμότητας επικοινωνιακών συστημάτων τα οποία και υπόκεινται σε συνεχείς επιθέσεις άρνησης της εξυπηρέτησης. Η συγκεκριμένη προσέγγιση μοντελοποιεί το παραπάνω σύστημα με την βοήθεια ημί-Μαρκοβιανών διεργασιών (semi-Marcov processes, SMP).

Ξεκινώντας από το συγκεκριμένο μοντέλο είναι εύκολο να εξαχθούν οι συγκεκριμένες ενθυλακωμένες Μαρκοβιανές αλυσίδες διακριτού χρόνου (DTMC), οι οποίες περιλαμβάνουν μόνο ένα μέρος των συνολικών πιθανοτήτων των μεταβάσεων του μοντέλου. Μετά και τον υπολογισμό των steady-state πιθανοτήτων των παραγόμενων DTMC, λαμβάνονται κάποιες ισχυρές υποθέσεις για την κατανομή των πιθανοτήτων των συνολικών καταστάσεων του μοντέλου με σκοπό τον υπολογισμό πιθανοτήτων για τα SMPs. Με βάση την τεχνική αυτή, είναι δυνατόν ο υπολογισμός της διαθεσιμότητας του συστήματος και κατά συνέπεια, μπορεί να πραγματοποιηθεί μια παραμετροποιημένη ανάλυση ευαισθησίας με στόχο την εξέταση της υπολογιζόμενης διαθεσιμότητας. Η συγκεκριμένη προσέγγιση παρόλα αυτά, απαιτεί ιδιαίτερη εξοικείωση με μοντελοποίηση στοχαστικών συστημάτων καθώς και ανάλυσή τους από τον

χρήστη μιας η συνολική προσέγγιση δεν δύναται να ολοκληρωθεί με αυτοματοποιημένο τρόπο, σε αντίθεση με το περιβάλλον του PRISM. Επιπρόσθετα στην παραπάνω ανάλυση δεν λαμβάνεται υπ' όψιν οι δαπάνες των διαθέσιμων πόρων όλων των καταστάσεων των συμμετεχόντων οντοτήτων, με αποτέλεσμα να μην περιλαμβάνονται κόστη που έχουν να κάνουν με την επεξεργασία των μηνυμάτων του πρωτοκόλλου, τα οποία και παίζουν σημαντικό ρόλο στην οριοθέτηση και εντοπισμό πιθανόν επιθέσεων της άρνησης της εξυπηρέτησης.

Η πιο σχετική εργασία που παρουσιάζεται στη σημερινή βιβλιογραφία είναι αυτή της [1]. Σε αυτήν οι συγγραφείς ορίζουν ένα μοντέλο με την βοήθεια μιας πιθανοκρατικής rewriting λογικής ένα ανεκτικό μηχανισμό άρνησης εξυπηρέτησης στο πρωτόκολλο TCP. Όπως και στην δικιά μας προσέγγιση, στη συγκεκριμένη τεχνική περιλαμβάνονται ένας αριθμός έντιμων συμμετεχόντων στο υπό εξέταση πρωτόκολλο και ένα μοντέλο εισβολέα, ο οποίος και δημιουργεί κάλπικες αιτήσεις με μια συγκεκριμένη κλίμακα (η οποία βασίζεται σε μια εκθετική κατανομή). Με τον τρόπο αυτό, δημιουργείται μια τεχνική «ποταμοειδούς» επίθεσης (flooding attack) από την μεριά του εισβολέα, η οποία έχει ως στόχο την απάλειψη διαθεσιμότητας μιας έντιμης οντότητας του πρωτοκόλλου. Το συγκεκριμένο μοντέλο υλοποιήθηκε με την εργαλειοθήκη VESTA [85] η οποία μετά την παραγωγή ενός χρονικού πιθανοκρατικού μοντέλου, αναλύεται με την βοήθεια της τεχνικής προσομοίωσης Monte Carlo (με χρήση μιας ακολουθίας δοκιμών ενδοσυσχετιζόμενων στατιστικών υποθέσεων), ελέγχοντας έτσι αν η ποσοτική ιδιότητα ικανοποιείται. Η συνολική αυτή τεχνική αναφέρεται στην βιβλιογραφία ως στατιστικός έλεγχος μοντέλων. Σε σύγκριση με την προτεινόμενη προσέγγιση που παρουσιάζεται στο παρόν κεφάλαιο με βάση τον πιθανοκρατικό έλεγχο μοντέλων, η προαναφερθείσα τεχνική δεν παράγει τα ίδια ακριβή αποτελέσματα [86]. Επιπρόσθετα και σε αυτή την ανάλυση δεν λαμβάνονται υπ' όψιν το συνολικό κόστος διαχείρισης των ανταλλασσόμενων μηνυμάτων, καθιστώντας μη δυνατή την ποσοτική ανάλυση των συμμετεχουσών οντοτήτων στο πρωτόκολλο.

6.4 Πιθανοκρατικός έλεγχος μοντέλων για την ανάλυση πρωτοκόλλων ασφαλείας

Τα σημερινά πρωτόκολλα ασφαλείας αρκετές φορές επιδεικνύουν μη προβλέψιμες συμπεριφορές κατά την διάρκεια της εφαρμογής τους σε υπολογιστικά συστήματα επικοινωνιών. Αρκετές φορές δημιουργούνται απρόβλεπτες καταστάσεις μεταξύ των συμμετεχόντων του πρωτοκόλλου, εξαιτίας μιας ανέντιμης οντότητας ή ενός εισβολέα. Για τον ακριβέστερο έλεγχο των καταστάσεων αυτών που οδηγούν σε παραβίαση των εγγυήσεων ασφαλείας που παρέχουν τα πρωτόκολλα, θα χρησιμοποιήσουμε την τεχνική του πιθανοκρατικού ελέγχου μοντέλων με σκοπό την ποσοτική ανάλυση των όποιων απειλών ασφαλείας. Η συνολική αυτή ανάλυση βασίζεται σε επισηματοθετημένες μεταβάσεις (labeled transitions) μεταξύ των καταστάσεων του μοντέλου με την επιπρόσθετη πληροφορία της πιθανότητας αυτές οι μεταβάσεις να πραγματοποιηθούν [56].

6.4.1 Θεωρία πιθανοκρατικού ελέγχου μοντέλων

Όπως προαναφέρθηκε και στην εισαγωγή του κεφαλαίου αυτού, για την περάτωση της όλης ανάλυσης θα χρησιμοποιηθεί ο πιθανολογικός ελεγκτής μοντέλων PRISM [91]. Στο εργαλείο αυτό, τα πιθανοκρατικά μοντέλα, υλοποιούνται στην PRISM μετα-γλώσσα (PRISM meta-language).

Ένα τέτοιο μοντέλο, ορίζεται σαν ένα σύνολο από m δομοστοιχεία (modules) $M_1,...,M_m$. Κάθε δομοστοιχείο M_i αποτελείται ένα ζευγάρι (Var_i, C_i) με Var_i να είναι ένα σύνολο από ακέραιες τοπικές μεταβλητές με πεπερασμένο εύρος, και C_i ένα σύνολο από εντολές. Με την μεταβλητή Var υποδηλώνουμε το σύνολο όλων των τοπικών μεταβλητών μέσα στο μοντέλο ως $Var = \bigcup_{i=1}^{m} Var_i$. Κάθε μεταβλητή $v \in Var$ έχει μια αρχική τιμή $\bar{v}$. Κάθε εντολή $c \in C_i$ θα έχει την μορφή (g, (λ_1, u_1), . . ., (λ_{nc}, u_{nc})), σχηματίζοντας μια συνθήκη φρουρό (guard) g και ένα σύνολο από ζεύγη (λ_j, u_j) όπου $\lambda_j \in$ IR > 0 με u_j να αποτελεί μια ανανέωση (update) για κάθε $1 \leq j \leq n_c$. Ένας φρουρός g αποτελεί ένα κατηγόρημα πάνω σε όλο το καθορισμένο σύνολο των τοπικών μεταβλητών Var, με την κάθε ανανέωση u_j να ανταποκρίνεται σε μία πιθανή μετάβαση του δομοστοιχείου M_i.

Εάν το σύνολο Var_i περιέχει μια n_i τοπική μεταβλητή του συνόλου $v_1, \ldots, v_{n_i}$, τότε μια ανανέωση θα έχει την μορφή $(v_1' = \text{expr}_1) \wedge \ldots \wedge (v_{n_i}' = \text{expr}_{n_i})$ όπου κάθε $expr_j$ αποτελεί μια έκφραση για τις μεταβλητές του συνόλου Var. Σε μια ανανέωση μεταβλητής, οι τιμές κάποιων μεταβλητών στο Var_i παραμένουν αναλλοίωτες, τότε το συνολικό μοντέλο μπορεί να παραλείψει την (μη-αλλαγμένη) πληροφορία αυτή. Σε ένα μοντέλο Μαρκοβιανής αλυσίδας διακριτού χρόνου DTMC οι τιμές λ_j καθορίζουν την πιθανότητα της συγκεκριμένης μετάβασης, έχοντας $\lambda_j \in (0, 1]$ με $1 \leq j \leq n_c$ και $\sum_{j=1}^{n_c} \lambda_j = 1$.

Ορισμός 1. *Ένα μοντέλο Μαρκοβιανής αλυσίδας διακριτού χρόνου (Discrete Time Marcov Chain, DTMC), αποτελεί μια πλειάδα (S, $\bar{s}$, P, L) όπου:*
Το S αποτελεί ένα πεπερασμένο σύνολο καταστάσεων
$\bar{s} \in S$ *είναι η αρχική κατάσταση*
P: $S \times S \rightarrow [0, 1]$ *δηλώνει τον πίνακα πιθανοτήτων μετάβασης έτσι ώστε* $\sum_{s' \in S} P(s, s') = 1$ *για κάθε* $s \in S$
L: $S \rightarrow 2^{AP}$ *αποτελεί μια συνάρτηση σηματοδοσίας η οποία αντιστοιχίζει καταστάσεις σε σύνολα ατομικών προτάσεων από ένα σύνολο AP το οποίο περιέχει τις επιθυμητές ιδιότητες*

Οι καταστάσεις τερματισμού μοντελοποιούνται με μία απλή μετάβαση η οποία ξαναγυρνά στην ίδια κατάσταση με πιθανότητα 1. Ένα μονοπάτι (path) ω αποτελεί μια μη κενή ακολουθία καταστάσεων $s_0 s_1 s_2 \ldots$, όπου $s_i \in S$ και $P(s_i, s_{i+1}) > 0$ για κάθε $i \geq 0$. Ένα τέτοιο μονοπάτι μπορεί να είναι πεπερασμένο ή μη πεπερασμένο. Το σύνολο όλων των μη πεπερασμένων μονοπατιών αρχίζοντας από μια κατάσταση s ορίζεται ως $Path_s$. Επίσης ορίζουμε ότι ένα πεπερασμένο μονοπάτι ω_{fn} μήκους n αποτελεί επίθεμα του μη πεπερασμένου μονοπατιού ω, εάν οι πρώτες $n+1$ καταστάσεις του ω αντιστοιχούν ακριβώς στο μονοπάτι ω_{fn}. Τέλος ορίζεται ως το κυλινδρικό σύνολο $C(\omega_{fn})$ (cylinder set) για ένα πεπερασμένο μονοπάτι ω_{fn} ορίζεται ως το σύνολο όλων των μονοπατιών με επίθεμα prefix ω_{fn}.

Για να μπορέσουμε να υπολογίσουμε την πιθανότητα να ακολουθηθεί ένα μονοπάτι ενός DTMC μοντέλου, ορίζουμε την μετρική πιθανότητας $Prob_s$ στο

μονοπάτι $Path_s$, για κάθε κατάσταση $s \in S$ ως ακολούθως. Πρώτα ορίζουμε την πιθανότητα $P(\omega_{fn})$ ενός πεπερασμένου μονοπατιού ω_{fn} ως:

$P(\omega_{fn})=1$ *εάν* ω_{fn} *αποτελείται από μια μοναδική κατάσταση*

$P(\omega_{fn})=P(s_0, s_1) \cdot P(s_1, s_2) \cdots P(s_{n-1}, s_n)$ *στην γενική περίπτωση, όπου* $\omega_{fn}= s_0 s_1 s_2 \ldots s_n$

Ορισμός 2. *Έστω* Σ_s *να αποτελεί την μικρότερη σ-άλγεβρα πάνω στο μονοπάτι* $Path_s$ *το οποίο περιέχει όλα τα σύνολα* $C(\omega_{fn})$ *όπου* ω_{fn} *κλιμακώνεται πάνω σε όλα τα πεπερασμένα μονοπάτια τα οποία ξεκινούν από το s. Ορίζουμε την μετρική πιθανότητας* $Prob_s$ *πάνω στο* Σ_s *ως την μοναδική μετρική όπου* $Prob_s(C(\omega_{fn}))=P(\omega_{fn})$.

Με βάση τα παραπάνω, είναι δυνατό να ποσοτικοποιηθεί η πιθανότητα με την οποία συμπεριφέρεται ένα DTMC με το να αναγνωρίζονται το σύνολο των μονοπατιών τα οποία ικανοποιούν την υπό εξέταση ιδιότητα που επιθυμούμε να ελέγξουμε, με την προϋπόθεση φυσικά ότι αυτή η πιθανότητα είναι μετρήσιμη χρησιμοποιώντας την μετρική $Prob_s$. Παρακάτω ακολουθεί ο ορισμός της λογικής που θα χρησιμοποιηθεί για την σύνταξη των ερωτημάτων-ιδιοτήτων που διενεργούνται πάνω στα υλοποιημένα μοντέλα.

Ορισμός 3. *Η σύνταξη της πιθανοκρατικής χρονικής λογικής (Probabilistic Computational Tree Logic, PCTL), ορίζεται ως εξής:*

$$\Phi ::= true \mid \alpha \mid \neg\Phi \mid \Phi \wedge \Phi \mid P_{\sim p}[\varphi], \text{ για ιδιότητες καταστάσεων}$$

$$\varphi ::= X\Phi \mid \Phi\, U^{\leq k}\, \Phi, \text{ για ιδιότητες μονοπατιών,}$$

οι οποίες υπολογίζονται πάνω από τις καταστάσεις και τα μονοπάτια προερχόμενα από ένα DTMC, όπου το α *αποτελεί μια ατομική πρόταση τύπου ,* $\sim \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$ *και* $k \in \mathbb{N} \cup \{\infty\}$.

Για τον ορισμό μιας ιδιότητας χρησιμοποιείται πάντα μια φόρμουλα κατάστασης ή μια φόρμουλα μονοπατιού, η οποία πρέπει να εμφανίζεται ως παράμετρος του τελεστή $P_{\sim p}[\cdot]$. Σε ένα DTMC, μια κατάσταση s ικανοποιεί την ιδιότητα $P_{\sim p}[\varphi]$ μόνο εάν η πιθανότητα παίρνοντας ένα μονοπάτι από το s ικανοποιεί την λογική έκφραση φ, και εμφανίζεται στο εύρος του $\sim p$. Μια τέτοια

περίπτωση μπορεί να ποσοτικοποιηθεί με την μετρική πιθανότητας $Prob_s$ που έχει οριστεί προηγουμένως πάνω στο $Path_s$.

Σε μια φόρμουλα μονοπατιού επιτρέπεται ο τελεστής X (επόμενος) και ο τελεστής $U^{\leq k}$ (οριοθετημένο μέχρι σε k χρονικά βήματα), οι οποίοι θεωρούνται προαπαιτούμενοι στην χρονική λογική. Το 'μη οριοθετημένο μέχρι' ορίζεται παίρνοντας το k να τείνει στο ∞, λ.χ Φ U Ψ = Φ $U^{\leq \infty}$ Ψ. Εκτός από του τελεστές που δίνονται στον ορισμό 2, η PCTL περιλαμβάνει και έναν αριθμό επιπρόσθετων τελεστών, που δίνονται παρακάτω:

$false \equiv \neg true$

$\Phi \vee \Psi \equiv \neg (\neg\Phi \wedge \neg\Psi)$

$\Phi \rightarrow \Psi \equiv \neg\Phi \vee \Psi$

Οι φόρμουλες μονοπατιών μπορούν επίσης να περιέχουν τελεστές όπως ο ◊ (eventually) και □ (always) στις οριοθετημένες ή όχι παραλλαγές τους:

$P_{\sim p}[\Diamond^{\leq k} \Phi] \equiv P_{\sim p}[true\ U^{\leq k} \Phi]$

$P_{\sim p}[\Diamond \Phi] \equiv P_{\sim p}[true\ U^{\leq \infty} \Phi]$

$P_{\sim p}[\Box^{\leq k} \Phi] \equiv P_{\approx\ 1-p}[\Diamond^{\leq k} \neg\Phi]$

$P_{\sim p}[\Box \Phi] \equiv P_{\approx\ 1-p}[\Diamond\neg\Phi]$

όπου $\overline{<} \equiv >$, $\overline{\leq} \equiv \geq$, $\overline{\geq} \equiv \leq$ *και* $\overline{>} \equiv <$.

Εκτός από ποσοτικές προτάσεις επιβεβαιώσεων, στην PCTL μπορούμε να εκφράσουμε ιδιότητες οι οποίες αντιστοιχούν σε αριθμητικές τιμές. Οι ιδιότητες αυτές ορίζονται με τον τύπο: $P_{=?}[\varphi]$. Στο συνολικό μοντέλο του DTMC που ορίζεται με την γλώσσα είσοδο του εργαλείου PRISM, η όλη υλοποίηση ακολουθεί μια παράλληλη σύνθεση τον δομοστοιχείων του με τον υπολογισμό τον προσεγγίσιμου χώρου των καταστάσεων που βρίσκονται σε ένα από τα παραγόμενα μονοπάτια. Όποιες καταστάσεις δεν ανήκουν σε ένα τέτοιο μονοπάτι απλά διαγράφονται [56][56].

Σε κάθε κατάσταση, υπάρχει ένα σύνολο από εντολές –οι οποίες ανήκουν σε κάποιο από τα δομοστοιχεία– και έχουν την δυνατότητα οποιαδήποτε στιγμή να ενεργοποιηθούν. Η επιλογή το ποια εντολή θα είναι αυτή που θα εκτελεστεί βασίζεται σε πιθανότητες, με κάθε εντολή να έχει ίση πιθανότητα ενεργοποίησης. Οι ορισμένες ιδιότητες PCTL επαληθεύονται με την εφαρμογή

ειδικών αλγορίθμων (ελέγχου μοντέλων) με επαγωγή πάνω στην σύνταξή τους. Οι υπολογιστικές δυνατότητες του PRISM αποτελούν έναν συνδυασμό:

- Θεωρητικών διαγραμματικών αλγορίθμων, για ανάλυση προσεγγισιμότητας και ποιοτική ανάλυση πιθανοκρατικού ελέγχου μοντέλων
- Αριθμητικούς υπολογιστικούς μηχανισμούς, για ποσοτικό πιθανοκρατικό έλεγχο μοντέλων που στην περίπτωση του DTMC αντιστοιχεί στην επίλυση πολλαπλών γραμμικών εξισώσεων.

Ορισμός 4. *Μια δομή κόστους (reward structure) ορίζεται ως ένας ζεύγος (ρ, ι) για ένα DTMC, το οποίο επιτρέπει τον ορισμό δύο διαφορετικών τύπου κόστους: Κόστη κατάστασης (ή αλλιώς συσσωρευτικό κόστος) που ορίζονται μέσω της συνάρτησης κόστους* $\rho: S \to IR \geq 0$ *, έτσι ώστε ένα κόστος ρ(s) επισύρεται εάν το μοντέλο του DTMC βρίσκεται στην κατάσταση s σε ένα χρονικό βήμα;*
Κόστη μετάβασης (ή αλλιώς στιγμιαίο ή ενστικτώδες κόστος), ορίζεται μέσω της συνάρτησης κόστους $\iota: S \times S \to IR \geq 0$ *έτσι ώστε ένα κόστος ι(s, s΄) απαιτεί κάθε φορά μια μετάβαση μιας κατάστασης s σε κατάσταση s'.*

Στο πρόβλημα το οποίο εξετάζουμε, τα κόστη που προσαρμόζουμε στις καταστάσεις ή τις μεταβάσεις του συγκεκριμένου μοντέλου, αναπαριστούν κατανάλωση ενός πεπερασμένου μετρήσιμου μεγέθους το οποίο στην μοντελοποιούμενη απειλή μιας επίθεσης άρνησης εξυπηρέτησης μπορεί να σημαίνει εύρος επικοινωνίας, μνήμη ή υπολογιστική ισχύ. Στην γλώσσα που μας παρέχει το περιβάλλον του PRISM, η λογική PCTL επεκτείνεται με τέτοιο τρόπο ώστε να επιτρέπει τον ορισμό ιδιοτήτων μέσω της επόμενου τύπου:

$$R_{\sim r}[C^{\leq k}] \mid R_{\sim r}[I^{=k}] \mid R_{\sim r}[\mathrm{F}\Phi] ,$$

όπου $\sim \in \{<, \leq, \geq, >\}$, $r \in \mathrm{IR} \geq 0$, και $k \in \aleph$ με το Φ να αποτελεί μια PCTL φόρμουλα κατάστασης. Οι περιγραφείσες φόρμουλες διερμηνεύονται ως ακολούθως: *Μια κατάσταση s ικανοποιεί τον τύπο* $R_{\sim r}[C^{\leq k}]$ *εάν, από αυτή την κατάσταση s, το αναμενόμενο συσσωρευτικό κόστος μετά από k βήματα χρόνου ικανοποιεί το* $\sim r$*;*

Ο τύπος $R_{\sim r}[I^{=k}]$ θα είναι αληθής εάν από μια κατάσταση s το αναμενόμενο κόστος σε ένα βήμα χρόνου κ βρίσκεται στο όριο του $\sim r$ και ο τύπος $R_{\sim r}[\mathrm{F}\Phi]$ θα

είναι αληθής εάν από μια κατάσταση s το αναμενόμενο *συσσωρευτικό κόστος* πριν η κατάσταση ακριβώς ικανοποιήσει το Φ βρίσκεται στο όριο του $\sim r$. Τέλος στην προτεινόμενη ανάλυση διεξάγουμε πιθανοκρατικά πειράματα πάνω στον συνολικό χώρο των καταστάσεων, μέσω φυσικά της λογικής PCTL. Με βάση τον αυτοματοποιημένο πιθανοκρατικό ελεγκτή μοντέλων, παράγονται συγκριτικά αποτελέσματα από μια ή περισσότερες ιδιότητες που θέλουμε να επαληθεύσουμε το μοντέλο μας, υπό την μορφή διαγραμμάτων.

6.5 Το μοντέλο πιθανοκρατικού εισβολέα ΠΕ

Είναι γενικά αποδεκτό με βάση το [66] ότι ένας αποτελεσματικός εισβολέας-επιτιθέμενος για την ανάλυση επιθέσεων άρνησης εξυπηρέτησης σε πρωτόκολλα ασφαλείας, μπορεί να είναι πιο αδύναμος από τον τυπικό εισβολέα των Dolev και Yao, ο οποίος τελευταίος χρησιμοποιείται τα τελευταία χρόνια στην επαλήθευση ιδιοτήτων επαλήθευσης. Κάτι τέτοιο δικαιολογείται και από το γεγονός ότι στις επιθέσεις αυθεντικοποίησης δεν συγκαταλέγονται άμεσα οι επιθέσεις που διεξάγει ένας εισβολέας δημιουργίας DoS καταστάσεων. Το πρόβλημα σχεδιασμού ενός εισβολέα ο οποίος εκτελεί ενέργειες –τις ελάχιστες δυνατόν– για την δημιουργία DoS επιθέσεων, έγκειται στην ακριβή κατανόηση της λειτουργίας (από την αρχή μέχρι το τέλος) ενός μηχανισμού αντίστασης σε DoS επιθέσεις. Με βάση το [66], μια DoS επίθεση *χαρακτηρίζεται από συγκεκριμένες ενέργειες εισβολέων να αποτρέψουν έντιμες οντότητες από την χρήση μιας προσφερόμενης υπηρεσίας*. Έτσι το αποτέλεσμα μιας DoS επίθεσης μπορεί να θεωρηθεί σαν την απουσία (κατά κάποιο τρόπο) *ιδιοτήτων προόδου*, οι οποίες αναμένονται να ικανοποιούνται από το συγκεκριμένο μηχανισμό αντίστασης. Στο [35], ο συγγραφέας χαρακτηρίζει τέτοιες ιδιότητες ως «αυτοελεγχόμενες» ιδιότητες βιωσιμότητας, τάσσοντάς τες σαν μια υποκατηγορία των ιδιοτήτων βιωσιμότητας όπου ο επιτιθέμενος δεν μπορεί να τις ελέγξει απόλυτα.

Όπως αναφέρθηκε και στο 2ο κεφάλαιο, το τυπικό μοντέλο εισβολέα των Dolev και Yao παραθέτει ισχυρές προϋποθέσεις για την λειτουργικότητα του εισβολέα, οι οποίες επηρεάζουν άμεσα την ανάλυση των εγγυήσεων ασφαλείας (όπως αυτές της μυστικότητας του μηνύματος ή της αυθεντικοποίησης μια

οντότητας). Τέτοιες υποθέσεις όμως καθιστούν την περαιτέρω ανάλυση άλλων ιδιοτήτων ασφαλείας ανέφικτη, αφού ο εισβολέας βρίσκεται ανήμπορος να διεξάγει ενέργειες κατάλληλες προς την δημιουργίας απειλών ως προς την βιωσιμότητα [23] (και καθ' επέκταση επιθέσεις DoS).

Ένας τυπικός Dolev – Yao εισβολέας έχει πλήρη έλεγχο του επικοινωνιακού μέσου ανάμεσα στις συμμετέχουσες οντότητες του πρωτοκόλλου, συμπεριφερόμενος ως μια μη-ντετερμινιστική διεργασία που μπορεί να επιχειρήσει οποιοδήποτε είδος επίθεσης. Ένα πρωτόκολλο θεωρείται ασφαλές *εάν και μόνο αν δεν υπάρχουν πιθανές διεμπλοκές ενεργειών που μπορούν να οδηγήσουν σε ρωγμή ασφαλείας*. Παρόλα αυτά, η παρουσία του μη-ντετερμινισμού οδηγεί στο γεγονός ότι ιδιότητες βιωσιμότητας δεν μπορούν να ελεγχθούν, εάν πρώτα δεν θεωρηθεί ότι πληρούνται οι βασικές ιδιότητες δικαιοσύνης. Από αυτή την προοπτική, με την προϋπόθεση ότι οι ιδιότητες δικαιοσύνης μπορούν να θεωρηθούν σαν μια αφαίρεση πιθανοκρατικής συμπεριφοράς, φαίνεται πιο φυσιολογικό να επενδύσει κανείς σε μια προσέγγιση πιθανοκρατικού ελέγχου μοντέλων, χωρίς να εκλάβει τις όποιες υποθέσεις δικαιοσύνης, οι οποίες και δεν είναι έγκυρες για όλες τις ικανότητες ενός εισβολέα, προερχόμενου από τον εισβολές των Dolev – Yao. Μερικά άλλα ενδεικτικά σημεία που ενισχύουν την επιλογή του πιθανοκρατικού ελέγχου μοντέλων, είναι τα εξής:

- Η απαίτηση για μοντελοποίηση ενός εισβολέα με την ικανότητα να αποστέλλει τυχαία επιλεγμένα μηνύματα ή για μοντελοποίηση μερικών πολύπλοκων (αλλά πιθανοκρατικού πολυωνύμου χρόνου) διεργασιών με σκοπό την εξαγωγή μιας επίθεσης βασισμένης σε προηγούμενο κρυφάκουσμα (eavesdropping) μηνυμάτων.
- Η ανάγκη για την μοντελοποίηση πιθανοκρατικών διεργασιών επιλογής παραμέτρων επίθεσης, των οποίων οι τιμές επηρεάζουν την ασφάλεια του πρωτοκόλλου.
- Η ανάγκη για την εύστοχη ικανότητα εντοπισμού μιας DoS επίθεσης που σχετίζονται με τις πιθανοκρατικές επιλογές ενεργειών των συμμετεχόντων του πρωτοκόλλου και τις ικανότητας του επιτιθέμενου να παρατηρεί τις ενέργειες αυτές (για παράδειγμα η πιθανότητα για μια οντότητα του πρωτοκόλλου να επιλέξει την επανέκδοση μιας αίτησης

προς εξυπηρέτηση, η οποία και στο παρελθόν είχε απορριφθεί από τον διαχειριστή).

Η προτεινόμενη προσέγγιση για την δημιουργία ενός DoS μοντέλου εισβολέα, βασίζεται σε μια '*ανοιχτή προς κλείσιμο*' (open-ended) βάση τακτικών επιθέσεων η οποία αρχικά είχε προταθεί στο [7], όπου ο αναλυτής έχει δύναται να επιλέξει το σωστό σύνολο ικανοτήτων επιθέσεων για την διερεύνηση ενός πιθανού προβλήματος DoS. Οι επιλεγμένες ενέργειες επιθέσεων (attack actions) συνδυάζονται σε ένα μοναδικό δομοστοιχείο στο περιβάλλον του PRISM, όπου ο αναλυτής προσαρτεί σε κάθε ενέργεια *τιμές κόστους οι οποίες εξαρτώνται από την κατανάλωση πόρων που προκαλεί η συγκεκριμένη ενέργεια, καθώς και από κάποιους περιορισμούς κόστους που τίθενται στον εισβολέα*.

Με την ίδια λογική, παρόμοια κόστη ενεργειών προσαρτώνται στις έντιμες οντότητες του πρωτοκόλλου. Σε κάθε περίπτωση, οι τιμές κόστους αναφέρονται και εδώ στους ίδιους προς κατανάλωση πόρους, που μπορούν να αντιπροσωπεύουν εύρος επικοινωνίας μιας οντότητας (bandwith), διαθέσιμη μνήμη ή υπολογιστική ισχύ. Ας υποθέσουμε ότι το υπό εξέταση πρωτόκολλο μπορεί να είναι ευάλωτο σε μια μορφή επιθέσεως κατανάλωσης όλων των διαθέσιμων πόρων του. Για ένα μήνυμα πρωτοκόλλου το οποίο απαιτεί u εντολές-ισχύος (instructions of computation) από μια οντότητα πρωτοκόλλου, έχοντας την δυνατότητα να επεξεργαστεί v εντολές το δευτερόλεπτο, το υπολογιστικό κόστος θα κυμανθεί σε u/v δευτερόλεπτα ανά ισχύς.

Ένα εισβολέας για την διερεύνηση DoS επιθέσεων, χρησιμοποιεί έναν σταθερό αριθμό N *μηχανών-οντοτήτων* που αποκαλούνται *μηχανές-οντότητες φαντάσματα* (zombie machines), μιας και ως προς τα τεχνικά χαρακτηριστικά τους είναι ίδιες με τις έντιμες οντότητες του υπό εξέταση πρωτοκόλλου.

Οι οντότητες-φαντάσματα μπορούν και δημιουργούν *διεφθαρμένα* (bogous) μηνύματα του πρωτοκόλλου, ικανά να ξεγελάσουν έντιμες οντότητες του πρωτοκόλλου με σκοπό την πλήρη εξάντληση των πόρων τους, πριν καν οι έντιμες οντότητες συνειδητοποιήσουν ότι πρόκειται για λάθος μηνύματα απορρίπτοντας την όποια αίτηση και εν τέλει την σύνοδο επικοινωνίας με τις οντότητες-φαντάσματα. Εναλλακτικά, αντί να θεωρήσουμε N οντότητες-φαντάσματα, μπορούμε να μεταβούμε στην σχεδίαση ενός ισχυρού εισβολέα με ικανότητες υποκλοπής ταυτοτήτων (identity spoofing abilities), όπου και θα του

επιτρέψουν την συνολική προσομοίωση της συμπεριφοράς των N οντοτήτων-φαντάσματα.

Η δριμύτητα όλων των πιθανοτήτων εξάντλησης των διαθέσιμων πόρων, μπορεί αρχικά να αποδοθεί στον υπολογισμό όλων των ασύμμετρου χαρακτήρα διαθέσιμων πόρων, μεταξύ του εισβολέα (συμπεριλαμβανομένου των δικών του περιορισμών κόστους) και των έντιμων οντοτήτων του πρωτοκόλλου. Εάν υποθέσουμε ότι κάθε οντότητα-φάντασμα βρίσκεται συνδεδεμένη στο διαδίκτυο μέσω μιας σύνδεσης με εύρος επικοινωνίας x δυφία ανά δευτερόλεπτο και μια οντότητα διαχειριστή που είναι συνδεδεμένη στο διαδίκτυο με εύρος X δυφία ανά δευτερόλεπτο, τότε οι οντότητες φαντάσματα μπορούν να επιφέρουν κορεσμό στο εύρος της επικοινωνίας του διαχειριστώ, όταν $Nx \geq X$.

Έστω μια οντότητα-φάντασμα συνδέεται στο διαδίκτυο με εύρος δυφίων x ανά δευτερόλεπτο και μια οντότητα-διαχειριστή με σύνδεση εύρους X δυφίων ανά δευτερόλεπτο, τότε οι οντότητες φαντάσματα μπορούν να επιφέρουν έναν κορεσμό στο διαθέσιμο εύρος της οντότητας διαχειριστή όπου $Nx \geq X$. Έχοντας υπ' όψιν την διαθέσιμη υπολογιστική ισχύ της οντότητας-διαχειριστή, αυτή μπορεί να καταναλωθεί πλήρως όταν $N \cdot k_s/m \geq u_s/v$, όπου k_s αποτελεί το συνολικό αριθμό εντολών, σε μία σύνοδο πρωτοκόλλου για τον εισβολέα και u_s ο αριθμός των εντολών για την οντότητα-διαχειριστή. Το πρόβλημα εντοπισμού μιας DoS απειλής έγκειται στην σύγκριση του ασύμμετρου σχήματος οντοτήτων διαχειριστή/φάντασμα εάν το αναγκαίο εύρος είναι μεγαλύτερο σε σχεσιακή αναλογία με την απαιτούμενη υπολογιστική ισχύ, δηλαδή εάν:

$$\frac{u_s / v}{k_s / m} < \frac{X}{x}$$

Η προσέγγιση η οποία ακολουθούμε εφαρμόζεται λαμβάνοντας υπ' όψιν τις τιμές κόστους για τις πιο επιβλαβείς-σε σχέση με την ισχύ- ενέργειες των οντοτήτων και του εισβολέα.

Ορισμός 5. *Ορίζουμε την ιδιότητα αντοχή σε επίθεση άρνησης εξυπηρέτησης (DoS Resistance) ως την πιο χαμηλή πιθανότητα για έναν επιτιθέμενο που προσομοιώνει μια τέτοια απειλή, να καταφέρει να αποτρέψει έντιμες οντότητες του πρωτοκόλλου να χρησιμοποιήσουν προσφερόμενες υπηρεσίες του πρωτοκόλλου.*

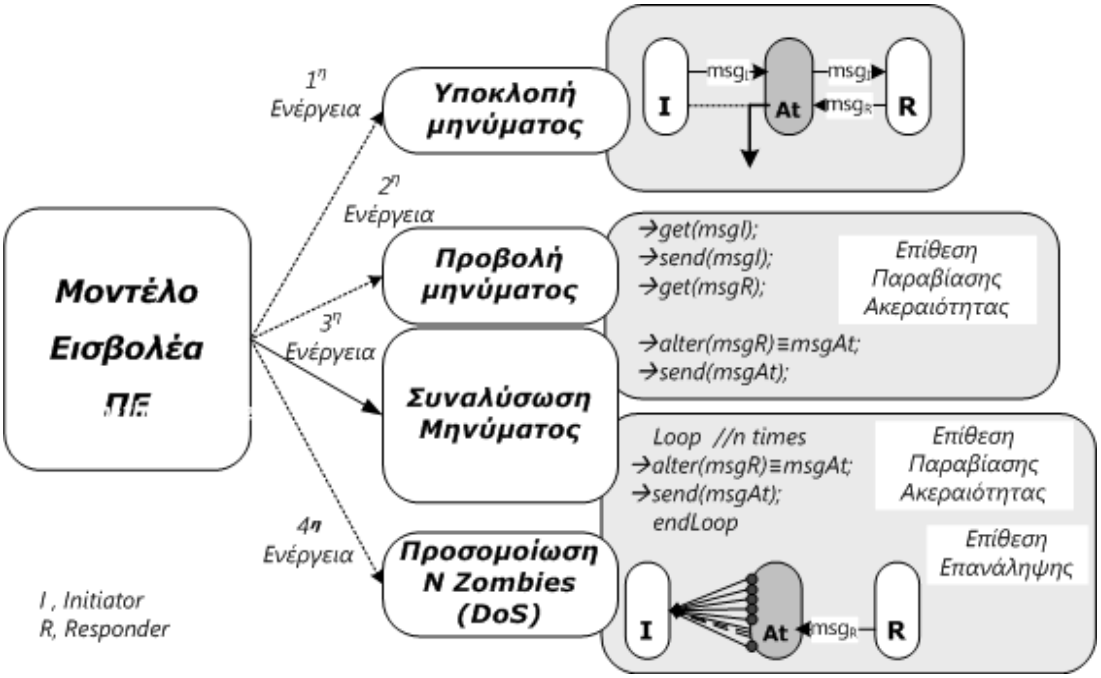
Σε αυτό το είδος της ανάλυσης είναι πολύ σημαντικό να ορισθούν επακριβώς οι παράμετροι επικοινωνίας του πρωτοκόλλου, για κάθε έναν συμμετέχοντα, έτσι ώστε να καταγράφεται –για οποιαδήποτε ενέργεια εκτελείται- το σύνολο των πόρων που καταναλώνονται. Η προστασία στις επιθέσεις DoS, συνήθως βασίζεται σε *cookie-based* [52] ή *client puzzle* [4] μηχανισμούς, όπου ένας συμμετέχοντας του πρωτοκόλλου στέλνει ένα "cookie" (λ.χ. μια unforgeable βασισμένη σε κλειδί συνάρτησης κατακερματισμού τιμή πληροφορίας εντοπισμού της σύνδεση) ή ένα 'puzzle' στην επικοινωνούσα οντότητα, με σκοπό την δημιουργία μιας συνόδου αμοιβαίας εμπιστοσύνης. Η ιδέα είναι ότι η δεύτερη οντότητα (ανταποκριτής) πρέπει να μην μεταβαίνει σε περαιτέρω καταστάσεις (προφύλαξη από εξάντληση μνήμης), αρνούμενος να εκτελέσει ενέργειες υψηλού κόστους όπως κρυπτογράφηση δεδομένων (προφύλαξη από εξάντληση υπολογιστικής ισχύος), μέχρι να επαληθευτεί η έντιμη ταυτότητα του αποστολέα (εναρκτήριος).

Με βάση τα παραπάνω, αναγνωρίζονται τρεις βασικές στρατηγικές που μπορεί να ακολουθήσει ένας εισβολέας για την δημιουργία DoS απειλών:

- *Counterfeiting*: Ο εισβολέας αποστέλλει μαζικά διεφθαρμένα cookies, puzzles ή puzzle επιλύσεις.
- *Time Shifting*: Ο εισβολέας προετοιμάζεται για μια επίθεση υπολογίζοντας διεφθαρμένα διαμοιραζόμενα μυστικά (είτε επιλύοντας προηγούμενα ληφθέντα puzzles ή χειραγωγώντας cookies), με σκοπό την χρήση τους σε μια συνολική DoS επίθεση.
- *Message replays*: Ο εισβολέας μπορεί να αποστείλει το ίδιο επαληθευμένα σωστό cookie ή puzzle (το οποίο έχει λάβει σε προηγούμενη σύνοδο του πρωτοκόλλου) πολλές φορές, προς μια οντότητα του πρωτοκόλλου.

Όλες οι προαναφερθείσες στρατηγικές επιθέσεων DoS που περιγράφθηκαν υποθέτουν ότι το χρησιμοποιούμενο μοντέλο εισβολέα μπορεί και επιχειρεί τρεις βασικές ενέργειες οι οποίες είναι α) *message interception*, β) *message projection* και γ) *message concatenation*. Με βάση τις ενέργειες αυτές, ο αναλυτής μπορεί να συνθέσει απλές επιθέσεις επιλέγοντας τις κατάλληλες τακτικές επιθέσεων (attack tactics) όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο για το μοντέλο ΕΠΕ. Με αυτό τον τρόπο ο εισβολέας συνδυάζει τις διαθέσιμες επιθέσεις *tactics* (deflection, message integrity violation and straight

replay), οποίες έχουν οριστεί και υλοποιηθεί στο [7], ενοποιώντας όλες τις απαραίτητες ενέργειές του σε ένα μοναδικό δομοστοιχείο στο PRISM, με όλες τις τιμές κόστους για κάθε μια ενέργεια.



Εικόνα 6.5.1 Μια DoS απειλή με counterfeiting μηνυμάτων για N οντότητες φαντάσματα

Στην εικόνα 6.5.1, παρουσιάζεται μια αποτύπωση υψηλού επιπέδου μιας DoS απειλής πάνω σε ένα πρωτόκολλο ασφαλείας (όπου στην δικιά μας ανάλυση είναι το HIP). Σε αντίθεση με άλλες αναλύσεις ασφαλείας που έχουν διεξαχθεί για το πρωτόκολλο HIP [49], οι οποίες ασχολούνται με την αποτελεσματικότητα ενός DoS μηχανισμού για την οντότητα ανταποκριτή (Responder, *R*), αυτή η ανάλυση αντιστρέφει την όλη φιλοσοφία της με την ανάλυση ασφαλείας DoS απειλής για την εναρκτήρια (Initiator, *I*) οντότητα. Κάτι τέτοιο μπορεί να συμβεί όταν η εναρκτήρια-οντότητα σε μια σύνοδο πρωτοκόλλου παίζει και το ρόλο της ανταποκρινόμενης οντότητας σε μια άλλη σύνοδο του HIP. Ο επιτιθέμενος (Attacker, *At*) παρεμβάλλεται και λαμβάνει όλα τα μηνύματα που ανταλλάσσονται μεταξύ της εναρκτήριας οντότητας (Initiator, *I*) και της οντότητας ανταποκριτή (Responder, *R*). Στη συνέχεια ο *At* σε κάθε μήνυμα διαφθείρει το puzzle το οποίο περιέχεται στο μήνυμα msg_R διεξάγοντας ένα απλό concatenation μηνύματος, δημιουργώντας με αυτό τον τρόπο *N* μηνύματα-φαντάσματα, τα οποία και αποστέλλονται στην οντότητα *I*. Για την δημιουργία αυτού του σεναρίου επίθεσης χρησιμοποιούνται οι τακτικές

επιθέσεων *παραποίηση μηνύματος* (message integrity violation) και *απευθείας επανάληψη μηνύματος* (straight message replays).

6.6 Ανάλυση DoS Επιθέσεων στο πρωτόκολλο HIP

Στην παράγραφο αυτή θα παρουσιαστεί η υλοποίησης του πρωτοκόλλου HIP και του μοντέλου του πιθανοκρατικού εισβολέα ΠΕ, ο οποίος θα συμβολίζεται με *At*.

6.6.1 Εισαγωγή στο πρωτόκολλο Host Identity Protocol (HIP)

Σε αυτό το σημείο του κεφαλαίου θα περιγραφθεί το πρωτόκολλο Host Identity Protocol (HIP) το οποίο και επιλέχθηκε για να διεξαχθεί και να δοκιμαστεί η αποτελεσματικότητα της προτεινόμενης μεθόδου, με την βοήθεια του πιθανοκρατικού ελέγχου μοντέλων και το περιβάλλον του PRISM.

Κύριος στόχος του HIP [50] αποτελεί ο διαχωρισμός των host identifiers από τις τοποθεσίες τους κατά την λειτουργία του IPv4 και IPv6. Σε μια τυπική δομή του TCP/IP, οι διευθύνσεις IP εξυπηρετούν και τις δύο εκδόσεις, κάτι το οποίο εισαγάγει νέους περιορισμούς όσον αφορά το mobility και το multi-homing. Το HIP αποτελεί ένα καινούργιο ‘στρώμα’ πρωτόκολλο το οποίο παρεμβάλλεται μεταξύ των επιπέδων network και transport (στο μοντέλο αναφοράς OSI), με σκοπό την αποτύπωση των host identifiers σε διευθύνσεις του δικτύου και αντίστροφα.

Παρόλα αυτά, το HIP αντιπροσωπεύει ένα πρωτόκολλο ασφαλείας το οποίο ορίζει τους host identifiers για την μετονομασία των επικοινωνιακών σημείων διεξάγοντας μια αυθεντικοποίηση ανάμεσα και στις συσχετίσεις ασφαλείας από το IPSec. Ο βασικός πυρήνας του HIP είναι υλοποιημένος πάνω σε μια κλασσική Diffie-Hellman αυθεντικοποίηση κλειδιού, σε μια προσπάθεια για την δημιουργία μιας επικοινωνιακής συνόδου μεταξύ των προαναφερθέντων σημείων.

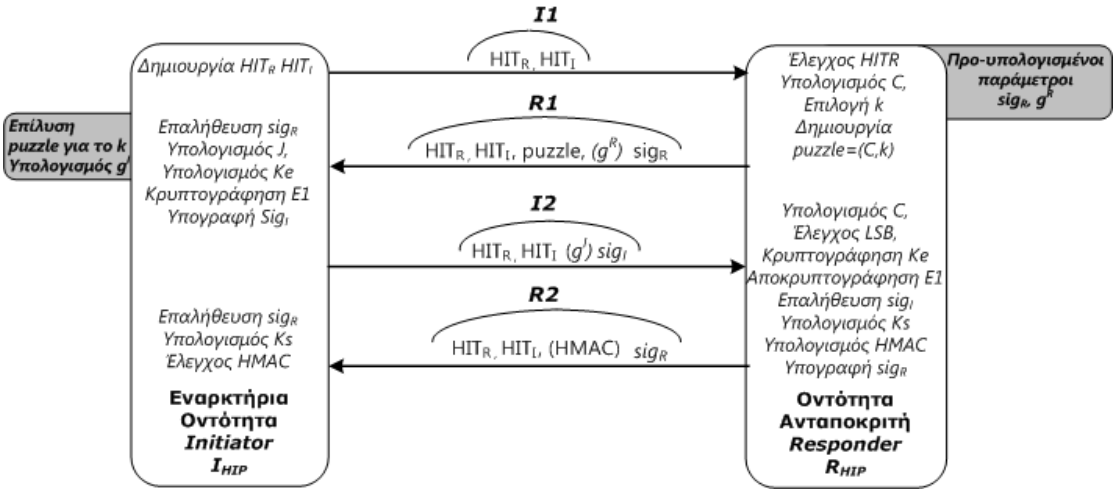
Εάν το πρωτόκολλο HIP ολοκληρώσει επιτυχώς την λειτουργία του, όλα τα πακέτα δεδομένων προστατεύονται από μια επικεφαλίδα ενθυλακωμένης ασφαλείας (Encapsulating Security Payload, ESP). Εξαρτώμενη από τις παραμέτρους που έχουν επιλεχθεί και από τα δύο επικοινωνιακά σημεία (με

βάση την χρονική διάρκεια δημιουργίας της παραπάνω συνόδου) το χρησιμοποιούμενος ESP μπορεί να παρέχει εμπιστευτικότητα, αυθεντικοποίηση της προέλευσης του μηνύματος, ακεραιότητα συνδεσιμότητας, υπηρεσίες μη-επαναληπτικότητας μηνυμάτων και μερική εμπιστευτικότητα της διακινούμενης ροής δεδομένων. Στο HIP επίσης, το host identity (HI) των οντοτήτων που συμμετέχουν στο πρωτόκολλο, παίζουν το ρόλο ενός δημόσιου κλειδιού κρυπτογράφησης: ο χρησιμοποιούμενος τύπος αναγνώρισης (identifier) μπορεί να χρησιμοποιηθεί για την επαλήθευση των ψηφιακών υπογραφών χωρίς να είναι αναγκαία η πρόσβαση και χρήση υποδομών πιστοποιητικών και μηχανισμών ασύμμετρης κρυπτογράφησης. Κάτι τέτοιο συνήθως αναπαριστάται με την host identity ετικέτα, πίνακας 6.6.1 (host identity Tag, HIT), η οποία και είναι μια 128-δυφίων βασισμένη σε συνάρτηση κατακερματισμού τιμή του HI.

Πίνακας 6.6.1 Πίνακας συμβόλων και βασική λειτουργία του πρωτοκόλλου HIP

HIT_I	Αναγνωριστική ετικέτα του I
HIT_R	Αναγνωριστική ετικέτα του R
g^R	Προ-υπολογισμένο μέρος του μηνύματος R1
sig_R	Ψηφιακή υπογραφή R
sig_I	Ψηφιακή υπογραφή I
C	puzzle nonce
k	Βαθμός δυσκολίας του puzzle
J	Επίλυση του puzzle
LSB_k	Συνάρτηση επιστροφή των k least significant δυφίων
K_e,K_s	Παραγόμενα κλειδιά Diffie-Hellman
E_x	Μήνυμα x κρυπτογραφημένο με K_e
HK_s	Συνάρτηση κατακερματισμού με κλειδί K_s
HMAC	Κώδικας μηνύματος αυθεντικοποίησης που υπολογίζεται μέσω του κλειδιού K_s

Όπως παρουσιάζεται και στην εικόνα 6.6.1, ο πυρήνας του HIP βασίζεται στην ανταλλαγή τεσσάρων (4) μηνυμάτων τα οποία και υποτίθεται ότι παρέχουν τον απαιτούμενο μηχανισμό για την προστασία από DoS απειλές. Η εναρκτήρια οντότητα *I* στην αρχή αποστέλλει το πρώτο μήνυμα *I1* το οποίο και περιέχει τις ετικέτες HIT_I and the HIT_R , προς την οντότητα ανταποκριτής *R*. Σημειώνεται ότι όλα τα μηνύματα που ανταλλάσσονται περιέχουν και τις δύο προαναφερθείσες ετικέτες μέσα στην επικεφαλίδα των μηνυμάτων.



Εικόνα 6.6.1 Βασικά βήματα του πρωτοκόλλου HIP

Το μήνυμα *R1* είναι μερικώς είναι προϋπολογισμένο από τον ανταποκριτή *R*, ακόμα και πριν την λήψη του μηνύματος *I1*. Το κομμάτι του μηνύματος αυτό (g^R) περιλαμβάνει α) το HIT_R, β) το Diffie-Hellman κλειδί της οντότητας ανταποκριτή, γ) το αναγνωριστικό HI του ανταποκριτή, δ) τους προτεινόμενους κρυπτογραφικούς αλγορίθμους για τα επόμενα βήματα ολοκλήρωσης του HIP, ε)οι προτεινόμενοι μετασχηματισμοί ESP και στ) μια αίτηση τύπου-ηχό (echo request). Το τελευταίο (στ) μέρος χρησιμοποιείται για την αποθήκευση κάποιων από τα δεδομένα, προσπαθώντας να γλιτώσει πόρους μνήμης από την πλευρά του ανταποκριτή *R*.

Ο ανταποκριτής *R* υπογράφει μέρος αυτό του μηνύματος *R1* με την υπογραφή sig_R. Όλο το υπόλοιπο μέρος του *R1* όπως το κρυπτογραφημένο puzzle και η ετικέτα του *I*, HIT_I μένει ως έχει, χωρίς να προστατεύεται από την υπογραφή sig_R. Ένας host μπορεί να λάβει περισσότερα από ένα μηνύματα *R1*, είτε εξαιτίας της αποστολής πολλαπλών μηνυμάτων *I1* ή μιας πιθανής επανάληψης ενός *R1*. Το χρησιμοποιούμενο puzzle αποτελείται από τρία ξεχωριστά μέρη: α) το puzzle nonce *C*, τον βαθμό δυσκολίας *k* και την ανταποκρινόμενη επίλυση *J*. Η επίλυση του puzzle επαληθεύεται ως εξής: υπολογίζεται η κατακερματισμένη τιμή SHA-1 του concatenation του *C*, του HIT_I, του HIT_R και της *J*. Στη συνέχεια γίνεται ο έλεγχος για το εάν τα *k* χαμηλής τάξης δυφία της κατακερματισμένης τιμής είναι όλα μηδέν (0). Θα έχουμε :

$$LSB_k(\text{SHA-1}(C \mid HIT_I \mid HIT_R \mid J), k) == 0$$

Ενώ η εναρκτήρια οντότητα *I* επιχειρεί μια 'βίαιη' (brute-force) αναζήτηση για το *J* όπου επαναλαμβάνεται περίπου $O(2^k)$ φορές, ο ανταποκριτής *R* επαληθεύει την λύση με τον υπολογισμό μιας απλής τιμής κατακερματισμού (προστασία από εξάντληση όλων των πόρων υπολογιστικής ισχύος). Με την λήψη του *R1* η εναρκτήρια οντότητα ελέγχει εάν έχει στείλει το αντίστοιχο *I1* και επαληθεύει την υπογραφή χρησιμοποιώντας το *HI* του ανταποκριτή. Στη συνέχεια, επιλύει το puzzle και δημιουργεί το μήνυμα *I2* το οποίο περιλαμβάνει τα HIT_I, HIT_R και ένα ψηφιακός υπογεγραμμένο μέρος του g^I.

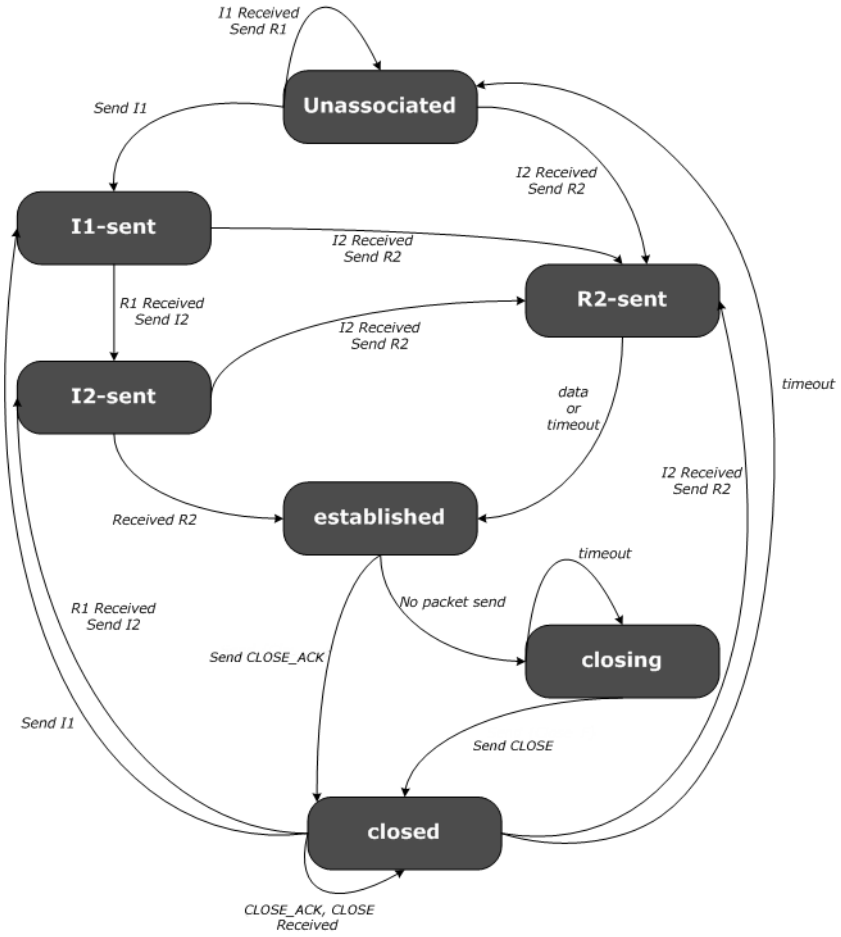
Το υπογεγραμμένο αυτό μέρος περιέχει α) το *puzzle* και τη λύση του β) το Diffie-Hellman κλειδί της εναρκτήριας οντότητας γ) τους μετασχηματισμούς ESP που προτείνονται από την εναρκτήρια οντότητα δ) το δημόσιο κλειδί του *I* (*HI*) κρυπτογραφημένο με χρήση του κλειδιού K_e, το οποίο φαίνεται ως E_1, ε) ένας πίνακας με τις παραμέτρους ασφαλείας για την συσχέτιση των οντοτήτων Εναρκτήριου-Ανταποκριτή στ) μια απάντηση 'ηχώ' η οποία παράγεται εξαιτίας της προηγούμενης αίτησης.

Με την λήψη του μηνύματος *I2*, ο ανταποκριτής επαληθεύει την λύση του puzzle, αποκρυπτογραφεί το *E1* το οποίο περιέχει το *HI* της εναρκτήριας οντότητας, επαληθεύει την ψηφιακή υπογραφή πάνω στο *I2* και υπολογίζει το καινούργιο κλειδί συνόδου K_s. Για την εναρκτήρια οντότητα, η σύνοδος του HIP ολοκληρώνεται με την λήψη του *R2* η οποία επιτρέπει την επαλήθευση του HMAC (Hash-Based Message Authentication Code) και της υπογραφής. Ο τερματισμός του πρωτοκόλλου υποδεικνύεται από την αποστολή και λήψη ενός μηνύματος *Close* και *Close_Ack* αντίστοιχα.

Εάν ένας *host* λάβει ένα μη αναμενόμενο μήνυμα, τότε το μήνυμα αυτό απορρίπτεται. Επίσης μια υλοποίηση του πρωτοκόλλου HIP προϋποθέτει την ανεξάρτητη απόρριψη μιας συνόδου (ακόμα και έντιμης) εάν οι πολιτικές ασφαλείας που έχουν ορισθεί προτείνουν μια τέτοια ενέργεια, βάσει προκαθορισμένων παραμέτρων. Η προσέγγιση που παρουσιάζεται σε αυτό το κεφάλαιο του πιθανοκρατικού ελέγχου μοντέλων επιτρέπει τον έλεγχο και την επαλήθευση πιθανών DoS επιθέσεων, με σκοπό τον σχεδιασμό καλύτερων και ανθεκτικότερων μηχανισμών αντοχής σε τέτοιου είδους επιθέσεις.

6.6.2 Το μοντέλο πρωτοκόλλου HIP στο PRISM

Το μοντέλο του PRISM για το πρωτόκολλο του HIP, βασίζεται σε μια μηχανή κατάστασης το οποίο παρουσιάζεται στην εικόνα 6.6.2.



Εικόνα 6.6.2. Το διάγραμμα μετάβαση καταστάσεων του HIP

Η μηχανή αυτή περιλαμβάνει όλες τις διεργασίες της εναρκτήριας οντότητας *I* και του ανταποκριτή *R* σε μια μόνο όψη του προς υλοποίηση συστήματος. Οι καταστάσεις των οντοτήτων του πρωτοκόλλου ορίζονται στη συνέχεια στον πίνακα 1. Παρατηρούμε ότι υπάρχουν καταστάσεις που προϋποθέτουν μια χρονική περίοδο αναμονής για την λήψη ενός συγκεκριμένου μηνύματος το οποίο εστάλη από την σωστή συμμετέχουσα οντότητα του πρωτοκόλλου. Για παράδειγμα η κατάσταση με την ονομασία `I1-sent` προσεγγίζεται είτε από την κατάσταση `Unassociated` ή την `Closed` μετά από την αποστολή του μηνύματος *I1*, όπου και σε αυτή τη κατάσταση ο συμμετέχοντας περιμένει για

ένα μήνυμα *R1* ή *I2*. Η λήψη ενός εξ' αυτών θα πυροδοτήσει τις ενέργειες των αντίστοιχων μεταβάσεων όπως φαίνεται και στην εικόνα 6.6.2.

Πίνακας 6.6.2 Πίνακας 1 Καταστάσεις των Οντοτήτων του HIP

Κατάσταση	**Περιγραφή**
Μη Συσχετιζόμενη (Unassociated)	Έναρξη της μηχανής κατάστασης (State machine start)
Αποστολή I1 (I1-sent)	Έναρξη του HIP πρωτοκόλλου (Initiating base exchange)
Αποστολή I2 (I2-sent)	Αναμονή (Waiting to complete base exchange)
Αποστολή R2 (R2-sent)	Αναμονή (Waiting to complete base exchange)
Καθιέρωση (established)	Ίδρυση HIP Σύνδεσης (HIP association established)
Προς Κλείσιμο (Closing)	Σύνοδος HIP προς Κλείσιμο (HIP association closing, no data can be sent)
Κλειστή (closed)	Σύνοδος HIP Κλειστή (HIP association closed, no data can be sent)

Οι καταστάσεις `Establish`, `Close` και `closing` αναπαριστούν την επιτυχής ολοκλήρωση του πρωτοκόλλου HIP, η κάθε μία για ξεχωριστές συσχετίσεις ασφαλείας που περιλαμβάνει το συγκεκριμένο πρωτόκολλο.

Το μοντέλο του PRISM για το HIP αποτελείται από τέσσερα δομοστοιχεία (modules): (*i*) το μέσο (Medium) *m* το οποίο αναπαριστά το επικοινωνιακό κανάλι που χρησιμοποιείται από τις οντότητες του πρωτοκόλλου για την ανταλλαγή μηνυμάτων (*ii*) την εναρκτήρια οντότητα (Initiator) *I*, (*iii*) την οντότητα ανταποκριτή (Responder) *R*, (*iv*) το μοντέλο του εισβολέα-επιτιθέμενου (Attacker) *At*. Κατά την διάρκεια της παράλληλης σύνθεσης του χώρου των καταστάσεων, τα προαναφερθέντα δομοστοιχεία αλληλεπιδρούν ανανεώνοντας (όπου χρειάζεται) τις τοπικές μεταβλητές τους, που αναπαριστούν τα χαρακτηριστικά τους. Αυτές οι ανανεώσεις (updates) ανταποκρίνονται στις μοντελοποιημένες μεταβάσεις των καταστάσεων.

Η εικόνα 6.6.3 παρουσιάζει τις δηλώσεις των καθολικών χρησιμοποιούμενων μεταβλητών του μοντέλου. Υποθέτοντας ότι το επικοινωνιακό κανάλι λειτουργεί ουσιαστικά σαν μια μνήμη μηνυμάτων, μοντελοποιούμε τρεις πιθανές καταστάσεις για το μέσο *m*, εικόνα 6.6.4 :

- Κανένα μήνυμα (C1=2 & C2=2) – δεν υπάρχει μήνυμα προς αποστολή
- 1ος προορισμός μηνύματος (C1=1 & C2=2) – η εναρκτήρια οντότητα (ή ο εισβολέας) στέλνουν ένα μήνυμα στην οντότητα ανταποκριτής

- 2ος προορισμός μηνύματος (C1=2 & C2=1) – η οντότητα ανταποκριτής (ή ο εισβολέας) στέλνουν ένα μήνυμα στην εναρκτήρια οντότητα

Δύο μεταβλητές σχετίζονται με τις παραμέτρους υλοποίησης του HIP (*i*) η μεταβλητή *proc_limit* αναπαριστά τον αριθμό των μηνυμάτων που εξυπηρετούνται ταυτόχρονα από την εναρκτήρια οντότητα και (*ii*) η μεταβλητή *B* η οποία αντιπροσωπεύει τον μέγιστο αριθμό μηνυμάτων που μπορούν να βρεθούν στην ουρά της εναρκτήριας οντότητας.

```
dtmc
const int MAX_TIME=2000;                //max time of the protocol's session
const int INIT_=10;                     //time for initiating/sending first message
const int PR_I1_GEN_R1=15;              //processing time of I1 and generation of R1
const double PR_R1_GEN_I2_k5=0.000934;  //processing time of R1 and generation of I2 with R1(puzzle(k)=5)
const double PR_R1_GEN_I2_k10=0.0225;   //processing time of R1 and generation of I2 with R1(puzzle(k)=10)
const double PR_R1_GEN_I2_k15=0.808;    //processing time of R1 and generation of I2 with R1(puzzle(k)=15)
const double PR_R1_GEN_I2_k20=15.3;     //processing time of R1 and generation of I2 with R1(puzzle(k)=20)
const double PR_R1_GEN_I2_k25=630;      //processing time of R1 and generation of I2 with R1(puzzle(k)=25)
const int PR_I2_GEN_R2=170;             //processing time of I2 and generation of R2
const int PR_R2=9;                      //processing time of R2
const int TRNSMT_=10;                   //transmition cost
const int TRNSMT_INTR=9;                //attacker's transmition cost


//Formulas being used according to the HIP standards [8]

formula no_action = c1=2 & c2=2;                        //no action over the channel
formula sending_init = c1=1 & c2=2;                     //Initiator sending
formula sending_resp = c1=2 & c2=1;                     //Responder sending

formula message1 = HITi=1 & HITr_sent=1;                //creation of message I1
formula message1_int = HITi_stolen=1 & HITr_stolen=1;   //message I1 for the attacker
formula message2 = HITi_received=1 & HITr=1 & puzzle=true & HITr=1 & gr=1 & HITi_received=1 & signr1=1;

//Constants
const int M = 50;                       //number of distinct counterfeiting messages
const int B = 80;                       //limit for the available queue resource of the Initiator
const int proc_limit = 40;              //number of messages to be processed
const int MAX = 40;                     //number of messages processed simultaneously
```

Εικόνα 6.6.3.Καθολικές μεταβλητές για το μοντέλο HIP

Τέλος η μεταβλητή *M* εκφράζει τον αριθμό των ξεχωριστών μηνυμάτων που δημιουργεί ο εισβολέας- επιτιθέμενος *At* με τον τρόπο που δείχνεται στην εικόνα 6.5.1. Οι δηλώσεις καθολικών μεταβλητών που απεικονίζονται στην εικόνα 6.6.3, περιλαμβάνουν επίσης και τα προσαρτούμενα κόστη επεξεργασίας των μηνυμάτων του HIP για τις συμμετέχουσες οντότητες. Οι τιμές αυτές του κόστους βασίζονται στις απαιτήσεις του κάθε μηνύματος (δεδομένα τα οποία πάρθηκαν από μελέτες πραγματικού χρόνου απόδοσης του HIP [49]) και σε σχετικές διαφορές μεταξύ των αλληλεπιδρώντων οντοτήτων, λαμβάνοντας υπ' όψιν και την εύρος της υπολογιστικής ισχύς του καθενός (λ.χ. ακριβής αριθμός των εντολών επεξεργασίας ανά δευτερόλεπτο).

```
//Medium of the communication network handling all exchanged messages
module medium
c1 : [1..2] init 2;                    //c1=1: no action towards responder
c2 : [1..2] init 2;                    //c1=2: action towards responder
//MESSAGE I1
[m1]c1=2 & c2=2 -> c1'=1;
[finish_m1](c1=1 & c2=2 & s3=1) -> (c1'=2) & (c2'=2);
////////////////
//MESSAGE Ix (either I1 or I2), Attacker->Responder
[intr_1]no_action -> c1'=1;
[finish_intr_1]sending_init & s3=4-> c1'=2;
/////////////////
//MESSAGE R1
[m2]no_action & message2 -> c2'=1;
[finish_m2]sending_resp & s1=4 -> (c1'=2) & (c2'=2);
/////////////////
//SPOOFED MESSAGE Rx (either R1 or R2) , Attacker->Initiator
[spoof]no_action -> c2'=1;
[finish_spoof]sending_resp -> c2'=2;
////////////////
//MESSAGE I2
[m3]no_action -> c1'=1;
[finish_m3]sending_init -> c1'=2;
////////////////
//MESSAGE R2
[sendingr2]no_action -> c2'=1;
[finishr2]sending_resp -> c2'=2;
endmodule
```

Εικόνα 6.6.4 Καθολικές μεταβλητές για το μοντέλο HIP

Το συνολικό μοντέλο του HIP στο PRISM που ενσωματώνει τον συγκεκριμένο εισβολέα με τον μηχανισμό ανίχνευσης DoS απειλών, έγκειται στο γεγονός ότι το *puzzle* που αποστέλλεται στην εναρκτήρια οντότητα δεν συμπεριλαμβάνεται στο κρυπτογραφημένο μέρος του μηνύματος *R1.* Το *puzzle* παράγεται κατ' απαίτηση βασισμένο σε ένα τυχαίο αριθμό (*Nonce*) και μια παράμετρο *k* η οποία καθορίζει την δυσκολία επίλυσής του, το οποίο έχει και την μεγαλύτερη επίπτωση κόστους για την εναρκτήρια οντότητα. Η λογική απαίτηση του πρωτοκόλλου για την παραγωγή καινούργιων *puzzle* μηνυμάτων το προστατεύει από φαινόμενα time-shifting και από απειλές επανάληψης μηνυμάτων που μπορούν να οδηγήσουν σε DoS επιθέσεις. Παρόλα αυτά όμως, κάτι τέτοιο καθιστά το πρωτόκολλο εκτεθειμένο σε counterfeiting επιθέσεις DoS, όπως φαίνεται και στην εικόνα 6.6.1.

Εάν σε μια υλοποίηση του HIP πρωτοκόλλου δεν υπάρχει μέριμνα για αποτροπή της αποδοχής μιας επανάληψης μηνύματος *R1*, ο μοντελοποιημένος εισβολέας *At* μπορεί να αποτρέψει την εναρκτήρια οντότητα από το να επιλύσει το αυθεντικό *puzzle*. Η παραπάνω επίθεση επιτυγχάνει όταν η συχνότητα δημιουργίας και αποστολής των counterfeited μηνυμάτων είναι μεγαλύτερη από

αυτή του ολοκληρωμένου βήματος αποστολής/λήψης τους *R1* μηνύματος από τις συμμετέχουσες οντότητες του HIP. Σε αυτή τη περίπτωση, το πρώτο μήνυμα *R1* που φτάνει στην εναρκτήρια οντότητα από τον εισβολέα (μετά από την αποστολή του μηνύματος *I1*) θα αποτελεί πάντα ένα διεφθαρμένο μήνυμα, ωθώντας την εναρκτήρια οντότητα σε αδυναμία εκπλήρωσης της ιδιότητας ανοχής σε DoS επιθέσεις, η οποία και στο μοντέλο μας θα εξαρτάται από τις τιμές των μεταβλητών *proc_limit*, *B* και *k*.

```
module attacker
s3 : [1..9] init 1;
//1=wait
//2=message I1 received
//3=get hit's
HITi_stolen : [0..1] init 0;
HITr_stolen : [0..1] init 0;
puzzle_received : [0..1] init 0;
puzzle_change : [0..M] init 0;
flag : bool init false;

//GETS I1 AND SENDS IT TO RESPONDER
[finish_m1]s3=1 & message1 & sending_init -> s3'=2;
[]s3=2 -> HITi_stolen'=1 & HITr_stolen'=1 & s3'=3;
[]s3=3 & message1 -> s3'=4;
[intr_1]no_action & s3=4 & flag=false -> s3'=4 & flag'=true;
[finish_intr_1]s3=4 -> s3'=5;
////////////////////////////////

//RECEIVES R1 AND CREATES AND SENDS THE SPOOFED MESSAGES
[finish_m2]s3=5 & message2 & sending_resp ->s3'=6;
[]s3=6 -> 0.5:(a'=10) & (s3'=7) + 0.5:(a'=20) & (s3'=7);
[]s3=7 -> puzzle_received'=1 & s3'=8;
[spoof]s3=8 & puzzle_change<M & no_action -> puzzle_change'=puzzle_change+1 & s3'=9;
[finish_spoof]s3=9-> s3'=8;
////////////////////////////////
endmodule
```

Εικόνα 6.6.5 Το δομοστοιχείο (module) του εισβολέα (*At*) στο PRISM

Η εικόνα 6.6.5 παρουσιάζει τις απαραίτητες εντολές-φρουρούς για τις ενέργειες-μεταβάσεις που θέλουμε να διεξάγει ο εισβολέας *At* στο υλοποιημένο HIP μοντέλο. Όλες οι εντολές έχουν την μορφή *[ ] <guard>! <command>;*. Για περισσότερες πληροφορίες για την σύνταξη της μεταγλώσσας προδιαγραφών PRISM, ο αναγνώστης μπορεί να βρει στο [70]. Οι εντολές αυτές και οι απαραίτητες συνθήκες ενεργοποίησής τους, πυροδοτούν συγχρονισμένες ή μη μεταβάσεις μόνο αν συνθήκη τους, σε οποιοδήποτε σημείο του παραγομένου χώρου καταστάσεων, είναι αληθής. Η συμπεριφορά του εισβολέα, κωδικοποιείται σε εννιά (9) διαφορετικές καταστάσεις, οι οποίες και

αναπαρίστανται από την τοπική μεταβλητή s3. Όπως φαίνεται και από την εικόνα 5, στην κατάσταση 2 ο εισβολέας υποκλέπτει το μήνυμα *I1* και αμέσως το προωθεί στον αρχικό προορισμό του. Στην κατάσταση 5, ο εισβολέας υποκλέπτει το μήνυμα *R1* της οντότητας ανταποκριτή. Το μήνυμα αυτό (που περιέχει το *puzzle*) υπόκειται από τον εισβολέα σε διαφθορά, παράγοντας *N* διαφορετικά διεφθαρμένα *R1s*, τα οποία και τα αποστέλλει στην εναρκτήρια οντότητα. Η δομή και ο σχεδιασμός της οντότητας *I*, αυτό-εξαναγκάζεται να επιλύσει όλα τα (λανθασμένα) *puzzles* των *R1* που λαμβάνει, οδηγώντας τον εαυτό του σε μια εξάντληση των διαθέσιμων πόρων υπολογιστικής ισχύος. Η συχνότητα αποστολής των μηνυμάτων αυτών, διακρίνεται στο μοντέλο με την μεταβλητή a η οποία και επιλέγεται από τον εισβολέα (με βάση το κόστος που θα έχει διαθέσιμος αυτός από την πλευρά του). Έτσι, κάθε διεφθαρμένο R1, θα αποστέλλεται a φορές προς την οντότητα *I* του HIP, έχοντας έναν συνολικό αριθμό διεφθαρμένων μηνυμάτων *M* με λανθασμένα *puzzles*.

Η εικόνα 6.6.6 παρουσιάζει τα κόστη επεξεργασίας για τον εισβολέα που δημιουργήθηκε στο εργαλείο PRISM τα οποία και αντιστοιχούν στην παραγωγή και αποστολή των μηνυμάτων, αλλά και τον ενεργειών των τακτικών επιθέσεων (εικόνα 6.5.1) που περιλαμβάνει ο εισβολέας στο δομοστοιχείο του.

```
rewards "attacker"
......
[finish_m1] true :a*(INIT_+TRNSMT_);
[intr_1] true : a*(TRNSMT_INTR);
[finish_intr_1] true : a*(TRNSMT_INTR);
[finish_m2] true : a*(TRNSMT_+PR_I1_GEN_R1);
[finish_spoof] true : a*(TRNSMT_INTR);
[spoof] true : a*(TRNSMT_INTR);
......
endrewards
```

Εικόνα 6.6.6 Κόστη επεξεργασίας για επιλεγμένες καταστάσεις του εισβολέα *At*

Προς την ίδια κατεύθυνση, ορίζονται κόστη για το υπολογιστικό κόστος της εναρκτήριας οντότητας , όπως φαίνεται και από το σχήμα 6.6.7.

```
rewards "k_value"
......
[m3] k=1 : a*(TRNSMT_+PR_R1_GEN_I2_k5);
[m3] k=10 : a*(TRNSMT_+PR_R1_GEN_I2_k10);
[m3] k=15 : a*(TRNSMT_+PR_R1_GEN_I2_k15);
[m3] k=20 : a*(TRNSMT_+PR_R1_GEN_I2_k20);
[m3] k=25 : a*(TRNSMT_+PR_R1_GEN_I2_k25);
......
endrewards
```

Εικόνα 6.6.7 Κόστη επεξεργασίας για την εναρκτήρια οντότητα *I*, για διαφορετικές τιμές του παράγοντα δυσκολίας *k*

Τα κόστη αυτά έχουν να κάνουν με την δυσκολία επίλυσης του *puzzle* που περιέχεται στο μήνυμα *R1*, κάτι το οποίο επηρεάζεται άμεσα από την επιλογή της τιμής του παράγοντα δυσκολίας *k*. Τα κόστη που ορίστηκαν για αυτή την οντότητα, παρουσιάζονται στην εικόνα 6.6.7.

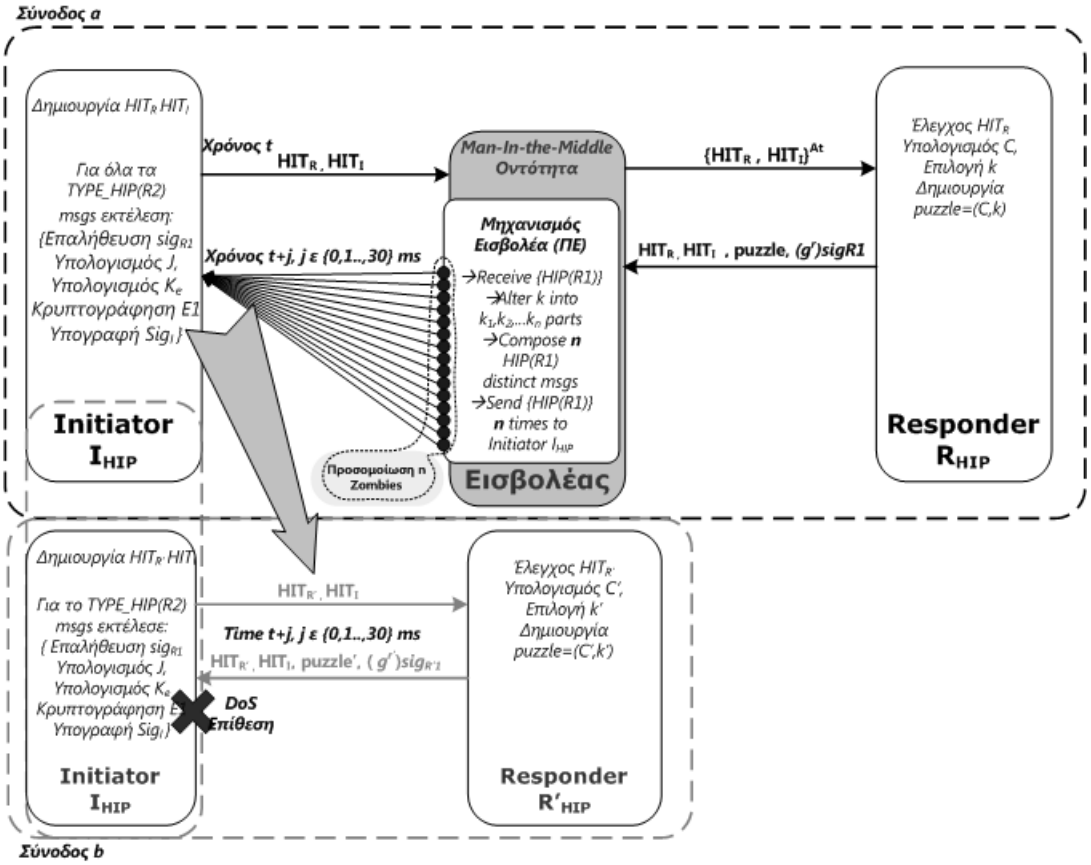
6.6.3 Αποτελέσματα του πιθανοκρατικού ελέγχου μοντέλων για το HIP

Για το μοντέλο DTMC που περιγράφθηκε παραπάνω, θα ορίζουμε το ερώτημα προδιαγραφών στην PCTL, (Q1) για τον έλεγχο συμβεί μια DoS επίθεση ασφαλείας στο πρωτόκολλο HIP ως εξής:

Q1: P=? [true U fail=2],

όπου εξετάζεται η πιθανότητα να ακολουθηθεί ένα μονοπάτι στον παραγόμενο χώρο των καταστάσεων του DTMC μοντέλου, όπου η τελικά υπάρχει κάποια κατάσταση όπου η μεταβλητή `fail` έχει την τιμή 2.

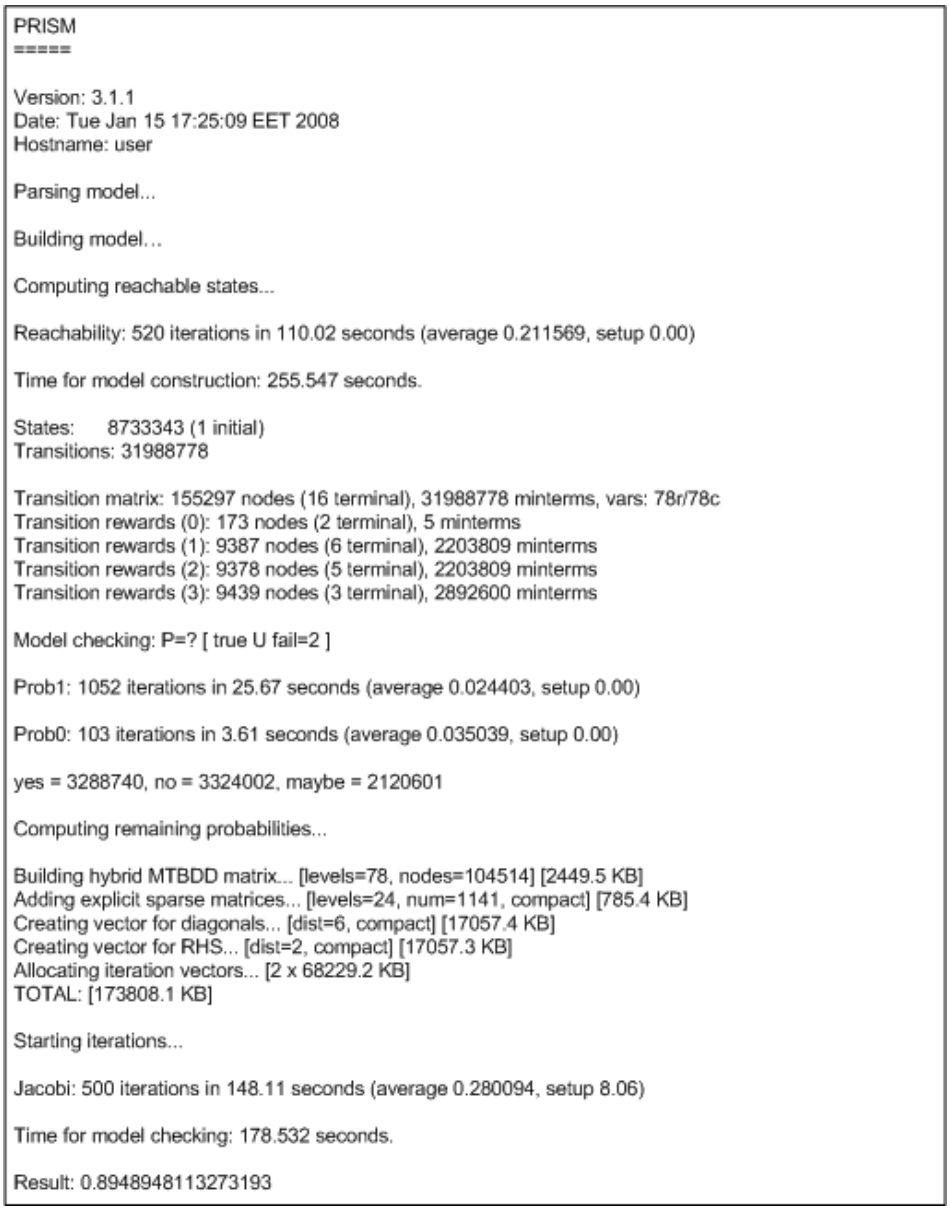
Σε οποιαδήποτε τέτοια κατάσταση, η εναρκτήρια οντότητα *Initiator* επεξεργάζεται ταυτόχρονα *proc_limit* μηνύματα με διεφθαρμένα *puzzles* με την ουρά υποδοχής μηνυμάτων να περιλαμβάνει *B* μηνύματα στο σύνολο. Σαν επακόλουθο του παραπάνω, οποιοδήποτε αυθεντικό μήνυμα τύπου *R1* προερχόμενο από μια έντιμη οντότητα θα απορριφθεί εξαιτίας μη διαθεσιμότητας του *Initiator*, σε μια άλλη σύνοδο του πρωτοκόλλου *b*.



Εικόνα 6.6.8 Σχηματική περιγραφή της DoS επίθεσης στο πρωτόκολλο HIP

Το όλο σχήμα της επίθεσης DoS απεικονίζεται στην εικόνα 6.6.8. Με βάση τον ορισμό 3, εάν ο εισβολέας καταφέρει να καταναλώσει τους διαθέσιμους υπολογιστικούς πόρους με μεγάλη πιθανότητα και με ένα επισυναπτόμενο κόστος μικρότερο σε σχέση με αυτό της εναρκτήριας οντότητας *Initiator*, θα έχουμε αποδείξει την ύπαρξη της απειλής DoS.

Σε αυτό το σημείο αξίζει να σημειωθεί η αποτελεσματικότητα του μοντέλου ΠΕ για την γρήγορη και αποδοτική ανάλυση του χώρου τον παραγόμενων καταστάσεων από το συνολικό DTMC μοντέλο του πρωτοκόλλου HIP. Κάτι τέτοιο, βοηθά σημαντικά τον αναλυτή, ο οποίος με παρόμοιο τρόπο της όλης ποσοτικοποιημένης ανάλυσης ασφαλείας που διεξάγεται, με τον πιθανοκρατικό ελεγκτή μοντέλων PRISM, μπορεί να ελέγξει με πρώτιστος γρήγορο τρόπο, τα πρωτόκολλα που επιθυμεί για πιθανές επιθέσεις τύπου DoS.



```
PRISM
=====

Version: 3.1.1
Date: Tue Jan 15 17:25:09 EET 2008
Hostname: user

Parsing model...

Building model...

Computing reachable states...

Reachability: 520 iterations in 110.02 seconds (average 0.211569, setup 0.00)

Time for model construction: 255.547 seconds.

States:      8733343 (1 initial)
Transitions: 31988778

Transition matrix: 155297 nodes (16 terminal), 31988778 minterms, vars: 78r/78c
Transition rewards (0): 173 nodes (2 terminal), 5 minterms
Transition rewards (1): 9387 nodes (6 terminal), 2203809 minterms
Transition rewards (2): 9378 nodes (5 terminal), 2203809 minterms
Transition rewards (3): 9439 nodes (3 terminal), 2892600 minterms

Model checking: P=? [ true U fail=2 ]

Prob1: 1052 iterations in 25.67 seconds (average 0.024403, setup 0.00)

Prob0: 103 iterations in 3.61 seconds (average 0.035039, setup 0.00)

yes = 3288740, no = 3324002, maybe = 2120601

Computing remaining probabilities...

Building hybrid MTBDD matrix... [levels=78, nodes=104514] [2449.5 KB]
Adding explicit sparse matrices... [levels=24, num=1141, compact] [785.4 KB]
Creating vector for diagonals... [dist=6, compact] [17057.4 KB]
Creating vector for RHS... [dist=2, compact] [17057.3 KB]
Allocating iteration vectors... [2 x 68229.2 KB]
TOTAL: [173808.1 KB]

Starting iterations...

Jacobi: 500 iterations in 148.11 seconds (average 0.280094, setup 8.06)

Time for model checking: 178.532 seconds.

Result: 0.8948948113273193
```

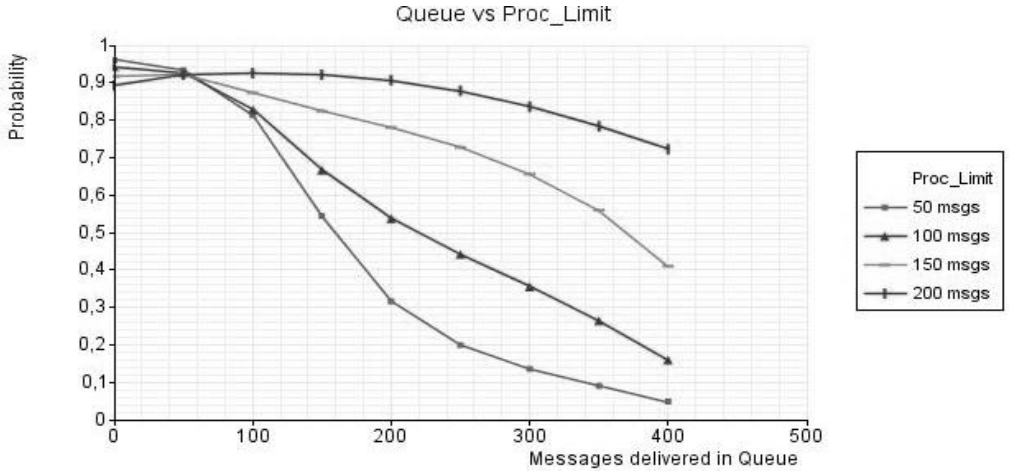
Εικόνα 6.6.9 Πιθανότητα το μοντέλο να προσεγγίσει μια κατάσταση όπου η οντότητα *Initiator* δεν είναι διαθέσιμη

Τα αποτελέσματα επαλήθευσης για το συνολικό πρωτόκολλο του HIP με τον μοντέλο ΠΕ που φαίνονται στην εικόνα 6.6.9, εντοπίζουν μια μεγάλη πιθανότητα (P=0.895) όπου το μοντέλο περιέρχεται σε κατάσταση η οποία αντικατοπτρίζει το ερώτημα *Q1* για ανίχνευση επιθέσεων τύπου DoS. Μια πιο χαρακτηριστική απεικόνιση της παραπάνω επιθέσεως DoS για τα επίπεδα της πιθανότητας σε σχέση με την ουρά επεξεργασίας και αναμονής, όπου η εναρκτήρια οντότητα *Initiator* τίθεται εκτός διαθεσιμότητας, φαίνεται στην εικόνα 6.6.10. Τα συγκεκριμένα αποτελέσματα λαμβάνονται με τον έλεγχο

ιδιοτήτων για την πιθανότητα, από το ερώτημα *Q2,* όπως αυτό ορίζεται στην γλώσσα προδιαγραφών PCTL:

Q2: P=? [true U msgs_in_service=proc_limit & msgs_in_queue=B],

όπου η μεταβλητή *proc_limit* ποικίλει από το 0 μέχρι το 200 με βήμα 50, και η *B* ποικίλει από το 0 μέχρι το 400, με βήμα 50.

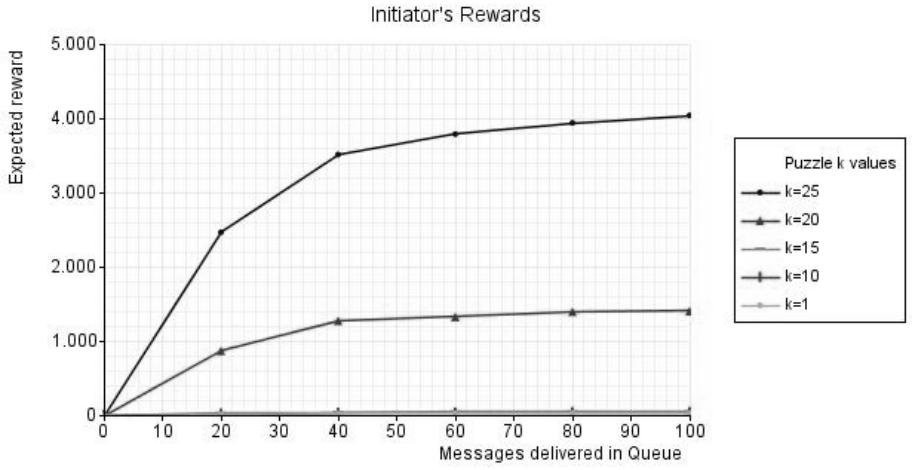


Εικόνα 6.6.10 Πιθανότητα για την προσέγγιση κατάστασης όπου η εναρκτήρια οντότητα Initiator δεν είναι διαθέσιμος για διαφορετικές τιμές των ουρών του HIP

Σημειώνεται ότι η προστασία που παρέχει το πρωτόκολλο HIP (DoS Resistance) βελτιώνεται για πλήθος μηνυμάτων μεγαλύτερο από 250, κάτω από την προϋπόθεση ότι η εναρκτήρια οντότητα δεν θα επεξεργάζεται παράλληλα πάνω από 50 μηνύματα (αιτήσεις). Παρόλα αυτά, μια ολοκληρωμένη εικόνα του μηχανισμού αντοχής του HIP στις επιθέσεις DoS, δίνεται από τα αποτελέσματα που παρέχονται από τον πιθανοκρατικό ελεγκτή μοντέλων PRISM και τα ερωτήματα ανταμοιβών (reward queries) που ορίζονται στην PCTL.Τα ερωτήματα αυτά θα βασίζονται στα δομοστοιχεία ανταμοιβών που ορίστηκαν παραπάνω για το υπολογιστικό κόστος των έντιμων οντοτήτων και του κόστους του εισβολέα *At*. Το ερώτημα *Q3* παρέχει αποτελέσματα για το κόστος της εναρκτήριας οντότητας *Initiator* και του εισβολέα *At*, για διαφορετικές τιμές της δυσκολίας επίλυσης του *puzzle k* μέχρι την προσέγγιση μιας κατάστασης όπου η εναρκτήρια οντότητα θα καταναλώσει τους διαθέσιμους πόρους της και θα τεθεί εκτός λειτουργίας. Σημειώνεται ότι στο μοντέλο του DTMC η τιμή δυσκολίας του *puzzle k* επιλέγεται αρχικά από την οντότητα ανταποκριτή *Responder* με πιθανολογικό τρόπο. Θα έχουμε:

Q3: R[Initiator_cost]=? [true U msgs_in_queue=B & k=puz_dif],

Η εικόνα 6.6.11 παρουσιάζει τα αποτελέσματα για το κόστος της εναρκτήριας οντότητας, όταν η ουρά *B* ποικίλει από τιμές 0 μέχρι 100 με βήμα 20, και οι τιμές δυσκολίας επίλυσης του *puzzle k* (μεταβλητή *puz_dif*) επιλέγονται 1, 10, 15, 20, 25. Για τα επιτρεπόμενα μήκη της ουράς από 0 μέχρι 60 το κόστος της εναρκτήριας οντότητας αυξάνεται δραματικά ειδικά για τις τιμές του *k* μεγαλύτερες από την τιμή 20.



Εικόνα 6.6.11 Συνολικό υπολογιστικό κόστος για την εναρκτήρια οντότητα *Initiator* όταν αυτή βρίσκεται σε κατάσταση μη διαθεσιμότητας, με βάση τα μηνύματα που βρίσκονται στην ουρά

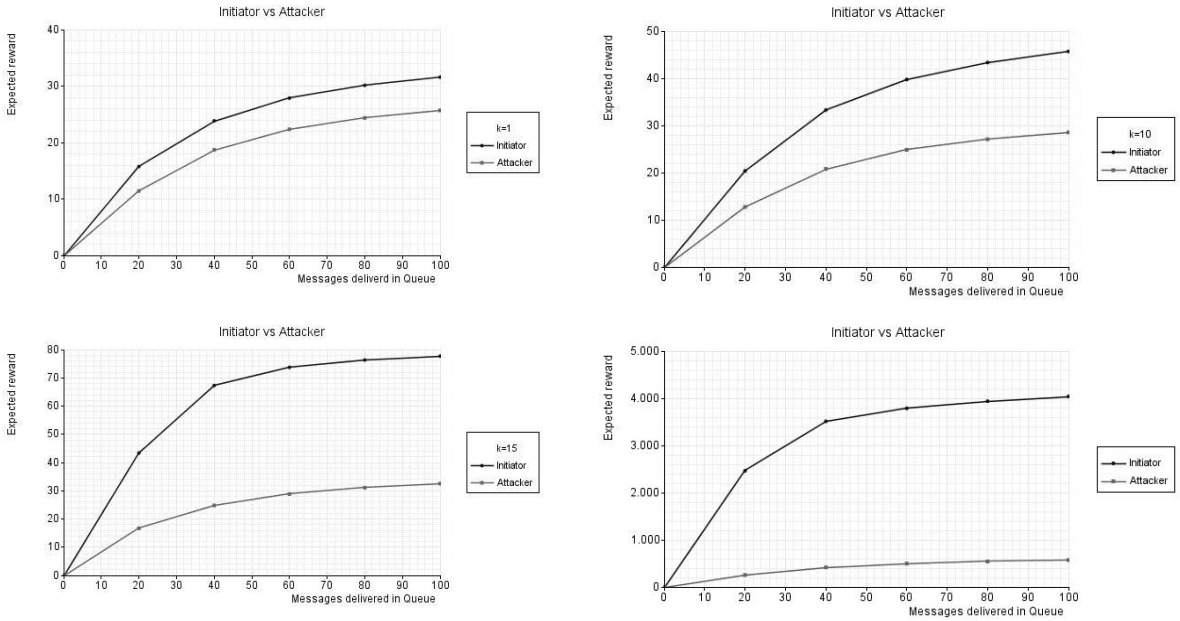
Τα ερωτήματα ανταμοιβών *Q4* και *Q5* έρχονται με την σειρά τους να αποτυπώσουν την αντοχή του συστήματος σε επιθέσεις DoS, με βάση και τον ορισμό 3 που δόθηκε σε προηγούμενη παράγραφο. Οι γράφοι που παρουσιάζονται στην εικόνα 6.6.12 συγκρίνουν το συνολικό κόστος για τον εισβολέα *At* σε σχέση με το κόστος της εναρκτήριας οντότητας *Initiator*. Έτσι ορίζεται:

Q4: R[Initiator_cost]=? [true U msgs_in_queue=B & k=puz_dif]

Q5: R[Attacker_cost]=? [true U msgs_in_queue=B & k=puz_dif]

Όπως και στα προηγούμενα ερωτήματα, με τον ίδιο τρόπο η ουρά *B* θα ποικίλει από τιμές 0 μέχρι 100 με βήμα 20, για διαφορετικές τιμές δυσκολίας επίλυσης του puzzle k (και σε αυτό το παράδειγμα οι τιμές επιλέγονται 1, 10, 15, 20). Τα απεικονιζόμενα αποτελέσματα δείχνουν την επίδραση της επίθεσης DoS που μπορεί να δεχτεί το πρωτόκολλο HIP σε σχέση με το κόστος της έντιμης οντότητας *Initiator* και του ενός εισβολέα *At*. Σε πιο ρεαλιστικό επίπεδο, η τιμή

της επίλυσης του puzzle k αναφέρεται στους διαθέσιμους πόρους που πρέπει να έχουν οι οντότητες για εκπληρώσουν επιτυχώς μια σύνοδο του HIP. Οι συγκεκριμένες τιμές λ.χ. μπορούν να αντιπροσωπεύσουν πόρους μηχανών που υλοποιείται το πρωτόκολλο HIP όπως για παράδειγμα ηλεκτρονικές συσκευές χειρός (palmtops), κινητά ή απλά τερματικά. Έτσι, για διαφορετικές τιμές του *puzzle k* υπολογίζονται τα συνολικά κόστη με βάση των αριθμό των μηνυμάτων που επεξεργάζεται η εναρκτήρια οντότητα *Initiator*.



Εικόνα 6.6.12 Συνολικό υπολογιστικό κόστος για την οντότητα *Initiator* και τον πιθανοκρατικό εισβολέα *At* για τιμές $k = 1, 10, 15, 20$

Το δομοστοιχείο του εισβολέα *At* που ορίστηκε στο μοντέλο DTMC διαφθείρει τα μερικώς υπογεγραμμένα μηνύματα που υποκλέπτει από την οντότητα *Responder*, εφαρμόζοντας παράλληλα μια επίθεσης επανάληψης μηνύματος με αποστολέα την οντότητα Initiator με σκοπό την εισαγωγή του σε διαφορετικές εσωτερικές διεργασίες επίλυσης των διεφθαρμένων puzzles, καταναλώνοντας άσκοπα τους διαθέσιμους υπολογιστικούς πόρους του. Παρατηρώντας την επίδραση της τιμής επίλυσης του *puzzle k*, συμπεραίνεται ότι σε περιπτώσεις όπου αυτό επιλεγεί ανάμεσα στις τιμές 10 και 18, μαζί με κατάλληλες ρυθμίσεις αποδοχής μηνυμάτων στην ουρά της οντότητας *Initiator*, μπορεί η συγκεκριμένη επίθεση να μην είναι εφικτή, όσον αφορά το σχετικό κόστος εισβολέα-οντότητας.

Μια εναλλακτική λύση για την αποφυγή της παραπάνω επίθεσης, με το υπολογιστικό κόστος της να κυμαίνεται σε λογικά πλαίσια, είναι η χρήση ενός μοναδικού μετρητή μέσα στο υπογεγραμμένο μέρος στο μήνυμα *R1* του HIP ανά σύνοδο που εκτελείται. Ένας τέτοιος μοναδιαίος αυξανόμενος μετρητής, θα αντιστοιχεί στην εκάστοτε παραγωγή των puzzles από την οντότητα *Responder* με την οντότητα *Initiator* να επιλύει puzzles Που προέρχονται από τον ίδιο μετρητή. Και σε αυτή την περίπτωση, η ποσοτικοποιημένη πιθανοκρατική ανάλυση με το μοντέλο ΠΕ μπορεί με εύκολο τρόπο να αποδείξει την αποτελεσματικότητα ή όχι των μέτρων ασφαλείας που επρόκειτο να ληφθούν. Σε μια τέτοια περίπτωση, το υλοποιημένο πιθανοκρατικό μοντέλο θα βασιστεί σε *Μαρκοβιανές Διεργασίες Αποφάσεων (Markov Decision Processes, MDP).*

Πίνακας 6.6.3: Στατιστικά απόδοσης ερωτημάτων PCTL

Παραγωγή μοντέλου DTMC		
Χρόνος CPU (sec)	**Αριθμός Καταστάσεων**	**Αριθμός Μεταβάσεων**
255.547	8733343	31988778
PCTL ερώτημα Q1		
Χρόνος CPU (sec)	**Δεσμευμένη Μνήμη (KB)**	**Επαναλήψεις Jaccobi**
178.532	173808.1	500 (in 148.11 sec)
PCTL ερώτημα Q2 (proc_limit=200 msgs, B=400)		
Χρόνος CPU (sec)	**Δεσμευμένη Μνήμη (KB)**	**Επαναλήψεις Jaccobi**
204.953	178593.2	488 (in 187.86 sec)
PCTL ερώτημα Q3 (k=10)		
Χρόνος CPU (sec)	**Μηνύματα στην ουρά**	**Επαναλήψεις Jaccobi**
7.547	20	113 (in 6.52 sec)
16.109	40	213 (in 15.17 sec)
27.625	60	313 (in 25.78 sec)
36.359	80	413 (in 35.20 sec)
45.547	100	513 (in 44.64 sec)
PCTL ερώτημα Q4, Q5 (k=20)		
Χρόνος CPU (sec)	**Μηνύματα στην ουρά**	**Επαναλήψεις Jaccobi**
7.469	20	173 (in 6.47 sec)
16.218	40	273 (in 15.19 sec)
27.516	60	373 (in 25.72 sec)
36.016	80	473 (in 34.78 sec)
45.875	100	573 (in 45.78 sec)

Παρόλα αυτά όμως το ρίσκο που λαμβάνεται σε αυτή την περίπτωση για την ανάλυση εκ νέου απειλών DoS, με το παρόν μοντέλο έγκειται στην δημιουργία τεράστιου χώρου των παραγόμενων καταστάσεων, ή μεγάλη χρονική διάρκεια για τον εξαντλητικό πιθανοκρατικό έλεγχο ενός ερωτήματος σε PCTL. Στον πίνακα 6.6.3 παρέχονται αντιπροσωπευτικά στατιστικά για το σύνολο των ερωτημάτων PCTL που μελετήθηκαν σε αυτή την παράγραφο. Όλα

τα αποτελέσματα διεξήχθησαν σε ένα τερματικό με επεξεργαστή Pentium IV (3.6 GHz) και διαθέσιμη μνήμη στα 2 GB.

6.7 Συμπεράσματα Κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκε μια προσέγγιση ποσοτικοποιημένης επαλήθευσης ως μέσο για την μελέτη επιθέσεων DoS από ένα μοντέλο πιθανοκρατικού εισβολέα ΠΕ, στο περιβάλλον του πιθανοκρατικού ελεγκτή μοντέλων PRISM. Περιγράφθηκε αρχικά οι ιδιότητες ασφαλείας που έχουν να κάνουν με την αντοχή των πρωτοκόλλων ασφαλείας σε επιθέσεις που έχουν σκοπό την άρνηση εξυπηρέτησης των συμμετεχόντων οντοτήτων στο πρωτόκολλο. Ορίζεται η ιδιότητα αντοχής σε DoS ως ποσοτικό μέγεθος, η οποία εξαρτάται α) από την πιθανότητα προσέγγισης μιας κατάστασης όπου μια έντιμη οντότητα του πρωτοκόλλου είναι μη διαθέσιμη και β) των υπολογιστικών απαιτήσεων ενός εισβολέα προς δημιουργίας της παραπάνω κατάστασης με βάση την δυσαναλογία του κόστους του εισβολέα, να προκαλέσει την επίθεση και το κόστος της οντότητας που προκαλείται. Στη συνέχεια παρουσιάστηκε και ορίστηκε φορμαλιστικά ο πιθανοκρατικός εισβολέας ΠΕ, ο οποίος βασίζεται σε προσεκτικά δομημένες ενέργειες αποστολής-λήψης μηνυμάτων, αλλά και της ικανότητάς του να παραβιάσει την ακεραιότητά τους. Το όλο μοντέλο του εισβολέα, που συμβολίζεται με *At*, δημιουργήθηκε στο περιβάλλον του εργαλείου PRISM βασιζόμενος σε Μαρκοβιανές αλυσίδες διακριτού χρόνου, DTMC. Το γενικό πλαίσιο εφαρμογής που υλοποιήθηκε, δίνει την δυνατότητα μοντελοποίησης οποιασδήποτε μορφής συστήματος κατανάλωσης πόρων (εύρος επικοινωνίας, μνήμη ή υπολογιστικό κόστος) δίνοντας στον αναλυτή την ικανότητα να δημιουργήσει δομοστοιχεία ανταμοιβών, αντιπροσωπευτικά των οντοτήτων που συμμετέχουν στην σύνοδο του πρωτοκόλλου που εξετάζεται.

Για την εφαρμογή της όλης ανάλυσης, επιλέχθηκε η ανάλυση ασφαλείας για επιθέσεις DoS του πρωτοκόλλου Host Identity Protocol (HIP). Το HIP αποτελεί ένα ειδικά σχεδιασμένο σύστημα για την προστασία των συμμετεχόντων του σε επιθέσεις DoS, το οποίο βασίζεται σε κρυπτογραφικούς μηχανισμούς φύσης επίλυσης δύσκολων προβλημάτων, puzzles. Μετά την

υλοποίηση του πρωτοκόλλου, των οντοτήτων του αλλά και του μοντέλου ΠΕ, προσαρτώνται στο πρωτόκολλο δομοστοιχείες ανταμοιβών, για όλες τις σχετιζόμενες με υπολογιστικό κόστος, ενέργειες των διεργασιών, τόσο του εισβολέα όσο και των έντιμων οντοτήτων. Τα αποτελέσματα της ανάλυσης, εντόπισαν μια σοβαρή επίθεσης DoS για το πρωτόκολλο HIP, δίνοντας πληροφορίες κάτω από τις οποίες μπορεί να περιέλθει το πρωτόκολλο στην κατάσταση αυτή.

Η συγκεκριμένη εργασία, αποτελεί ουσιαστικά την πρώτη απόπειρα ελέγχου επιθέσεων DoS με πιθανοκρατικό τρόπο, για πρωτόκολλα ασφαλείας. Πιστεύεται ότι η συγκεκριμένη προσέγγιση του εισβολέα ΠΕ, αποτελεί ένα αξιόλογο μέσο για τον αναλυτή που θέλει να επαληθεύσει την απουσία επιθέσεων DoS, για το πρωτόκολλο που σχεδιάζει. Ακολουθώντας τα διαδικαστικά βήματα δόμησης του εισβολέα ΠΕ, και τις υποθέσεις που πάρθηκαν για την τοποθέτησή του στο όλο επικοινωνιακό σύστημα, ο αναλυτής μπορεί να μοντελοποιήσει το πρωτόκολλο της αρεσκείας του ως ένα σύστημα κατάστασης-μεταβάσεων, βασισμένο στις αρχές των DTMCs. Η συνολική εργασία δεν είναι ιδιαίτερα δύσκολη, ειδικά για τους αναλυτές-ειδικούς της ασφάλειας πρωτοκόλλων.

Κεφάλαιο 7ο

Επίλογος – Μελλοντικές Προοπτικές

7.1 Γενικά

Στην σημερινή εποχή, ένα από τα θέματα τα οποία απασχολούν και θα συνεχίζουν να απασχολούν τους επιστήμονες της πληροφορικής είναι και αυτό της ποιότητας, αξιοπιστίας και της ασφάλειας του λογισμικού. Τα θέματα αυτά αποτελούν καίρια αν μη τι άλλο ζητήματα στα οποία οι αναλυτές-σχεδιαστές συστημάτων επιθυμούν πάντα να εκπληρώνουν τα τελικά τους προϊόντα. Τα πρωτόκολλα ασφαλείας σήμερα αποτελούν κατανεμημένα προϊόντα λογισμικού, τα οποία υπερκαλύπτουν την πλειοψηφία των επικοινωνιών στο διαδίκτυο. Εξαιτίας της διάδοσής τους όμως, πολλοί εξωτερικοί παράγοντες, όπως κακόβουλοι χρήστες, μπορούν με κατάλληλες ενέργειες να αποτρέψουν την ολοκλήρωση της επικοινωνίας των πρωτοκόλλων, προσπαθώντας είτε να βλάψουν τις έντιμες οντότητες-χρήστες που συμμετέχουν σε αυτή, είτε να αποκομίσουν διάφορα οικονομικά οφέλη.

Τη λύση στο συγκεκριμένο πρόβλημα έρχεται να δώσει ο συστηματικός και εξαντλητικός έλεγχος των πρωτοκόλλων προσπαθώντας να υπερκαλύψει όλα τα σενάρια εκείνα που μπορούν να διαφθείρουν τις υπηρεσίες ασφαλείας του πρωτοκόλλου. Σε αυτό το σημείο επεμβαίνουν οι τυπικές μέθοδοι ανάλυσης συστημάτων και πιο συγκεκριμένα ο αυτόματος έλεγχος μοντέλων. Ο αναλυτής μπορεί πριν τον σχεδιασμό και την υλοποίηση του πρωτοκόλλου ασφαλείας, να

δημιουργήσει ένα μοντέλο το οποίο θα αντικατοπτρίζει τις απαραίτητες λειτουργίες του πρωτοκόλλου. Στη συνέχεια θα πρέπει να ορίσει τις βασικές ιδιότητες και εγγυήσεις που επιθυμεί να προσφέρει το πρωτόκολλο αυτό, κατά την διάρκεια της λειτουργίας του, προς στις οντότητες του πρωτοκόλλου. Έπειτα αφήνεται στον αυτόματο ελεγκτή μοντέλων να παράγει όλες τις πιθανές καταστάσεις του μοντέλου του πρωτοκόλλου ασφαλείας στο οποίες μπορεί να περιέλθει αυτό, ελέγχοντάς τες εξαντλητικά για το εάν κάποια από αυτές παραβιάζει τις ιδιότητες εκείνες που έχει ορίσει αρχικά ο αναλυτής. Η επονομαζόμενη διαδικασία αυτής της τυπικής επαλήθευσης του χώρου των καταστάσεων ενός μοντέλου, θα επιστρέψει είτε μια θετική απάντηση, ότι δηλαδή το μοντέλο πληρεί τις προϋποθέσεις του, είτε μια αρνητική απάντηση εντοπίζοντας μια παραβίαση ιδιότητας (και επομένως ένα λάθος στο πρωτόκολλο) μαζί με ένα παράδειγμα-μονοπάτι που οδηγεί στο λάθος αυτό. Το παράδειγμα αυτό, αποκαλούμενο και ίχνος λάθους, μπορεί να βοηθήσει τον αναλυτή να κατανοήσει το λάθος έτσι ώστε να προβεί στις κατάλληλες διορθώσεις του μοντέλου του.

Εξαιτίας των προβλημάτων ασφαλείας που παρουσιάζονται σήμερα στο διαδίκτυο, από διάφορους εξωγενείς παράγοντες, ο έλεγχος των πρωτοκόλλων ασφαλείας έχει ταυτιστεί με την δημιουργία ενός κατάλληλου μοντέλου εισβολέα, το οποίο σκοπεύει στην εκμετάλλευση των υπηρεσιών ασφαλείας του πρωτοκόλλου με κάθε τρόπο. Από τα σημαντικότερα μοντέλα εισβολέων που παρουσιάζονται στη βιβλιογραφία είναι αυτό των Dolev και Yao. Στο συγκεκριμένο μοντέλο, ο εισβολέας θεωρείται ως ο κυρίαρχος του επικοινωνιακού μέσου που χρησιμοποιείται για την πραγματοποίηση της επικοινωνίας μεταξύ των οντοτήτων, κατέχοντας παράλληλα μια σειρά από ενέργειες, με βάση των οποίων προσπαθεί να αυξήσει την γνώση του για τεχνικές λεπτομέρειες, είτε του πρωτοκόλλου είτε των οντοτήτων που συμμετέχουν σε αυτό.

Σε προσπάθειες που έχουν γίνει για την αποτύπωση ενός τέτοιου μοντέλου εισβολέα με την βοήθεια της τεχνικής του αυτόματου ελέγχου μοντέλων, οι αναλυτές έρχονται αντιμέτωποι με το γνωστό φαινόμενο της έκρηξης του χώρου των καταστάσεων, ΕΧΚ. Πρόκειται για τις περιπτώσεις εκείνες, όπου το μοντέλο του εισβολέα εμπεριέχει ενέργειες οι οποίες εκτινάσσουν τις

καταστάσεις του πρωτοκόλλου σε περιπτώσεις αλληλεπίδρασής του, θέτοντας ανέφικτη τον εξαντλητικό έλεγχο του μοντέλου του πρωτοκόλλου. Επιπρόσθετα, στην παρούσα βιβλιογραφία παρατηρείται μια έλλειψη ελέγχου για επιθέσεις που έχουν καταγραφεί σήμερα; επιθέσεις οι οποίες συγκαταλέγονται στις δυνατότητες των μοντέλων εισβολέων που χρησιμοποιούνται, για τον ολοκληρωτικό έλεγχο ασφαλείας των πρωτοκόλλων αυτών. Συνοψίζοντας τα προβλήματα στα οποία καλείται να συνεισφέρει η παρούσα διατριβή, έχουμε:

- Φορμαλιστικός ορισμός, σχεδιασμός και υλοποίηση κατάλληλων μοντέλων εισβολέων για αποτελεσματικό έλεγχο πρωτοκόλλων ασφαλείας μέσω τυπικών μεθόδων ανάλυσης συστημάτων
- Αποτύπωση σημερινών (ή νέων) επιθέσεων ασφαλείας στα μοντέλα των εισβολέων για έλεγχο αυτών σε πρωτόκολλα ασφαλείας
- Αποφυγή του φαινομένου της έκρηξης του χώρου των καταστάσεων ΕΧΚ, κατά την διάρκεια αλληλεπίδρασης των εισβολέων με τις έντιμες οντότητες του πρωτοκόλλου
- Συνδυασμός των μοντέλων των εισβολέων με τα εργαλεία αυτόματου ελέγχου μοντέλων, για επικάλυψη και εντοπισμό περισσότερων περιπτώσεων παραβιάσεων ασφαλείας πέρα αυτών που εξαπολύουν οι εισβολείς απέναντι στα πρωτόκολλα

7.2 Συνεισφορά της διατριβής

Η συνολική συνεισφορά της παρούσας διατριβής έγκειται στην δημιουργία εξειδικευμένων θεωριών υλοποίησης μοντέλων εισβολέων μέσω του ελέγχου μοντέλων, για τον αποτελεσματικό έλεγχο ασφαλείας πρωτοκόλλων αποφεύγοντας το φαινόμενο της ΕΧΚ. Ειδικότερα, δημιουργήθηκε το πρώτο μοντέλο εισβολέα που περιέχει στη δομή του ένα σύνολο από τις πιο γνωστές επιθέσεις που συναντιόνται σήμερα στα διαδικτυακά πρωτόκολλα ασφαλείας. Το επονομαζόμενο μοντέλο Εισβολέα Πολλαπλών Επιθέσεων (ΕΠΕ), που παρουσιάστηκε στο κεφάλαιο 4, βασίζεται στον επακριβή ορισμό των διακριτών ακολουθιακών ενεργειών αποστολής επεξεργασίας και λήψης μηνυμάτων, οι οποίες συνολικά αποτιμώνται με την διενέργεια επιθέσεων

έναντι των οντοτήτων του πρωτοκόλλου ασφαλείας. Υλοποιώντας βασικές τακτικές επιθέσεων, ο εισβολέας έχει την δυνατότητα να τις συνδυάσει με σκοπό την δόμηση πολύπλοκων επιθέσεων ασφαλείας, όπως για παράδειγμα μια επίθεση άρνησης της εξυπηρέτησης (DoS), χωρίς όμως να εκτινάσσει τις καταστάσεις κατά την διάρκεια επαλήθευσης των εγγυήσεων ασφαλείας στον παραγόμενο χώρο των καταστάσεων από τον ελεγκτή μοντέλων. Ως αποτέλεσμα εφαρμογής του μοντέλου ΕΠΕ, παρατίθενται η τυπική ανάλυση δύο πρωτοκόλλων ασφαλείας μικροπληρωμών, το PayWord και το MicroMint. Αποτέλεσμα της επιτυχούς εφαρμογής του μοντέλου ΕΠΕ, ήταν από την μία ο έλεγχος μια σειράς ενεργειών επιθέσεων στα πρωτόκολλα αυτά αποφεύγοντας παράλληλα το φαινόμενο ΕΧΚ. Επιπρόσθετα ανακαλύφθηκε και επιβεβαιώθηκε μέσω επαλήθευσης ένα σοβαρό λάθος παραβίασης της ασφάλειας που υπάρχει στην παρούσα έκδοση του πρωτοκόλλου PayWord.

Δημιουργείται η θεωρία του μοντέλου εισβολέα διερεύνησης μηνύματος (ΕΔΜ), όπως περιγράφθηκε στο 5ο κεφάλαιο. Ο εισβολέας αυτός βασίζεται σε υποθέσεις μοντελοποίησης του γνωστού εισβολέα DY , παρέχοντας επιπλέον πληροφορίες για το πρωτόκολλο που αλληλεπιδρά, διερευνώντας τα μηνύματα που ανταλλάσσουν οι οντότητες του πρωτοκόλλου. Κύριο χαρακτηριστικό του η επίδρασή του στον παραγόμενο χώρο των καταστάσεων, βελτιώνοντας την γνώση του εισβολέα, με σκοπό την επιλογή των αποτελεσματικών επιθέσεων που μπορούν να οδηγήσουν σε εντοπισμό παραβιάσεων σε πρωτόκολλα ασφαλείας. Ο συγκεκριμένος εισβολέας αποδεικνύει την αποτελεσματικότητά του, οδηγώντας επιτυχώς στην ανακάλυψη της γνωστής επίθεσης πλαστοπροσωπίας στο πρωτόκολλο ασφαλείας ασύμμετρης κρυπτογράφησης των Needham και Schroeder, εμφανίζοντας μεγάλο πλεονέκτημα σε σύγκριση με τον εισβολέα DY, στις παραγόμενες καταστάσεις.

Τέλος, στο κεφάλαιο 6, παρουσιάζεται η θεωρία του πιθανοκρατικού εισβολέα (ΠΕ). Ο συγκεκριμένος εισβολέας ορίζεται και υλοποιείται μέσα στο εργαλείο πιθανοκρατικού ελέγχου μοντέλων PRISM. Λόγω της φύσης της συγκεκριμένης τεχνικής, η οποία βασίζεται σε Μαρκοβιανές αλυσίδες διακριτού χρόνου (DTMC), ο εισβολέας αυτός ακολουθεί τις συγκεκριμένες αρχές, επικεντρώνοντας στην αποκάλυψη λάθος ασφαλείας που χαρακτηρίζονται ως επιθέσεις άρνησης της εξυπηρέτησης (Denial of Service), καθιστώντας τον

πρώτο εισβολέα που δημιουργήθηκε με τον τρόπο αυτό. Ως αποτέλεσμα του εισβολέα ΠΕ, είναι η ανακάλυψη μιας άγνωστης επίθεσης DoS στο πρωτόκολλο ασφαλείας ασύρματου ή ενσύρματου περιβάλλοντος HIP.

7.3 Επίλογος και μελλοντικές προοπτικές

Με σκοπό την αποτύπωση των μελλοντικών προοπτικών της όλης ερευνητικής προσπάθειας, θα ήταν σωστό να αναφερθούν ξεχωριστά οι προοπτικές ερευνητικών εργασιών προς επέκταση και των τριών μοντέλων εισβολέων που παρουσιάστηκαν στην διατριβή αυτή.

Το μοντέλο του εισβολέα πολλαπλών επιθέσεων (ΕΠΕ) επιτρέπει μέσα στη δομή του την προσθήκη επιπλέον επιθέσεων που επιθυμεί ο ίδιος ο χρήστης να δοκιμάσει πάνω στο πρωτόκολλο ασφαλείας που ελέγχει, πέρα από αυτές που ήδη διαθέτει στην διεργασία του. Επιπλέον, το μοντέλο ΕΠΕ δεν θέτει περιορισμούς στον εκάστοτε ελεγκτή μοντέλων που χρησιμοποιείται, μιας και δεν αποτρέπει τον έλεγχο για γενικές παραβιάσεις ιδιοτήτων ασφάλειας, βιωσιμότητας ή δικαιοσύνης του πρωτοκόλλου. Αντίθετα, δίνεται ώθηση με την συμμετοχή μιας δυνατής κακόβουλης οντότητας να προκαλέσει (εάν μπορεί) καταστάσεις επίθεσης στο πρωτόκολλο, που θεωρούνται ως καταστάσεις παραβίασης της ασφάλειας. Τέλος, αξίζει να σημειωθεί ότι η δομή του εισβολέα ΕΠΕ που είναι ορισμένη στην γλώσσα PROMELA, είναι τέτοια που επιτρέπει την επέκταση της βάσης των επιθέσεών του, με περισσότερες τακτικές ή και πιο πολύπλοκες επιθέσεις, με στόχο έλεγχο ιδιοτήτων ασφαλείας όπως μη-αποποίηση της ευθύνης (nor-repudiation) ή ανωνυμία (anonymity) των συμμετεχόντων οντοτήτων.

Στο ίδιο μήκος, επιπλέον επιθέσεις μπορούν να ορισθούν και στο μοντέλο του εισβολέα διερεύνησης μηνύματος, ΕΔΜ. Σε αυτή την περίπτωση όμως, ο αναλυτής θα πρέπει να τροποποιήσει τους ήδη υπάρχοντες κανόνες διερεύνησης των μηνυμάτων, τα οποία υποκλέπτει ο εισβολέας, με σκοπό την αποτελεσματική χειραγώγηση του παραγόμενου χώρου των καταστάσεων.

Άλλες προοπτικές εξέλιξης του μοντέλου ΕΔΜ, θα μπορούσε να είναι ο έλεγχός που θα διεξήγαγε για ιδιότητες ασφαλείας [88] που αφορούν γενικές ιδιότητες βιωσιμότητας του πρωτοκόλλου [23], όπως ορθός τερματισμός της

συνόδου του πρωτοκόλλου, ή επικαιρότητα μηνύματος. Σε αυτή την περίπτωση το μοντέλο ΕΔΜ θα πρέπει να παρουσιάζει ενέργειες εκτελέσεων που αποσκοπούν στην εκπλήρωση υποθέσεων δικαιοσύνης που πρέπει να παρέχουν οι συμμετέχουσες οντότητες στο πρωτόκολλο, κάτι που δεν καλύπτεται σήμερα από τους υπάρχοντες εισβολείς στην βιβλιογραφία. Τέλος θα μπορούσε η όλη ανάλυση να υλοποιηθεί και να δομήσει έναν ξεχωριστό εργαλείο ελέγχου μοντέλων, όπου ανεξάρτητα από τους περιορισμούς που τίθενται από τα υπό εξέταση πρωτόκολλα ασφαλείας, θα μπορούσε να συμπεριλάβει τον εισβολέα ΕΔΜ αυτόματα στις προδιαγραφές που ορίζει ο αναλυτής για το μοντέλο του.

Η εισβολέας του πιθανοκρατικού ελέγχου (ΠΕ), αποτελεί ουσιαστικά την πρώτη απόπειρα ελέγχου επιθέσεων DoS με πιθανοκρατικό τρόπο, για πρωτόκολλα ασφαλείας. Πιστεύεται ότι η συγκεκριμένη προσέγγιση του εισβολέα ΠΕ, αποτελεί ένα αξιόλογο μέσο για τον αναλυτή που θέλει να επαληθεύσει την απουσία επιθέσεων DoS, για το πρωτόκολλο που σχεδιάζει. Ακολουθώντας τα διαδικαστικά βήματα δόμησης του εισβολέα ΠΕ, και τις υποθέσεις που πάρθηκαν για την τοποθέτησή του στο όλο επικοινωνιακό σύστημα, ο αναλυτής μπορεί να μοντελοποιήσει το πρωτόκολλο της αρεσκείας του ως ένα σύστημα κατάστασης-μεταβάσεων, βασισμένο στις αρχές των DTMCs. Η συνολική διεργασία δεν είναι ιδιαίτερα δύσκολη, ειδικά για τους αναλυτές-ειδικούς της ασφάλειας πρωτοκόλλων. Ως μελλοντική ερευνητική επέκταση της προτεινόμενης μεθοδολογίας, αποτελεί η επικάλυψη και άλλων επιθέσεων για το μοντέλο του εισβολέα στο πιθανοκρατικό περιβάλλον, προσπαθώντας με αυτό τον τρόπο τον υπολογισμό πιθανοτήτων γενικών παραβιάσεων ασφαλείας που μπορούν να προκύψουν. Σαν πρώτη προτεραιότητα τίθεται και η επέκταση της όλης δομής του μοντέλου του εισβολέα προσπαθώντας να αποτυπώσει τεχνικές επίθεσης που συγκαταλέγονται ως κατανεμημένες επιθέσεις άρνησης εξυπηρέτησης DDoS (Distributed Denial of Service).

Όπως αναφέρθηκε και στην εισαγωγή του κεφαλαίου αυτού ο έλεγχος του λογισμικού αποτελεί βασικό κομμάτι κάθε εταιρείας η οποία θα θέλει να παράγει αξιόπιστα ασφαλή προϊόντα. Κάτι τέτοιο έχει μεγαλύτερη ακόμα σημασία για τους αναλυτές και σχεδιαστές πρωτοκόλλων ασφαλείας, τα οποία αποτελούν λογισμικά συστήματα που διαχειρίζονται ευαίσθητες πληροφορίες

και προσωπικά δεδομένα της πλειοψηφίας των χρηστών του διαδικτύου. Με την βοήθεια του αυτόματου ελέγχου μοντέλων, μπορούν να αποφευχθούν παραβιάσεις ασφαλείας αλλά και γενικά σφάλματα του λογισμικού που μπορεί να αποβούν μοιραίες, ειδικά σε περιπτώσεις συστημάτων κρίσιμα σε ασφάλεια. Κάνοντας αναφορά στις ενότητες με τις οποίες ασχοληθήκαμε, για την πραγματοποίηση του έλεγχου μοντέλων ο κάθε αναλυτής θα πρέπει να έχει γνώση των βασικών εννοιών που αφορούν την περιοχή αυτή, έχοντας επίσης εξοικείωση με μαθηματικούς όρους που αφορούν την μοντελοποίηση και γενικά την αναπαράσταση προτάσεων. Στην ίδια κατεύθυνση θα πρέπει να γνωρίζει την γλώσσα προδιαγραφών στην οποία θα μοντελοποιήσει τόσο τον εισβολέα όσο και τις έντιμες οντότητες του πρωτοκόλλου, ακολουθώντας πιστά τους κανόνες επικοινωνίας που διέπουν το πρωτόκολλο.

Ένα από τα μειονεκτήματα των τυπικών μεθόδων άρα και του ελέγχου μοντέλων, είναι το φαινόμενο της έκρηξης του χώρου των καταστάσεων. Πρόκειται για την περίπτωση όπου ο αναλυτής, αδυνατεί να ελέγξει το μοντέλο του εξαιτίας της πολυπλοκότητας των εντολών που χρησιμοποιούνται για την δόμηση των διεργασιών του, προκαλώντας την εκτίναξη του χώρου των καταστάσεων, άρα και την αδυναμία της εξαντλητικής ανάλυσής του. Ειδικά για τα πρωτόκολλα ασφαλείας, όπου συνήθως εμπεριέχουν μηχανισμούς επαλήθευσης και αυθεντικοποίησης των οντοτήτων, η επακριβείς αποτύπωσή τους μαζί με την αλληλεπίδραση ενός μοντέλου εισβολέα (τύπου DY) οδηγεί κατά πλειοψηφία στο φαινόμενο ΕΧΚ.

Σε κάθε περίπτωση, ανακύπτει με ιδιαίτερη έμφαση η ανάγκη σχεδιασμού και υλοποίησης ενός εξειδικευμένου μοντέλου εισβολέα ο οποίος θα επιτρέπει από την μια να διενεργεί εχθρικές ενέργειες προς το πρωτόκολλο και από την άλλη να μην οδηγεί την διαδικασία του ελέγχου μοντέλων σε έκρηξη του χώρου των καταστάσεων. Το γεγονός αυτό ενισχύει την άποψη όπου το σώμα της διεργασίας του εισβολέα θα πρέπει να είναι προσεκτικά ορισμένο, από διακριτές ενέργειες οι οποίες δεν θα εκτινάσσουν τις καταστάσεις που παράγονται από τον ελεγκτή μοντέλων, οπότε και θα καθιστούν δυνατή την διενέργεια επαλήθευσης των ιδιοτήτων ασφαλείας που επιθυμεί να ελέγξει ο αναλυτής.

Παράρτημα Α

Η γραμμική χρονική λογική (Linear Temporal Logic, LTL)

Στη γραμμική χρονική λογική (Linear Temporal Logic) ο χρόνος είναι γραμμικά διατεταγμένος και συνήθως μετράται με φυσικούς αριθμούς. Σε ένα γραμμικό πλαίσιο (S,R), το R είναι μία συνάρτηση που αναθέτει σε κάθε χρονικό στιγμιότυπο, το αμέσως επόμενο. Η χρονική διάταξη ≤ στην LTL είναι μία συνολική διάταξη. Για παράδειγμα για κάθε δύο χρονικά στιγμιότυπα s, t ∈ S θα είναι είτε s≤t είτε t≤s. Σε αυτό το σημείο πρέπει να δοθεί η σημασιολογία αυτής της λογικής: στο πλαίσιο (N, Succ) το N είναι το σύνολο των φυσικών αριθμών και το Succ {Succ(n) =n+1} είναι η συνάρτηση αυτού του συνόλου. Η γλώσσα της γραμμικής χρονικής λογικής αποτελείται από τους παρακάτω τέσσερις τελεστές.

Πίνακας 7.3.1 Συμβολισμοί γλώσσας προδιαγραφών ιδιοτήτων LTL

Σύμβολο	***ενέργειες***
○	unary τελεστής, διάβασε «την επόμενη φορά»
□	unary τελεστής, διάβασε «πάντα»
U	binary τελεστής, διάβασε «μέχρι»
◊	unary τελεστής, διάβασε «τελικά»

Αν τα f και g είναι τελεστές τότε είναι και τα $\circ f$, $\Box f$, $\Diamond f$, f **U** g.

Στην LTL η μετάφραση *I=(N, Succ,I)* αναθέτει μια πραγματική τιμή σε οποιαδήποτε LTL φόρμουλα σε οποιοδήποτε στιγμιότυπο $s \in N$ με τον ακόλουθο τρόπο: $\rightarrow I(s,f)=I(s,f)$, *iff* $f \in P$

Λογικοί τελεστές:

$I(s, f \wedge g) = I(s, f) \wedge I(s, g)$

$I(s, \neg f) = \neg I(s, f)$

Χρονικοί τελεστές:

$I(s, \circ f) = I(s+1, f)$

$I(s, f \mathbf{U} g) = true$ *iff* $\exists j \in N . I(s+j, g) = true$ *και* $\forall 0 \leq i < j, I(s+i, f) = true$

$I(s, \Box f) = true$ *iff* $I(s+i, f) = true, \forall i \geq 0$

$I(s, \Diamond f) = true$ *iff* $\exists j \in N . I(s+j, f) = true$

Έστω *M=(S,q,P,L,R)* μία δομή Kripke [29] ενός προγράμματος. Κάθε μονοπάτι $p=(s_0, s_1, s_2, \ldots)$ στο M προσδιορίζει μία χρονική μετάφραση i ως μια ακολουθία αναθέσεων πραγματικών τιμών σε ατομικές προτάσεις στο P, με τον ακόλουθο τρόπο: σε κάθε στιγμιότυπο $n \in N$ η πρόταση $m \in P$ είναι αληθής στο $n, iff\ m \in L(s_n)$.

Λέμε ότι ένα μονοπάτι $p=(s_0, s_1, s_2 \ldots)$ ικανοποιεί μία φόρμουλα *f*, αν η *f* είναι *true* στο *s* στη μετάφραση που προσδιορίζεται από το *p*. Επίσης ένα πρόγραμμα *M* ικανοποιεί μία φόρμουλα *f*, αν κάθε μονοπάτι που ξεκινάει από την αρχική κατάσταση *q* του *M* ικανοποιεί την *f*. Κατά συνέπεια ένα πρόγραμμα ικανοποιεί μία ιδιότητα της γραμμικής χρονικής λογικής, αν όλες οι πιθανές εκτελέσεις του προγράμματος ικανοποιούν την ιδιότητα αυτή.

Οι Clarke et al, [31] πρότεινε LTL model checking αλγόριθμους που είναι εκθετικοί στο μήκος της φόρμουλας, αλλά γραμμικοί στο μέγεθος του μοντέλου. Βασιζόμενοι σε αυτό το συμπέρασμα πολλοί ερευνητές συμφώνησαν ότι η LTL μπορεί να χρησιμοποιηθεί και για μικρές φόρμουλες. Όταν λέμε μήκος φόρμουλας εννοούμε τον αριθμό των συμβόλων (τελεστές, ατομικές προτάσεις και άλλα) που χρησιμοποιούνται για την αναπαράσταση της φόρμουλας.

Οι αλγόριθμοι που προτάθηκαν βασίζονται σε Buchi αυτόματα, τα οποία είναι πεπερασμένα αυτόματα που δέχονται άπειρες λέξεις. Αυτή η προσέγγιση έχει χρησιμοποιηθεί σε ένα μεγάλο αριθμό εργαλείων που κάνουν χρήση της γραμμικής χρονικής λογικής. Η ιδέα είναι απλή: έχει διαπιστωθεί ότι μία LTL φόρμουλα f μπορεί να μετατραπεί σε Buchi αυτόματο που δέχεται ακριβώς τις άπειρες λέξεις που ικανοποιεί η δοθείσα φόρμουλα. Η μετατροπή μπορεί να γίνει με έναν αποδοτικό αλγόριθμο, αλλά επειδή στη χειρότερη περίπτωση το μέγεθος του αυτομάτου μεγαλώνει εκθετικά με το μήκος της φόρμουλας, η μέθοδος είναι καλύτερη για μικρές φόρμουλες.

Η πιθανοκρατική χρονική λογική (Probabilistic Computation Tree Logic, PCTL)

Δύο βασικοί τελεστές για να ορίζονται ιδιότητες είναι ο τελεστής P και ο τελεστής S. Ο τελεστής P μπορεί να χρησιμοποιηθεί και στα τρία είδη μοντέλων και επιτρέπει την εξαγωγή συμπερασμάτων για την πιθανότητα να παρατηρηθεί

μια συγκεκριμένη συμπεριφορά στο μοντέλο. Γενικά: P bound[pathproperty] είναι true για μια κατάσταση s, αν η πιθανότητα να ικανοποιείται η ιδιότητα pathprop ικανοποιεί το όριο bound. Ένα τυπικό παράδειγμα είναι το εξής: P>0.98[pathprop], αυτό σημαίνει πως η πιθανότητα να ικανοποιείται η pathprop ξεπερνά το 0.98.

Για ένα DTMC η μέτρηση της πιθανότητας για ένα σύνολο μονοπατιών που ξεκινούν από μια κατάσταση s, είναι σαφώς ορισμένη. Επίσης για ένα CTMC η πιθανότητα αυτή μπορεί να οριστεί. Αντιθέτως για ένα MDP, η μέτρηση της πιθανότητας μπορεί μόνο να προσεγγιστεί, αφού όλα τα στοιχεία μη-ντετερμινισμού έχουν απομακρυνθεί. Ως εκ τούτου η πραγματική ερμηνεία του : P bound [pathprop], είναι η πιθανότητα ώστε η ιδιότητα pathprop να ικανοποιείται από τα μονοπάτια που ξεκινούν από την αρχική κατάσταση s και να βρίσκουν το όριο bound για όλες τις πιθανές μη-ντετερμινιστικές αποφάσεις. Αυτό σημαίνει ότι, για ένα MDP, οι ιδιότητες που χρησιμοποιούν τον τελετή P, ουσιαστικά αναφέρονται στην ελάχιστη και μέγιστη πιθανότητα, από όλες τις μη-ντετερμινιστικές αποφάσεις ,ώστε να παρατηρείται μια συμπεριφορά. Αυτό εξαρτάται από ένα όριο που συνοδεύει τον τελεστή P. Ένα κατώτερο όριο(> ή >=) αναφέρεται σε ελάχιστες πιθανότητες και ένα ανώτερο όριο(< ή <=) σε μέγιστες πιθανότητες.

Η ιδιότητα prop1 U prop2 είναι αληθής για ένα μονοπάτι, εάν η prop2 είναι αληθής για κάποια κατάσταση και η prop1 είναι αληθής σε όλες τις προηγούμενες καταστάσεις. Μια κοινή εφαρμογή για αυτού του είδους την ιδιότητα είναι όταν η prop1 είναι true. Ένα τυπικό παράδειγμα είναι το εξής: P>0.5 [true U z=2], το οποίο είναι αληθές σε μια κατάσταση, εαν η πιθανότητα του z να είναι ίσο με 2 είναι μεγαλύτερη του 0.5 .

Οι ιδιότητες του τύπου bounded until είναι μια γενίκευση των ιδιοτήτων until, όπου ένα όριο προστίθεται ενώ το δεύτερο όρισμα (prop2) πρέπει να ικανοποιείται. Λόγω του ότι τα DTMCs και τα MDPs προχωρούν με διακριτά βήματα χρόνου, εν αντιθέσει με τα CTMCs που είναι μοντέλα συνεχούς χρόνου, οι δύο περιπτώσεις αντιμετωπίζονται με διαφορετικό τρόπο από τον τελεστή.

Σε μια ιδιότητα τύπου "bounded until" με prop1 Utime prop2 που αφορούν dtmc ή mdp, το τμήμα του ορισμού που αφορά τον χρόνο πρέπει να είναι της μορφής “<=t”, όπου t είναι μια έκφραση PRISM που αντιστοιχεί σε έναν μη

αρνητικό ακέραιο. Η ιδιότητα ικανοποιείται για ένα μονοπάτι αν η prop2 γίνει αληθής εντός κάποιων χρονικών βημάτων και η prop1 είναι αληθής σε όλα τα βήματα πριν από το σημείο όπου η prop2 γίνεται αληθής. Ένα παράδειγμα είναι το: P>=0.98 [true U<=7 y=4], που είναι αληθές σε μια κατάσταση αν η πιθανότητα του υ να είναι ίσο με 4, εντός 7 χρονικών βημάτων, είναι μεγαλύτερη ή ίση από 0.98 . Σε μια ιδιότητα τύπου "bounded until" με prop1 U time prop2 που αφορούν CTMC, το τμήμα του ορισμού που αφορά τον χρόνο μπορεί να είναι οποιοδήποτε από τα παρακάτω: >=t, <=t ή [t1,t2], όπου t,t1 και t2 είναι εκφράσεις PRISM που αντιστοιχούν σε μη αρνητική μεταβλητή τύπου double και με t1<=t2. Το μονοπάτι ικανοποιεί την ιδιότητα αν η prop2 είναι αληθής σε χρόνο που η prop1 είναι true, αλλά και σε όλους τους προηγούμενους χρόνους πριν τη στιγμή αυτή. Για παράδειγμα: P>0 [y<=1 U[5.5,6.5] y>1] ,μεταφράζεται ως η πιθανότητα του y να ξεπεράσει το 1 σε χρόνο μεταξύ του 5.5 και 5.6 είναι μεγαλύτερη του μηδέν. Με την χρήση του τελεστή “bounded until” μπορούμε να αναφερθούμε σε μια συγκεκριμένη χρονική στιγμή, όπως εδώ: *P<0.01 [true U[10,10] y=6*].

Όλες οι ιδιότητες που περιγράφτηκαν στην προηγούμενες ενότητες αντιστοιχούν σε μια τιμή bool. Αν και η ικανοποίηση μιας ιδιότητας ορίζεται για μια κατάσταση συγκεκριμένα, όταν αναλύεται ένα μοντέλο το PRISM θεωρεί την ιδιότητα αληθή, αν ικανοποιείται σε όλες τις καταστάσεις του μοντέλου και ψευδή στην αντίθετη περίπτωση. Για να ελεγχθεί αν μια ιδιότητα ικανοποιείται σε ένα υποσύνολο καταστάσεων, για παράδειγμα τις αρχικές καταστάσεις, οι ιδιότητες μπορούν να έχουν ένα πρόθεμα κάποιας συγκεκριμένης σημασίας. Για παράδειγμα: *"init" => P>=1 [true U leader_elected=true]*

Είναι χρήσιμο να γνωρίζουμε την πραγματική πιθανότητα να παρατηρηθεί κάποια συγκεκριμένη συμπεριφορά στο μοντέλο και όχι μόνο αν κάποια πιθανότητα ξεπερνά ένα όριο ή όχι. Το PRISM επιτρέπει ιδιότητες του παρακάτω τύπου: *P=?[...]* ή *S=?[...].* Αυτές οι πιθανότητες επιστρέφουν μια αριθμητική τιμή ως αποτέλεσμα. Πρέπει να σημειωθεί πως το όριο της πιθανότητας που εφαρμόζεται σε τελεστές P ή S μπορεί να αντικατασταθεί με « =? », εαν είναι έσχατος τελεστής της ιδιότητας που εμφανίζεται, αλλιώς η σημασιολογία δεν ορίζεται σαφώς.

Όλοι αυτοί οι τελεστές επιστρέφουν μια αριθμητική τιμή. Στην απλούστερη περίπτωση όπου το μοντέλο έχει μια μοναδική αρχική κατάσταση, η αριθμητική τιμή που επιστρέφει είναι αυτή που αντιστοιχεί σε αυτήν την κατάσταση. Για παράδειγμα: *P=? [true U x=5&y=5],* επιστρέφει την πιθανότητα του: *από την αρχική κατάσταση, το μοντέλο να μεταβεί σε κατάσταση όπου ικανοποιούνται τα εξής: x=5 και y=5*.

Βιβλιογραφία

[1] Agha, G., Greenwald, M., Gunter, C. A., Khanna, S., Meseguer, J., Sen, K., Thati, P. Formal modelling and analysis of DoS using probabilistic rewrite theories, Proceedings of the IEEE Workshop on Foundations of Computer Security (FCS '05), Chicago, 2005

[2] Amadio R. M. and Charatonik W., "On Name Generation and Set-Based Analysis in the Dolev-Yao Model", In Proc. of CONCUR, Springer-Verlag LNCS 2421, pp.499-514, 2002.

[3] Audun J., "Security Protocol Verification using SPIN". Proceedings of the First SPIN Workshop, INRS-Telecommunications, Montreal, Canada, 1995.

[4] Aura, T., Nikander, P., Leiwo, J. DOS-resistant authentication with client puzzles, Proceedings of the Security Protocols Workshop, Cambridge, UK, LNCS 2133, Springer Verlag, 170-181, 2001

[5] AVISPA: Automated validation of internet security protocols and applications, 2003, FET Open Project IST-2001-39252, http://www.avispa-project.org

[6] Basagiannis, S., Katsaros, P., Pombortsis, A., An intruder model with Message Inspection for model checking security protocols, (In Press) Computers & Security, Elsevier, 2009.

[7] Basagiannis, S. Katsaros, P. and Pombortsis, A., "Synthesis of Attack Actions Using Model Checking for the Verification of Security Protocols" (in Press) Wiley & Sons, Security and Communication Journal, 2009.

[8] Basagiannis, S., Katsaros, P. and Pombortsis, A., Alexiou, N., "Probabilistic model checking for the quantification of DoS security threats", Computers & Security, Vol. 28 (6), 450-465, Elsevier September 2009.

[9] Basagiannis, S. Katsaros, P. and Pombortsis, A. and Alexiou N. ,"A Probabilistic Attacker Model for Quantitative Verification of DoS Security Threats", In Proceedings of the 32nd IEEE international Computer Software and Applications conference, COMPSAC '08, Turku, Finland, 28 July – 1 August 2008.

[10] Basagiannis, S. Katsaros, P. and Pombortsis A., "Intrusion Attack Tactics for the Model Checking of E-Commerce Security Guarantees" ,Lectures Notes in Computer Science 4680, 238-252, Springer Verlag, Nuremberg, Germany, SAFECOMP '07 in the 26th International Conference on Computer Safety, Reliability and Security 18 - 21 September 2007.

[11] Basagiannis, S. Katsaros, P. and Pombortsis, A., "Interlocking Control by Distributed Signal Boxes: Design and Verification with the SPIN Model Checker", Lectures Notes in Computer Science 4330, 317-328, Springer Verlag, Sorrento, Italy, ISPA' '06, in the Fourth International Symposium on Parallel and Distributed Processing and Applications, 4-6 December 2006.

[12] Basagiannis, S. Katsaros, P. and Pombortsis, A.,"State Space Reduction with Message Inspection in Security Protocol Model Checking", In Corr eprint archieve, arXiv:0909.0174, informal publication, (online http://arxiv.org/abs/0909.0174), September 2009.

[13] Basagiannis, S. Katsaros, P. and Pombortsis, A.,"Attacking an OT-Based Blind Signature Scheme", In Corr eprint archieve, arXiv:0906.2947v3, informal publication, (online http://arxiv.org/abs/0906.2947), June 2009.

[14] Basagiannis, S., Katsaros, P. and Pombortsis, A., " Network Interlocking Control", submitted to journal, September 2009.

[15] Basagiannis, S., Katsaros, P. and Pombortsis, A., "Fail-Safe Network Interlocking Control", submitted to journal, September 2009.

[16] Basin D., Modersheim S., Vigano L., OFMC: A Symbolic Model-Checker for Security Protocols, International Journal of Information Security, 2004.

[17] Basu A., Saddek Bensalem, Doron Peled, Joseph Sifakis: Priority Scheduling of Distributed Systems Based on Model Checking. CAV 2009: 79-93

[18] Black P. E., Hall K. M., Jones M. D., Larson T. N., Windley P. J., "A Brief Introduction to Formal Methods". Proceedings of the IEEE 1996, Custom Integrated Circuits Conference (CICC '96), San Diego, California, USA, pages 377-380, 1996.

[19] Briand, L., Bunse, C., Daly, J., and Differding, C., "An experimental comparison of the maintainability of object-oriented and structured design documents", Empirical Software Engineering, Vol. 2, Issue 3, pp. 291–312, 1997.

[20] Burrows M., Abadi M., Needham R., A logic of authentication, ACM Transaction on Computer Systems 8/1, pp. 18-36, 1990. (38)

[21] Buttyan L., "Formal methods in the design of cryptographic protocols". Swiss Federal Institute of Technology, ICA EPFL-Di-ICA, available, 1999.

[22] Carlsen U., Cryptographic protocol flaws – Know your enemy, In Proc. of the 7th IEEE Computer Security Foundations Workshop, IEEE Computer Society, pp. 192-200, 1994.

[23] Cederquist J.G., Dashti M.T., An intruder model for verifying liveness in security protocols, In Proc. of the fourth ACM workshop on Formal methods in security (FMSE '06), Alexandria, Virginia, USA, pp. 23-32, 2006.

[24] CERT Coordination Center. Denial of service attacks. http://www.cert.org/tech_tips/denial_of_service.html, (last access: 28th of December 2007)

[25] Cervesato I., "The Dolev-Yao intruder is the most powerful attacker", In Proc. of the 16th Annual Symposium on Logic in Computer Science (LICS), IEEE Computer Society Press, 2001.

[26] Chevalier Y., Kuesters R., Rusinowitch M., Turuani M. and Vigneron L. "Extending the Dolev-Yao Intruder for Analyzing an Unbounded Number of Sessions", Computer Science Logic (CSL) and 8th Kurt Goedel Colloquium (8th KCG), 2003.

[27] Cimatti, A., Clarke, E. M., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, M. and Tacchella, A., "NuSMV 2: An OpenSource Tool for Symbolic Model Checking", In Proc. of the International Conference on Computer-Aided Verification (CAV), Copenhagen, Denmark, 2002.

[28] Clarke E. M., Jha S., Marrero W., Verifying security protocols with Brutus, ACM Transactions on Software Engineering and Methodology 9/4, pp. 443-487, 2000.

[29] Clarke M. E., Grumberg O. and Peled D. A., "Model Checking", MIT Press, 1999

[30] Clarke, E., Jha, S. and Marrero, W., "Partial order reduction for security protocol verification", In Proc. of the 6th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS), Berlin, Germany, 2000.

[31] Clarke E. , Orna Grumberg , David E. Long, Model checking and abstraction, ACM Transactions on Programming Languages and Systems (TOPLAS), v.16 n.5, p.1512-1542, Sept. 1994

[32] Clark J., Jacob J., A survey of authentication protocol literature: version 1.0, Technical Report, University of York, 1997.

[33] Cremers C. J. F., Feasibility of multi-protocol attacks, In Proc. of the First International Conference on Availability, Reliability and Security, IEEE Computer Society Press, 2006.

[34] Dolev D., Yao A., On the security of public-key protocols, IEEE Transactions on Information Theory 2/29, pp. 198-208, 1983.

[35] Gärtner, F. Revisiting liveness properties in the context of secure systems, Proceedings of the 1st International Conference on Formal Aspects on Security, Springer LNCS 2629, 203-211, 2003

[36] Gritzalis S., Spinellis D., "Addressing threats and security issues in World Wide Web technology", Proceedings of the IFIP joint TC6/TC11 Working Conference on Communications and Multimedia Security, pp.33-46, Athens, Greece, September 1997.

[37] Gritzalis S., Katsikas S. and Gritzalis D. "Computer Network Security", (in Greek), Papadotiriou publications, ISBN: 960-7530-45-4, 2003.

[38] Gritzalis S., Spinellis D., Georgiadis, P., Security protocols over open networks and distributed systems: formal methods for their analysis, design, and verification, Computer Communications 22, pp.697-709, 1999.

[39] Godefroid P. "Partial-Order Methods for the Verification of Concurrent Systems: An Approach to the State-Explosion Problem Pages: 142 Year of Publication: 1996 ISBN:3540607617

[40] Gregoire, J.-C., "State space compression in SPIN with GETSs", In Proc. of the 2nd SPIN Workshop, pp. 90-108, 1996.

[41] Grunske, L., Joyce, D., "Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles", Journal of Systems and Software, Vol. 81, pp.1327-1345, 2008.

[42] Hamdi M., Boudriga N., Computer and network security risk management: Theory, challenges, and countermeasures, International Journal of Communication Systems, Vol. 18(8), pp.763-793, 2005.

[43] Heather J., Lowe G., Schneider S., How to prevent type flaw attacks on security protocols, In Proc. of the 13th IEEE Computer Security Foundations Workshop, IEEE Computer Society, pp, 255-268, 2000.

[44] Henzinger T., Xavier Nicollin, Joseph Sifakis, Sergio Yovine: Symbolic Model Checking for Real-Time Systems Inf. Comput. 111(2): 193-244 (1994)

[45] Holzmann G. J., Design and Validation of Computer Protocols. Prentice-Hall, Englewood Cli.s, New Jersey, USA, 1991.

[46] Holzmann G. J., "The Spin Model Checker - Primer and Reference Manual", Addison-Wesley, 2003.

[47] Host Identity Protocol, Internet Engineering Task Force (IETF) - Network Working Group, Internet Draft, February 2007.

[48] Huang D., Sinha A. and D. Medhi, "A double authentication scheme to detect impersonation attack in link state routing protocols", In Proc. of the IEEE International Conference on Communications (ICC), 2003.

[49] InfraHIP Project Web Site, http://infrahip.hiit.fi/, (last access: 21st of December 2007)

[50] Internet Engineering Task Force (IETF) - Network Working Group. Host Identity Protocol. Internet Draft, February 2007

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

[52] Karn, P. and Simpson, A. Photuris: Session-key management protocol, RFC 2522, IETF Network Working Group, 1999

[53] Kim K., Abraham J. A. and Bhadra J., "Model Checking of Security Protocols with Pre-configuration", In Proc. of the 4th International Workshop on Information Security Applications, WISA, Korea, LNCS 2908, Springer-Verlag, pp.1–15, 2003.

[54] Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K. and Rosenberg, J. "Preliminary guidelines for empirical research in software engineering", IEEE Transactions on Software Engineering, Vol. 28, Issue 8, pp. 721 – 734, 2002.

[55] Kremer S., Markowitch O., Zhou J., An intensive survey of fair non-repudiation protocols, Computer Communications, 25/17, pp. 1606-1621, 2002.

[56] Kwiatkowska, M., Norman, G., Parker, D. Stochastic model checking, In: Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07), Ed: M. Bernardo and J. Hillston, Springer LNCS 4486, 220-270, 2007

[57] Kwiatkowska, M. Quantitative verification: Models, Techniques and Tools, Proceedings of the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE), ACM Press, 449-458, 2007

[58] Liao Q., Cieslak D. A., Striegel A. D., Chawla N. V., Using selective, short-term memory to improve resilience against DDoS exhaustion attacks, in Security and Communication Networks, vol. 1, no. 4, pp. 287-299, Jul/Aug 2008.

[59] Lin F. J., P. M. Chu , M. T. Liu, Protocol verification using reachability analysis: the state space explosion problem and relief strategies, ACM SIGCOMM Computer Communication Review, v.17 n.5, p.126-135, Oct./Nov. 1987

[60] Lopez, J., Ortega, J. J., Troya, J. M., "Protocol engineering applied to formal analysis of security systems", In Proc. of Int. Conf. of Infrastructure Security, Bristol, UK, LNCS 2437, Springer-Verlag, pp. 246-259, 2002.

[61] Lowe G., Casper: a compiler for the analysis of security protocols, In Proc. of the IEEE Computer Security Foundations Workshop, IEEE Computer Society, pp. 18-30, 1997.

[62] Lowe G., Towards a completeness result for model-checking of Security Protocols, In Proc. of the 11th Computer Security Foundations Workshop. IEEE Computer Society Press, 1998.

[63] Lowe G., "An attack on the Needham-Schroeder public key authentication protocol", Information Processing Letters, Vol. 56 (3), pp.131-136, 1995.

[64] Madan, B. B., Goseva-Popstojanova, K., Vaidyanathan, K., Trivedi, K. S. Modeling and quantification of security attributes of software systems, Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 02), IEEE Computer Society Press, 2002

[65] McMillan K. L.. Symbolic Model Checking: An Approach to the State Explosion Problem. Kluwer Academic Publishers, 1993.

[66] Meadows, C. A cost-based framework for analysis of DoS in networks, Journal of Computer Security, 9 (1/2), 143-164, 2001

[67] Meadows, C. Formal methods for cryptographic protocol analysis: emerging issues and trends, IEEE Journal on Selected Areas in Communications, 21 (1), 44-54, 2003

[68] Meadows C. A., Formal verification of cryptographic protocols: A survey, Advances in Cryptology International Conference on the Theory and Application of Cryptology (Asiacrypt), LNCS 917 Springer-Verlag, pp. 133-150, 1995.

[69] Meadows C., Kemmerer R., Millen, J., Three systems for cryptographic protocol analysis, Journal of Cryptology 7/2, pp.79-130, 1994.

[70] Message Inspection Intruder Modeling Framework, online: http://mathind.csd.auth.gr/mi_work.html (last accessed 13/2/2009).

[71] Millen J. K., Clark S. C., Freedman S. B., The Interrogator: Protocol Security Analysis, IEEE Transactions on Software Engineering Vol13/2, 1987.

[72] Mitchell J. C., Mitchell M., Stern U., Automated analysis of cryptographic protocols using Murφ, In Proc. of the IEEE Symposium on Research in Security and Privacy, IEEE Computer Society, pp. 141-153, 1997.

[73] Najm E. and F. Olsen, "Reactive EFSMs, Reactive PROMELA/ RSPIN," Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS96), pp. 349-368, Passau, Germany, Lecture Notes In Computer Science 1,055, Springer-Verlag, Mar. 1996.

[74] Needham, R. M. and Schroeder, M. D., "Using Encryption for Authentication in Large Networks of Computers," Communications of the ACM 21, 1978.

[75] Nessett D, A critique of the Burrows, Abadi and Needham logic (ACM SIGOPS) Operating Systems Review, v.24 n.2, pp.35-38, 1990.

[76] Obaidat M. S., A Methodology for Improving Computer Access Security, Computers & Security, Vol. 12, pp.657-662, 1993.

[77] Panti, M., Spalazzi, L. and Tacconi., S., "Using the NUSMV model checker to verify the Kerberos Protocol". In Proc. of the 3rd Collaborative Technologies Symposium (CTS), pp. 27-31, 2002.

[78] Prasad Sistla, Edmund M. Clarke: The Complexity of Propositional Linear Temporal Logics J. ACM 32(3): 733-749 (1985)

[79] Qian Y., Kejie Lu, Bo Rong, Tipper D., A Design of Optimal Key Management Scheme for Secure and Survivable Wireless Sensor Networks, Security and Communication Networks, Vol. 1, No. 1, pp. 75-82, Jan. 2008.

[80] Rivest R. L., The MD5 Message-Digest Algorithm, Internet informational RFC 1321, 1992.

[81] Rivest R. L., Shamir A., Payword and Micromint: Two simple micropayment schemes, In Proc. of the Fourth International Workshop on Security Protocols, LNCS 1189 Springer-Verlag, pp. 69-87, 1996.

[82] Roscoe A. W., Modeling and verifying key-exchange protocols using CSP and FDR, In Proc. of the 8th IEEE Computer Security Foundations Workshop, IEEE Computer Society, pp. 98-107, 1995.

[83] Roscoe A. W., The theory and practice of concurrency, Prentice Hall, 1997.

[84] Roscoe A. W., Goldsmith, M., The perfect spy for model-checking cryptoprotocols, In Proc. of the Workshop on Design and Formal Verification of Security Protocols (DIMACS), 1997.

[85] Sen, K., Viswanathan, M., Agha, G. VESTA: A statistical model-checker and analyzer for probabilistic systems, Proceedings of the 2nd Int. Conference on the Quantitative Evaluation of Systems (QEST'05), 251-252, 2005

[86] Sen, K., Viswanathan, M., Agha, G. On statistical model checking of stochastic systems, Proceedings of the 17th Int. Conference on Computer Aided Verification (CAV'05), Springer LNCS 3576, 266-288, 2005

[87] Shmatikov V., Mitchell J. C., Finite-state analysis of two contract signing protocols, Theoretical Computer Science, 283, pp. 419-450, 2002.

[88] Shmatikov, V. and Stern, U., "Efficient finite-state analysis for large security protocols", In Proc. of the 11th Workshop on Computer Security Foundations (CSFW), pp. 106-120, 2000.

[89] Song D., "Athena: a new efficient automatic checker for security protocol analysis". In P. Syverson, ed., In Proc. of the 12th IEEE Computer Security Foundations Workshop, Italy, IEEE Computer Society Press, pp.192-202, 1992.

[90] Syverson P., Cervesato I., The logic of authentication protocols, In Proc. of the 1st International School on Foundations of Security Analysis and Design (FOSAD 2000), LNCS 2171, Springer-Verlag, pp. 63-137, 2001.

[91] The PRISM Model Checker Web Site, http://www.prismmodelchecker.org/

[92] Tritilanunt, S., Boyd, C., Foo, E., Gonzalez Neto, J. M. Using Coloured Petri Nets to simulate DoS-resistant protocols, Proceedings of the Seventh Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, University of Aarhus, Denmark, 2006

[93] The SPIN model checker official website, available at http://spinroot.com/ (last accessed 12/12/2008).

[94] Valmari A. : Stubborn sets for reduced state space generation. Applications and Theory of Petri Nets 1989: 491-515

[95] Valmari A.: Compositional State Space Generation. Applications and Theory of Petri Nets 1991: 427-457

[96] Valmari A.: On-the-Fly Verification with Stubborn Sets. CAV 1993: 397-408

[97] Valmari A. : The State Explosion Problem. Petri Nets 1996: 429-528

[98] Wohlin, C., Petersson, H., Aurum, A., Shull, F. and Ciolkowski, M., "Software inspection benchmarking – A qualitative and quantitative comparative opportunity", In Proc. of the 8th IEEE International Symposium on Software Metrics, IEEE Computer Society, pp. 118-127, 2002.

[99] Woo T. Y. C., Lam S. S., A semantic model for authentication protocols, In Proc. of the IEEE Symposium on Research in Security and Privacy, 1993.

[100] Zhang, C., Zhou, M. C. and Yu, M., "Ad hoc network routing and security: A review", International Journal of Communication Systems, Vol. 20 (8), pp.909-925, 2007.

Κατάλογος Δημοσιεύσεων

Διεθνή Περιοδικά με σύστημα κριτών

Basagiannis, S., Katsaros, P., Pombortsis, A., An intruder model with Message Inspection for model checking security protocols, (In Press) Computers & Security, Elsevier (online: http://dx.doi.org/10.1016/j.cose.2009.08.003), 2009.

Basagiannis, S. Katsaros, P. and Pombortsis, A., "Synthesis of Attack Actions Using Model Checking for the Verification of Security Protocols" (in Press) Wiley & Sons, Security and Communication Journal, (online: http://dx.doi.org/10.1002/sec.119), 2009.

Basagiannis, S., Katsaros, P. and Pombortsis, A., Alexiou, N., "Probabilistic model checking for the quantification of DoS security threats", Computers & Security, Vol. 28 (6), 450-465, Elsevier (online: http://dx.doi.org/10.1016/j.cose.2009.01.002), September 2009.

Διεθνή Συνέδρια με σύστημα κριτών

Basagiannis, S. Katsaros, P. and Pombortsis, A. and Alexiou N. ,"A Probabilistic Attacker Model for Quantitative Verification of DoS Security Threats", In Proceedings of the 32nd IEEE international Computer Software and Applications conference, COMPSAC '08, Turku, Finland, 28 July – 1 August 2008.

Basagiannis, S. Katsaros, P. and Pombortsis A., "Intrusion Attack Tactics for the Model Checking of E-Commerce Security Guarantees" ,Lectures Notes in Computer Science 4680, 238-252, Springer Verlag, Nuremberg, Germany, SAFECOMP '07 in the 26th International Conference on Computer Safety, Reliability and Security 18 - 21 September 2007.

Basagiannis, S. Katsaros, P. and Pombortsis, A., "Interlocking Control by Distributed Signal Boxes: Design and Verification with the SPIN Model Checker", Lectures Notes in Computer Science 4330, 317-328, Springer Verlag, Sorrento, Italy, ISPA' '06, in the Fourth International Symposium on Parallel and Distributed Processing and Applications, 4-6 December 2006.

Άλλες δημοσιεύσεις

Basagiannis, S. Katsaros, P. and Pombortsis, A.,"State Space Reduction with Message Inspection in Security Protocol Model Checking", In Corr eprint archieve, arXiv:0909.0174, informal publication, (online http://arxiv.org/abs/0909.0174), September 2009.

Basagiannis, S. Katsaros, P. and Pombortsis, A.,"Attacking an OT-Based Blind Signature Scheme", In Corr eprint archieve, arXiv:0906.2947v3, informal publication, (online http://arxiv.org/abs/0906.2947), June 2009.

Εργασίες υπό κρίση:

Basagiannis, S. Katsaros, P. and Pombortsis, A., "Network Interlocking Control by Distributed Signal Boxes" submitted to journal, March 2009.

Basagiannis, S., Katsaros, P. and Pombortsis, A., "Security Analysis of an OT-Based Blind Signature Scheme", submitted, to journal July 2009.

Basagiannis, S., Katsaros, P. and Pombortsis, A., "Fail-Safe Network Interlocking Control", submitted to journal, September 2009.

Basagiannis, S., Katsaros, P., Alexiou N. and Pombortsis, A., "Probabilistic Model Checking of Oblivious Transfer Mechanisms", submitted to journal, October 2009.

Basagiannis, S., Katsaros, P., Paparizos I. and Pombortsis, A., "Quantitative Verification of Song and Mitchell's RFID Protocol Using Probabilistic Model Checking ", submitted to journal, December 2009.