Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών Ηλεκτρονικής και Συστημάτων
Πληροφορικής

Ανάπτυξη εμπιστοσύνης σε κατανεμημένα ad-hoc δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Άγγελος-Μάριος Π. Καπουκάκης

Επιβλέπων : Βασίλειος Μάγκλαρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2009

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών Ηλεκτρονικής και Συστημάτων
Πληροφορικής

Ανάπτυξη εμπιστοσύνης σε κατανεμημένα ad-hoc δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Άγγελος-Μάριος Π. Καπουκάκης

Επιβλέπων : Βασίλειος Μάγκλαρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28η Σεπτεμβρίου 2009.

.....	.....	.....
Βασίλειος Μάγκλαρης	Συμεών Παπαβασιλείου	Δημήτριος Καλογεράς
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Ερευνητής ΕΠΙΣΕΥ

Αθήνα, Σεπτέμβριος 2009

.....

Άγγελος-Μάριος Π. Καπουκάκης

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

Περίληψη

Τα αυτόνομα δίκτυα έχουν ιδιαίτερες απαιτήσεις σε ανταλλαγή πληροφορίας μεταξύ των κόμβων τους. Το ζήτημα της διπλωματικής εργασίας είναι να μελετήσει, κυρίως, πως μπορεί να αναπτυχθεί εμπιστοσύνη μεταξύ των κόμβων/δικτυακών οντοτήτων σε ένα ανοιχτό ad-hoc δίκτυο. Για το σκοπό αυτό αναλύθηκαν μηχανισμοί και αλγόριθμοι από p2p δίκτυα και έγινε έρευνα στην δυνατότητα δυναμικής ανάπτυξης της εμπιστοσύνης.

Επιπρόσθετα, πραγματοποιήθηκε υλοποίηση κάποιων αλγορίθμων και προσομοίωσή τους με την βοήθεια της πλατφόρμας του P2P Simulator. Τέλος, έγινε σύγκριση του τρόπου χτισίματος και επέκτασης τέτοιων συστημάτων εμπιστοσύνης που στηρίζονται στην φήμη των χρηστών τους.

Λέξεις-Κλειδιά: ad-hoc, εμπιστοσύνη, φήμη, ομότιμα δίκτυα, αποκεντρωμένα δίκτυα, συγκεντρωμένα δίκτυα, αυτόνομα συστήματα.

Abstract

The autonomic networks have particular requirements in exchange of information between their nodes. The subject of our diplomatic work is the study, mainly, on how trust can be developed between the nodes/network entities in an open ad-hoc network. For this purpose, mechanisms and algorithms from p2p networks were analyzed and research was done about the possibility of dynamic growth of trust.

Besides, implementation of certain algorithms and their simulation were carried out using the P2P Simulator platform. Finally, we compared the different ways of build and extension for such reputation-based trust systems.

Keywords: ad-hoc, reputation-based trust, peer-to-peer, decentralized networks, centralized networks, autonomic systems.

Αφιερώνεται στην μητέρα μου Ευτυχία

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Βασίλη Μάγκλαρη για την δυνατότητα που μου έδωσε να εκπονήσω την διπλωματική μου εργασία, καθώς και το ενδιαφέρον που έδειξε να αναλάβει την επίβλεψή της. Εν συνεχεία να ευχαριστήσω τον υποψήφιο διδάκτορα Βασίλη Μερεκούλια για την καθοδήγηση και την συμβολή του στην διεκπεραίωση της διπλωματικής εργασίας αυτής. Τέλος, να ευχαριστήσω θερμά την Ελένη Βάθη για την πολύτιμή βοήθεια της στην επιμέλεια της εργασίας, αλλά και για τις συμβουλές της στο κομμάτι του προγραμματισμού.

Πίνακας περιεχομένων

1 Εισαγωγή

1.1 Εμπιστοσύνη στο διαδίκτυο

> ... η εμπιστοσύνη είναι ένα κοινωνικό αγαθό που πρέπει να προστατεύεται ακριβώς όπως ο αέρας που αναπνέουμε ή το ύδωρ που πίνουμε. Όταν βλάπτεται, η κοινότητα συνολικά υποφέρει, και όταν καταστρέφεται, οι κοινωνίες υποχωρούν και καταρρέουν.
>
> Bok, 1978, pp 26 and 27.

Εμπιστοσύνη είναι η ικανότητα ενός ανθρώπου να εκπληρώσει τις νόμιμες προσδοκίες που έχουν άλλοι άνθρωποι από αυτόν. Η ύπαρξη οποιασδήποτε κοινότητας, βασίζεται στην έννοια της εμπιστοσύνης μεταξύ των μελών της. Σε ένα μεγάλο μέρος της καθημερινής μας ζωής, λαμβάνονται αποφάσεις που σχετίζονται με την έννοια της εμπιστοσύνης είτε με άμεσο, είτε με έμμεσο τρόπο. Σκεφτείτε το απλό παράδειγμα αγοράς κάποιου αντικειμένου από ένα κατάστημα. Μπορεί να επιλέξουμε να αγοράσουμε ένα αντικείμενο συγκεκριμένης φίρμας, επειδή το έχουμε κρίνει αξιόπιστο από προηγούμενες αγορές στο παρελθόν ή διότι έχει την φήμη ευρέως πως είναι αξιόπιστο. Άμεσα εμπιστευόμαστε ότι το κατάστημα πωλεί γνήσια προϊόντα και όχι πλαστά προϊόντα της φίρμας. Κατά τη διάρκεια της αγοράς, η συναλλαγή με πιστωτική κάρτα γίνεται με ένα ηλεκτρονικό σύστημα, που ο ταμίας εμπιστεύεται. Εάν η μηχανή απορρίψει την πιστωτική κάρτα, ο πελάτης είναι συνήθως ο ύποπτος και όχι το σύστημα συναλλαγής. Επίσης, υπάρχει μια καθολική εμπιστοσύνη για την υγεία του νομισματικού συστήματος, ώστε να έχουμε συναλλαγές μετρητών.

Σύμφωνα με τον Luhmann, η εμπιστοσύνη είναι, επίσης, ένα εργαλείο για τη μείωση της πολυπλοκότητας. Αυτό επιτυγχάνεται, καθώς η εμπιστοσύνη παρέχει την εσωτερική ασφάλεια προτού ληφθεί κάποια ενέργεια, παρά την αβεβαιότητα και τις ελλιπείς πληροφορίες που μπορεί να υπάρχουν.

Πολλές άλλες ορολογίες για την εμπιστοσύνη έχουν διατυπωθεί. Μερικές, κατατάσσουν την εμπιστοσύνη ως κάτι ουσιαστικό στην οικονομία και το εμπόριο, στην απορρόφηση της γνώσης, στη διατύπωση μιας αίσθησης αυτογνωσίας και ως βάση της πολιτικής υγείας. Η εμπιστοσύνη έχει μια σιωπηλή παρουσία σε όλη την κοινωνική αλληλεπίδραση.

Η ίδια η κοινωνική αλληλεπίδραση γίνεται γρήγορα μια έννοια που εκτείνεται σε πολλά γεωγραφικά, πολιτικά και πολιτιστικά όρια. Οι εικονικές κοινότητες είναι τόσο πραγματικές, όσο οι κοινότητες που ορίζονται φυσικά ή τα μέλη των οποίων υπάρχουν σε κοντινή -κατάλληλη- απόσταση. Κατά συνέπεια, ότι ρόλο παίζει η εμπιστοσύνη σε αυτές τις «φυσικές» κοινότητες, το ίδιο ισχύει επίσης για τις εικονικές κοινότητες, δεδομένου ότι, τελικά, όλες οι εικονικές αλληλεπιδράσεις είναι ανθρώπινα συνδεδεμένες. Αυτό ισχύει ακόμη και για τεχνητές οντότητες, όπως οι τεχνητοί πράκτορες (artificial agents) δεδομένου ότι δημιουργούνται για να εξυπηρετήσουν ένα ανθρώπινο πρόσωπο και το αποτέλεσμα των αλληλεπιδράσεών τους ανατροφοδοτείται στους ανθρώπους με τη μία μορφή ή την άλλη. Επομένως, είναι ζωτικής σημασίας να υπάρχει ένα ικανοποιητικό μοντέλο εμπιστοσύνης για τις εικονικές κοινότητες έτσι ώστε 1) η αυξανόμενη πολυπλοκότητα μεγάλων κατανεμημένων συστημάτων, όπως το Διαδίκτυο, να μπορεί να διαχειριστεί αποτελεσματικότερα, 2) το ηλεκτρονικό εμπόριο να πραγματοποιείται ομαλά και 3) οι αυτόνομοι πράκτορες να είναι εύρωστοι, ευπροσάρμοστοι και αποτελεσματικοί, παρέχοντάς τους την ικανότητα να κρίνουν τα δεδομένα εμπιστοσύνης.

Το διαδίκτυο ολοένα γίνεται πιο μεγάλο και χρησιμοποιείται καθημερινώς από εκατομμύρια ανθρώπους. Όπως γίνεται κατανοητό, η ανάπτυξη της εμπιστοσύνης στο διαδίκτυο είναι το ίδιο σημαντική με αυτή στις ανθρώπινες σχέσεις. Το ηλεκτρονικό εμπόριο, οι υπηρεσίες ηλεκτρονικής πρόσβασης (πχ στον λογαριασμό της τράπεζας), οι εφαρμογές διαμοιρασμού αρχείων (file sharing applications) και άλλα τέτοια παραδείγματα, έχουν ως κοινό κριτήριο την εμπιστοσύνη των χρηστών που τα χρησιμοποιούν.

Ένα από τα πρώτα μοντέλα εμπιστοσύνης που αναπτύχθηκαν είναι το PGP (Pretty Good Privacy), το οποίο είναι ένα δωρεάν κρυπτογραφικό σύστημα δημοσίου κλειδιού που δημιουργήθηκε από τον Phil Zimmermann το 1991 με βάση το X.509 πρότυπο PKI (Public Key Infrastructure). Η εμπιστοσύνη στο PGP επιτυγχάνεται χρησιμοποιώντας το μοντέλο Web of Trust. Η ελλοχεύουσα ιδέα αυτού του μοντέλου

είναι ότι αποδεχόμαστε το δημόσιο κλειδί ενός χρήστη PGP, εάν ένας ή περισσότεροι άλλοι αξιόπιστοι χρήστες PGP το έχουν υπογράψει. Κάθε χρήστης PGP διατηρεί έναν κατάλογο δημόσιων κλειδιών, που αποκαλείται keyring. Μπορείτε να το φανταστείτε σαν ένα μπρελόκ με τα κλειδιά που χρησιμοποιείτε. Τα keyrings μπορούν να ανταλλαχθούν μεταξύ των χρηστών. Το μοντέλο εμπιστοσύνης X.509 πρότυπο PKI είναι βασισμένο στις νόμιμες αρχές πιστοποιητικών, που παράγουν και διαχειρίζονται τα πιστοποιητικά. Αυτά τα CA τακτοποιούνται σε μια ιεραρχία, αποκαλούμενη αλυσίδα πιστοποίησης.

Τα μοντέλα ανάπτυξης εμπιστοσύνης μπορούν να τοποθετηθούν σε δύο κύριες κατηγορίες. Τα συγκεντρωμένα μοντέλα εμπιστοσύνης (centralized trust models) και τα αποκεντρωμένα μοντέλα εμπιστοσύνης (decentralized trust models) ή αλλιώς κατανεμημένα (distributed). Η πιο σημαντική διαφορά τους έχει να κάνει με τον τρόπο που επιτυγχάνεται η εμπιστοσύνη.

Στα συγκεντρωμένα μοντέλα εμπιστοσύνης υπάρχει μια κεντρική αρχή πιστοποίησης (certificate authority), η οποία είναι μια οντότητα που εκδίδει ψηφιακά πιστοποιητικά για χρήση από άλλα συμβαλλόμενα μέρη. Ουσιαστικά αυτή η οντότητα είναι ένας εξυπηρετητής (server).

Στα κατανεμημένα μοντέλα εμπιστοσύνης , τα οποία πρότεινε ο Alfarez Abdul-Rahman, η εμπιστοσύνη δεν χαρίζεται από κάποιον κεντρικό κόμβο, αλλά κατακτιέται από τους χρήστες του συστήματος με τις καθημερινές αλληλεπιδράσεις με άλλους χρήστες. Αυτή η προσέγγιση είναι βασισμένη στην ύπαρξη δύο διαφορετικών σχέσεων εμπιστοσύνης. Πρώτον, την *άμεση σχέση εμπιστοσύνης*, την οποία έχει αναπτύξει ένας χρήστης για κάποιον άλλο από προηγούμενες συναλλαγές τους και δεύτερον την *έμμεση σχέση εμπιστοσύνης*, που εκφράζεται ως η αξιοπιστία σε χρήστες να συστήνουν άλλους χρήστες με τους οποίους δεν έχει αλληλεπιδράσει ακόμα μαζί τους. Μια σύσταση είναι μεταβιβασμένη πληροφορία εμπιστοσύνης, που περιέχει πληροφορίες φήμης.

Τα πιο χαρακτηριστικά παραδείγματα αποκεντρωμένων μοντέλων εμπιστοσύνης είναι εκείνα που χρησιμοποιούνται στα peer-to-peer δίκτυα, αλλά και στα δίκτυα ηλεκτρονικού εμπορίου, όπως είναι το eBay. Όπως γίνεται αντιληπτό, οι μηχανισμοί εμπιστοσύνης που χρησιμοποιούνται στα αποκεντρωμένα μοντέλα εμπιστοσύνης είναι περισσότερο πολύπλοκοι από αυτούς των συγκεντρωμένων μοντέλων. Αυτό, βέβαια, αντισταθμίζεται από το γεγονός ότι είναι πιο αποδοτικά και με λιγότερα προβλήματα ασφαλείας. Αναλυτικά όλα αυτά θα παρουσιαστούν στα Κεφάλαια 2 και

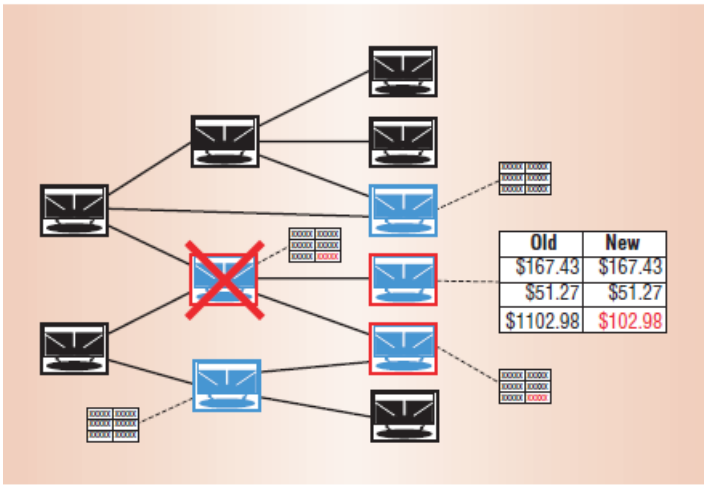
3, όπου θα αναφερθούμε σε συγκεκριμένα παραδείγματα μοντέλων και με περισσότερες λεπτομέρειες.

1.2 Αυτόνομα Συστήματα

Η δυσκολία της διαχείρισης των σημερινών συστημάτων υπολογισμού (computing systems) ξεπερνάει αυτήν της διοίκησης μεμονωμένων περιβαλλόντων λογισμικού. Η ανάγκη να ενσωματωθούν διάφορα ετερογενή περιβάλλοντα στα υπάρχοντα εταιρικά συστήματα υπολογισμού, και να γίνει περαιτέρω επέκταση στις επιχειρήσεις του διαδικτύου, εισάγει νέα επίπεδα πολυπλοκότητας. Ένα γενικό πρόβλημα των σύγχρονων κατανεμημένων συστημάτων υπολογισμού είναι ότι η πολυπλοκότητά τους, και ειδικότερα η πολυπλοκότητα της διαχείρισής τους, γίνεται ένας σημαντικός περιοριστικός παράγοντας στην περαιτέρω ανάπτυξή τους. Οι μεγάλες επιχειρήσεις χρησιμοποιούν τα δίκτυα μεγάλης κλίμακας υπολογιστών για επικοινωνία και εργασίες υπολογισμού. Οι κατανεμημένες εφαρμογές που τρέχουν σε αυτά τα δίκτυα υπολογιστών είναι διάφορες και ασχολούνται με πολλές διαφορετικές δουλειές. Επιπλέον, ο κινητός εξοπλισμός εισχωρεί σε αυτά τα δίκτυα με αυξανόμενη ταχύτητα: οι υπάλληλοι πρέπει να επικοινωνούν με τις επιχειρήσεις τους, ενώ δεν είναι στο γραφείο τους. Κάνουν έτσι χρήση φορητών υπολογιστών, PDA ή κινητών τηλεφώνων (γενικώς την ασύρματη τεχνολογία), ώστε να αποκτήσουν πρόσβαση στα δεδομένα των επιχειρήσεών τους. Η πολυπλοκότητα των συστημάτων υπολογισμού έχει φτάσει στα όρια των ανθρωπίνων ικανοτήτων. Τα συστήματα γίνονται πάρα πολύ ογκώδη και σύνθετα για τους πιο εξειδικευμένους αρχιτέκτονες συστημάτων ακόμη και για να τα εγκαθιστούν, να τα διαμορφώνουν, να τα βελτιστοποιούν και να τα διατηρούν.

Την λύση στο πρόβλημα αυτό, μπορούν να δώσουν τα αυτόνομα συστήματα, όπως προτάθηκαν από την IBM το 2001. Τα αυτόνομα συστήματα είναι αυτό-διαχειριζόμενα συστήματα που παίρνουν αποφάσεις από μόνα τους, ελέγχουν και βελτιστοποιούν την κατάστασή τους συνεχώς και αυτόματα προσαρμόζονται σε ευμετάβλητες συνθήκες. Η ονομασία τους δεν είναι τυχαία, αλλά σκοπίμως παραπέμπει σε ορολογία της βιολογίας. Το αυτόνομο νευρικό σύστημα του ανθρώπου ελέγχει τον ρυθμό της καρδιάς και την θερμοκρασία του σώματος, έτσι ώστε να

απαλλάσσει τον «ενσυνείδητο» εγκέφαλο από αυτές και άλλες πολλές χαμηλού επιπέδου, αλλά παρόλα αυτά ζωτικής σημασίας, λειτουργίες.



Σχήμα 1.1: Πρόβλημα διάγνωσης στην αναβάθμιση ενός αυτόνομου συστήματος

Η αναβάθμιση εισάγει 5 μονάδες λογισμικού (μπλε), καθεμία είναι αυτόνομο στοιχείο. Λεπτά μετά την εγκατάσταση, βρίσκεται λανθασμένη έξοδος σε τρεις από τις μονάδες (κόκκινο) και το σύστημα επανέρχεται στην παλιά του κατάσταση. Ένα άλλο αυτόνομο διαγνωστικό στοιχείο, λαμβάνει τις πληροφορίες για τις εξαρτήσεις μεταξύ των οντοτήτων, από μια συσκευή ανάλυσης, η οποία ελέγχει το σύστημα κατά περιόδους. Η διαγνωστική οντότητα αναλύει τα log files και συμπεραίνει ποια από τις τρεις μονάδες είναι η ένοχη (κόκκινο Χ). Παράγει εν συνεχεία ένα δελτίο του προβλήματος που περιέχει διαγνωστικές πληροφορίες και το στέλνει σε κάποιον software developer που εκτελεί το debugging της ελαττωματικής μονάδας.

Η ουσία των αυτόνομων συστημάτων υπολογισμού είναι η αυτό-διαχειριστική τους ικανότητα, πρόθεση της οποίας είναι να ελευθερώσει τους administrators των συστημάτων από τις λεπτομέρειες της λειτουργίας και της συντήρησης τους, αλλά και για να παρέχει στους χρήστες ένα μηχάνημα που τρέχει στη μέγιστη απόδοση 24/7. Τα αυτόνομα συστήματα θα διατηρήσουν και θα ρυθμίσουν τη λειτουργία τους σε μεταβαλλόμενες συνθήκες ενημερώσεων υλικού και λογισμικού, καθώς και φόρτου εργασίας. Το αυτόνομο σύστημα καλείται να ελέγχει συνεχώς την ίδια του την χρήση. Όταν παρουσιαστούν αναβαθμίσεις, τότε το σύστημα θα τις εγκαταστήσει, θα ρυθμίσει κατάλληλα την λειτουργία του και θα τρέξει ένα τεστ οπισθοδρόμησης για να σιγουρευτεί ότι όλα είναι καλά. Όταν ανιχνεύει σφάλματα, το σύστημα θα επανέλθει στην προηγούμενη έκδοση του, ενώ οι *αλγόριθμοι αυτόματης ανεύρεσης*

προβλήματος, προσπαθούν να απομονώσουν την πηγή του σφάλματος. Το σχήμα 1.1 επεξηγεί την διαδικασία αυτή.

Η IBM αναφέρει τέσσερις πτυχές της έννοιας της αυτοδιαχείρισης (self-management), οι οποίες παρουσιάζονται και στον πίνακα 1.1 συνοπτικά. Αυτές οι πτυχές είναι προκύπτουσες ιδιότητες μιας γενικής αρχιτεκτονικής των αυτόνομων συστημάτων.

Self-configuration

Τα αυτόνομα συστήματα θα ρυθμίζονται αυτόματα σύμφωνα με υψηλού επιπέδου πολιτικές (high-level policies), οι οποίες προσδιορίζουν το επιδιωκόμενο αποτέλεσμα και όχι τον τρόπο επίτευξης του. Όταν ένα στοιχείο εισάγεται στο σύστημα, θα ενσωματωθεί ομαλά, και το υπόλοιπο σύστημα θα προσαρμοστεί στην ύπαρξή του, όπως ένα νέο κύτταρο στον ανθρώπινο οργανισμό. Παραδείγματος χάριν, όταν εισάγεται ένα νέο στοιχείο σε ένα αυτόνομο λογιστικό σύστημα, όπως στο σχήμα 1.1, αυτόματα θα μάθει τη σύνθεση και τη διαμόρφωση του συστήματος. Θα καταχωρήσει το ίδιο και τις ικανότητές του, έτσι ώστε άλλες οντότητες να μπορούν είτε να το χρησιμοποιήσουν, είτε να τροποποιήσουν τη συμπεριφορά τους κατάλληλα.

Έννοια	Σημερινή Τεχνολογία	Αυτόνομη Τεχνολογία
Self-configuration	Τα εταιρικά κέντρα δεδομένων διαθέτουν πολλούς προμηθευτές και πλατφόρμες. Η εγκατάσταση, η ρύθμιση και η ενσωμάτωση συστημάτων είναι χρονοβόρα και επιρρεπής σε λάθη.	Η αυτόματη ρύθμιση των στοιχείων και των συστημάτων ακολουθεί πολιτικές υψηλού επιπέδου. Το υπόλοιπο σύστημα προσαρμόζεται αυτόματα και ομαλά.
Self-optimization	Τα συστήματα έχουν εκατοντάδες μη γραμμικές παραμετροποιήσιμες μεταβλητές που τίθενται manually.	Οι οντότητες και τα συστήματα συνεχώς επιδιώκουν τρόπους να βελτιώσουν την απόδοση τους.
Self-healing	Ο καθορισμός προβλημάτων σε μεγάλα και πολύπλοκα συστήματα μπορεί να πάρει εβδομάδες.	Το σύστημα αυτόματα εντοπίζει, αναλύει και διορθώνει τοπικά προβλήματα υλικού και λογισμικού
Self-protection	Ο εντοπισμός και η ανάκτηση από επιθέσεις και απότομες βλάβες γίνεται με χειροκίνητο τρόπο.	Τα συστήματα αμύνονται αυτόματα σε κακόβουλες επιθέσεις και απότομες βλάβες.

Πίνακας 1.1 : Οι 4 πτυχές της έννοιας της αυτοδιαχείρισης

Self-optimization

Το σύνθετο middleware, όπως είναι το WebSphere, ή τα συστήματα βάσεων δεδομένων, όπως είναι ή Oracle ή η DB2, μπορεί να έχουν εκατοντάδες παραμετροποιήσιμες μεταβλητές που πρέπει να τεθούν σωστά, ώστε το σύστημα να λειτουργεί βέλτιστα. Όμως λίγοι άνθρωποι ξέρουν πώς να τις παραμετροποιήσουν. Τέτοια συστήματα είναι συχνά ενσωματωμένα σε άλλα, το ίδιο σύνθετα και πολύπλοκα, συστήματα. Συνεπώς, ο συντονισμός ενός μεγάλου υποσυστήματος μπορεί να έχει απρόβλεπτα αποτελέσματα σε ολόκληρο το σύστημα. Τα αυτόνομα συστήματα επιδιώκουν συνεχώς να βρίσκουν τους τρόπους, ώστε να βελτιώνουν τη λειτουργία τους και να επωφελούνται τις ευκαιρίες να καταστούν αποτελεσματικότερα σε απόδοση ή κόστος. Ακριβώς όπως οι μύες του σώματος γίνονται ισχυρότεροι μέσω της άσκησης.

Self-healing

Σοβαρά προβλήματα πελατών της IBM και άλλων IT (Information Technology) προμηθευτών μπορεί να πάρουν στις ομάδες των προγραμματιστών αρκετές εβδομάδες μέχρι να εντοπιστούν και να διορθωθούν, ενώ μερικές φορές το πρόβλημα εξαφανίζεται μυστηριωδώς, χωρίς οποιαδήποτε διάγνωση. Τα αυτόνομα συστήματα υπολογισμού θα ανιχνεύσουν, θα διαγνώσουν και θα διορθώσουν τοπικά προβλήματα που οφείλονται σε αποτυχίες λογισμικού και υλικού, μέσω ενός ελεγκτή οπισθοδρόμησης, όπως στο σχήμα 1.1. Ένας τέτοιος ελεγκτής αναλύει όλες τις πληροφορίες και τα δεδομένα από τα log files και έπειτα τεστάρει γνωστά software patches για την επίλυση του προβλήματος, εγκαθιστά το κατάλληλο patch και επανελέγχει το σύστημα. Σε διαφορετική περίπτωση ειδοποιεί κάποιον διαχειριστή του συστήματος.

Self-protection

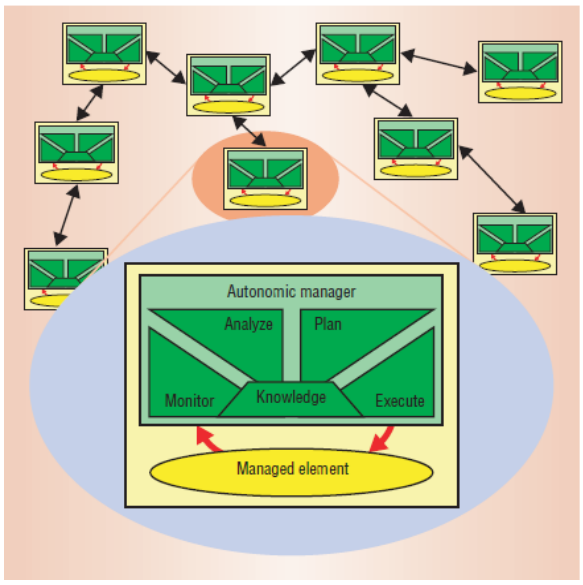
Παρά την ύπαρξη τοίχων προστασίας (firewalls) και των εργαλείων ελέγχου παρείσφρησης συστήματος, οι άνθρωποι είναι αυτοί που πρέπει να αποφασίσουν πώς να προστατεύσουν τα συστήματα από κακόβουλες επιθέσεις και απότομες βλάβες. Τα αυτόνομα συστήματα είναι αυτό-προστατευμένα υπό δύο έννοιες. Θα υπερασπίσουν το σύστημα συνολικά ενάντια στα μεγάλης κλίμακας προβλήματα που προκύπτουν

από τις κακόβουλες επιθέσεις ή τις αποτυχίες απότομης βλάβης και παραμένουν μη διορθωμένα από τα αυτό-θεραπευόμενα μέτρα. Επίσης θα προλαμβάνουν προβλήματα βασιζόμενα σε πρόωρες αναφορές από τους αισθητήρες.

Αρχιτεκτονική αυτόνομου συστήματος

Όπως φαίνεται στο σχήμα 1.2, ένα αυτόνομο στοιχείο αποτελείται τυπικά από ένα ή περισσότερα διοικούμενα στοιχεία, που συνδέονται με έναν ενιαίο αυτόνομο διαχειριστή που τα ελέγχει και τα αντιπροσωπεύει. Το διοικούμενο στοιχείο θα είναι ουσιαστικά ισοδύναμο με αυτό που βρίσκεται στα συνηθισμένα μη αυτόνομα συστήματα, αν και μπορεί να προσαρμοστεί, ώστε να επιτρέψει στον αυτόνομο διαχειριστή να το επιτηρήσει και να το ελέγξει.

Το διοικούμενο στοιχείο θα μπορούσε να είναι είτε κάποιος πόρος υλικού (hardware), όπως αποθηκευτικό μέσο, μια ΚΜΕ, ή ένας εκτυπωτής, είτε κάποιος πόρος λογισμικού (software), όπως μια βάση δεδομένων ή μια υπηρεσία καταλόγου αρχείων. Σε υψηλότερο επίπεδο, το διοικούμενο στοιχείο θα μπορούσε να είναι ένα ηλεκτρονικό βοήθημα, μια υπηρεσία εφαρμογής ή ακόμα και μια μεμονωμένη επιχείρηση. Ο αυτόνομος διαχειριστής διακρίνει το αυτόνομο στοιχείο από το αντίστοιχό μη αυτόνομο. Με τον έλεγχο του διοικούμενου στοιχείου και του εξωτερικού περιβάλλοντός του, καθώς και την κατασκευή και την εκτέλεση των σχεδίων βασισμένων στις αναλύσεις πληροφοριών, ο αυτόνομος διαχειριστής απαλλάσσει τον άνθρωπο από την επίπονη ευθύνη της διοίκησης του διοικούμενου στοιχείου.

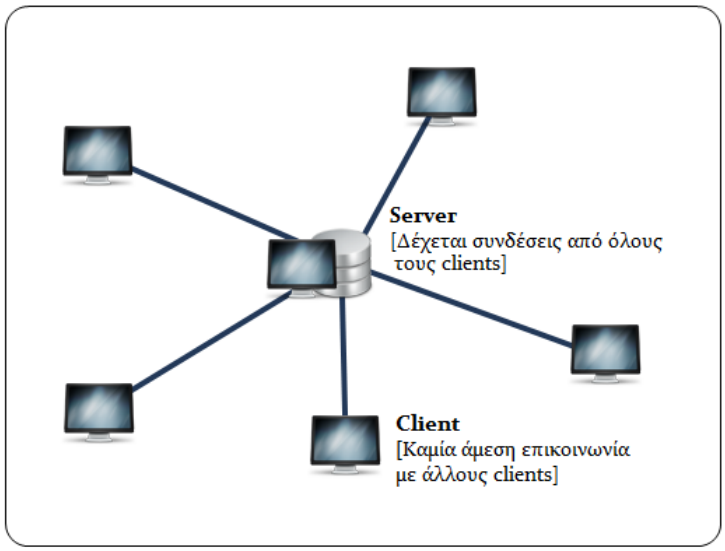


Σχήμα 1.2 : Δομή ενός αυτόνομου στοιχείου

1.3 Συγκεντρωμένα δίκτυα

Τα δίκτυα υπολογιστών μπορούν να ταξινομηθούν με βάση διάφορα κριτήρια, όπως το εύρος της περιοχής που καλύπτει το δίκτυο, την αρχιτεκτονική του δικτύου, την τοπολογία του δικτύου και το φυσικό μέσο διασύνδεσης. Θα επικεντρωθούμε στις δύο κατηγορίες αρχιτεκτονικής δικτύων υπολογιστών που εντοπίζονται και είναι η αρχιτεκτονική των συγκεντρωμένων δικτύων (centralized networks) και η αρχιτεκτονική των κατανεμημένων δικτύων (distributed networks).

Τα συγκεντρωμένα ή αλλιώς κεντροποιημένα δίκτυα αποτελούνται από τερματικούς κόμβους που εξυπηρετούνται από κάποιο μεγάλο υπολογιστικό «κέντρο». Τα πρώτα δίκτυα υπολογιστών που δημιουργήθηκαν από τον στρατό ήταν τέτοια δίκτυα, όπου όλοι οι υπολογιστικοί πόροι του συστήματος βρίσκονταν συγκεντρωμένοι σε ένα ή περισσότερα κεντρικά hubs. Τα δίκτυα αυτά στηρίζονται στο μοντέλο client-server. Ειδικότερα, ένα σύστημα client-server είναι ένα σύστημα στο οποίο το δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε οι clients (ή αλλιώς front end) να μπορούν να ζητούν υπηρεσίες από έναν server (ή αλλιώς back end), ο οποίος προσφέρει πληροφορίες ή επιπρόσθετη υπολογιστική ισχύ.



Σχήμα 1.3 : Συγκεντρωμένο δίκτυο

Με άλλα λόγια, στο client-server μοντέλο, ο client θέτει μια αίτηση και ο server επιστρέφει μια ανταπόκριση ή κάνει μια σειρά από ενέργειες. Στα συγκεντρωμένα δίκτυα δεν υπάρχει κατευθείαν επικοινωνία μεταξύ των κόμβων του συστήματος. Για

να αποκτήσει ένας κόμβος πρόσβαση στους πόρους ενός άλλου κόμβου, πρέπει να πάρει την έγκριση από τον κεντρικό κόμβο. Τα συγκεντρωμένα δίκτυα παρά τα πλεονεκτήματα που τα διακρίνουν, έχουν ένα σοβαρό μειονέκτημα. Εάν η κεντρική αρχή[1] υποστεί κάποια απότομη βλάβη ή δεχτεί επίθεση από κακόβουλους χρήστες, τότε ολόκληρο το σύστημα θα οδηγηθεί σε κατάρρευση.

Τα συγκεντρωμένα δίκτυα με την αυστηρή έννοια του όρου centralized system, σπάνια υφίστανται στις μέρες μας. Σαν παράδειγμα αναφέρουμε, τα δίκτυα που χρησιμοποιούν συχνά οι τράπεζες για την επίτευξη της επικοινωνίας μεταξύ υποκαταστημάτων και μηχανογραφικού κέντρου. Πλέον, στην εποχή που ζούμε, η πλειονότητα των δικτύων υπολογιστών είναι στηριγμένη στην κατανεμημένη αρχιτεκτονική. Όλοι οι κόμβοι είναι servers και clients μαζί και καμία κεντρική αρχή δεν υπάρχει στο σύστημα.

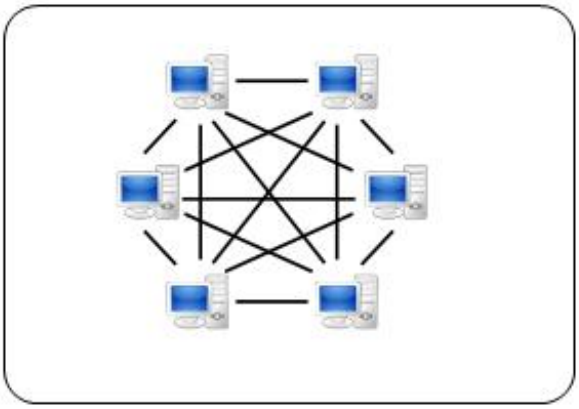
1.4 Κατανεμημένα δίκτυα

Όπως προδίδει και ο όρος, τα κατανεμημένα δίκτυα είναι εκείνα τα δίκτυα υπολογιστών στα οποία οι υπολογιστικές δυνατότητες είναι μοιρασμένες (κατανεμημένες) σε πολλά υπολογιστικά συστήματα. Η κατανεμημένη δικτυακή σχεδίαση διασκορπίζει τις υπολογιστικές και τις επικοινωνιακές δυνατότητες σε ολόκληρο το δίκτυο επιτρέποντας, στις δικτυακές συσκευές να επικοινωνούν απευθείας μεταξύ τους. Η απόφαση του να χρησιμοποιηθεί κατανεμημένη ή συγκεντρωτική δικτυακή σχεδίαση εξαρτάται σε μεγάλο βαθμό από τις γεωγραφικές, οικονομικές, λειτουργικές και σχετικές με την απόδοση εκτιμήσεις, οι οποίες αξιολογούνται ανάλογα με τις ανάγκες της εκάστοτε επιχείρησης. Ένα κατανεμημένο δίκτυο μπορεί να χρησιμοποιηθεί για τους εξής σκοπούς:

- Για τη μείωση της συμφόρησης σε υψηλής χρήσεως κέντρα με την κατανομή του φόρτου εργασίας.
- Για τη μείωση της επίδρασης που μπορεί να έχει ένα κεντρικό σημείο βλάβης στη λειτουργία ολόκληρου του δικτύου.
- Για την παροχή βελτιωμένων υπηρεσιών και λειτουργιών.

[1] Ο server στα συγκεντρωμένα συστήματα λέγεται κεντρική αρχή λόγω της απόλυτης εξουσίας που κατέχει

- Για τη διευκόλυνση σταδιακών αλλαγών στο δίκτυο, χωρίς το φόβο της πλήρους αποδιοργάνωσής του.



Σχήμα 1.4 : P2P δίκτυο

1.4.1 Ομότιμα δίκτυα P2P

Το πιο χαρακτηριστικό παράδειγμα κατανεμημένων συστημάτων, είναι τα ομότιμα δίκτυα (peer-to-peer), γνωστά και ως p2p. Ένα p2p δίκτυο είναι μία κατανεμημένη εφαρμογή που μοιράζει τις στοιχειώδεις εργασίες ή τον φόρτο εργασίας ισόνομα στους κόμβους που το αποτελούν. Οι κόμβοι αυτοί είναι εξίσου προνομιούχοι και ισοδύναμοι συμμετέχοντες στην εφαρμογή και γι' αυτόν τον λόγο ονομάζονται ομότιμοι κόμβοι. Το σύστημα p2p παρέχει τις υπηρεσίες για τις οποίες οι συμμετέχοντες μοιράζονται ένα μέρος των πόρων τους, όπως ισχύ επεξεργασίας, μνήμη δίσκου (disk storage), εύρος ζώνης δικτύου, ευχέρεια εκτύπωσης. Τέτοιοι πόροι παρέχονται άμεσα σε άλλους ομότιμους κόμβους χωρίς ενδιάμεσους εξυπηρετητές. Οι ομότιμοι κόμβοι είναι ταυτόχρονα πάροχοι και καταναλωτές των υπηρεσιών ή των πόρων. Σε αντιστοιχία με το μοντέλο client-server, ένας p2p κόμβος είναι ταυτόχρονα πελάτης, αλλά και εξυπηρετητής. Αυτό βέβαια έρχεται σε αντίθεση με το παραδοσιακό μοντέλο client-server, όπου οι πελάτες καταναλώνουν τους πόρους που παρέχονται μόνο από τους κεντρικούς υπολογιστές (servers). Τα κατανεμημένα p2p δίκτυα, τυπικά, συγκροτούνται με δυναμικό τρόπο με ad-hoc προσθήκες κόμβων.

Τα peer-to-peer συστήματα συνήθως υλοποιούν ένα επικαλυπτόμενο επίπεδο εφαρμογής δικτύου (*Application Layer overlay network*) πάνω από την φυσική

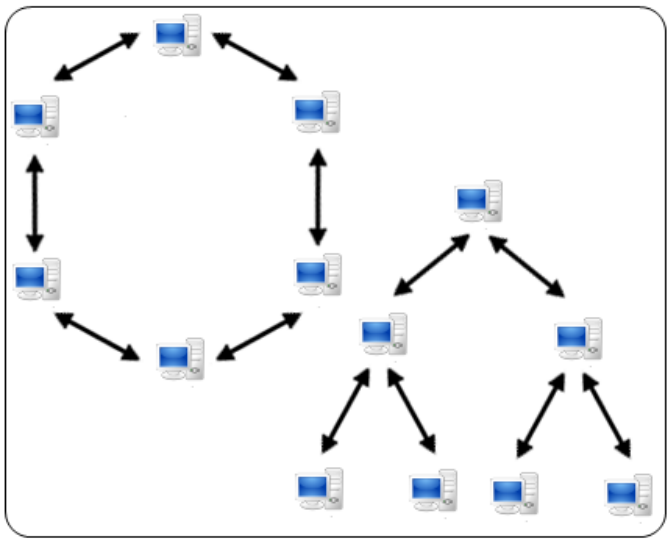
τοπολογία του δικτύου (φυσικό στρώμα). Τέτοια επικαλυπτόμενα δίκτυα χρησιμοποιούνται για την δημιουργία ευρετηρίου με τους κόμβους του δικτύου, για να εντοπίζουν τους κόμβους και για να διευκολύνουν την επικοινωνία μεταξύ των κόμβων. Οι επικαλύψεις χρησιμοποιούνται συνήθως μόνο για τη συντήρηση των συνδέσεων. Τα δεδομένα ανταλλάσσονται άμεσα στο ελλοχεύον IP δίκτυο. Μια αξιοσημείωτη εξαίρεση είναι συστήματα που απαιτούν την ανωνυμία των συμμετεχόντων τους.

Με βάση την λογική αρχιτεκτονική του δικτύου, τα p2p μπορούν να τοποθετηθούν σε διάφορες κατηγορίες. Κάποιες από αυτές, υπάρχουν μόνο θεωρητικά στις μέρες μας, καθώς η υλοποίηση τους είτε είναι ανέφικτη με την σημερινή τεχνολογία, είτε είναι μη πρακτικές από άποψη απόδοσης. Στη συνέχεια παρουσιάζουμε τις αρχιτεκτονικές p2p δικτύων που έχουν «υιοθετηθεί» και αναφέρουμε ως παραδείγματα τις πιο γνωστές τους εφαρμογές. Το επικαλυπτόμενο δίκτυο p2p αποτελείται από όλους τους συμμετέχοντες ομότιμους χρήστες, που παρουσιάζονται ως δικτυακοί κόμβοι. Υπάρχουν αμφίδρομες κατευθυνόμενες συνδέσεις μεταξύ δύο τυχαίων κόμβων που ξέρουν ο ένας τον άλλον. Με βάση τον τρόπο με τον οποίο οι κόμβοι συνδέονται στο επικαλυπτόμενο δίκτυο μπορούμε να διακρίνουμε τα δομημένα p2p δίκτυα (structured) και τα αδόμητα p2p δίκτυα (unstructured).

Δομημένα P2P δίκτυα

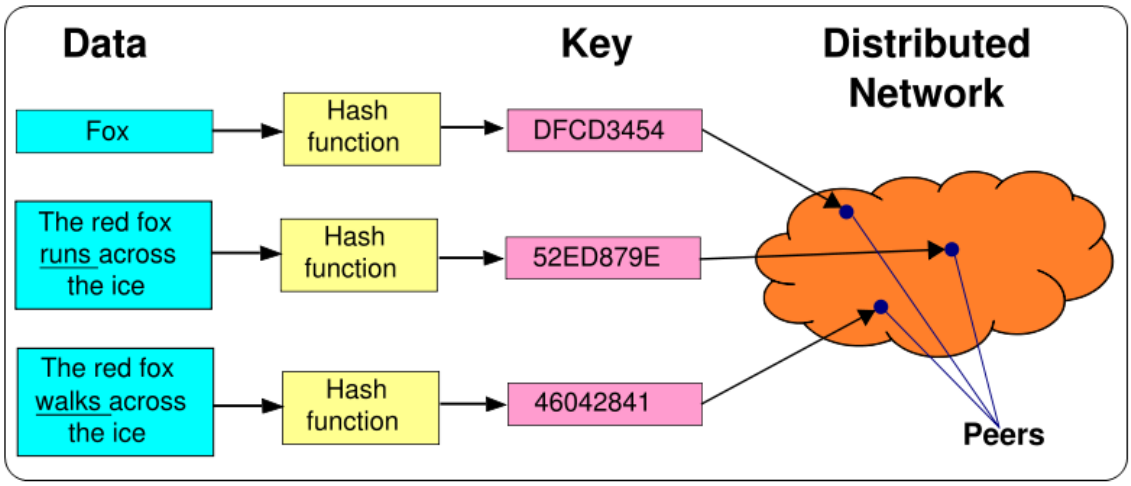
Στα δομημένα p2p δίκτυα οι συνδέσεις στο επικαλυπτόμενο δίκτυο (overlay) είναι σταθερές. Χρησιμοποιείται ένα καθολικό και σταθερό πρωτόκολλο για να εξασφαλίσει ότι οποιοσδήποτε κόμβος μπορεί αποτελεσματικά να κάνει αναζήτηση ενός κόμβου που έχει το επιθυμητό αρχείο, ακόμα κι αν το αρχείο είναι εξαιρετικά σπάνιο. Το πιο κοινό σύστημα που προσφέρει αυτή την δυνατότητα είναι ο κατανεμημένος πίνακας κατακερματισμού (DHT), στον οποίο γίνεται χρήση μιας παραλλαγής του consistent hashing[2] ώστε να αναθέσει την ιδιοκτησία κάθε αρχείου σε έναν συγκεκριμένο κόμβο, με τρόπο ανάλογο του παραδοσιακού πίνακα hash, όπου γινόταν ανάθεση μιας λέξης-κλειδί σε μια συγκεκριμένη θέση στον πίνακα.

[2] Περισσότερες πληροφορίες στην ιστοσελίδα : http://en.wikipedia.org/wiki/Consistent_hashing



Σχήμα 1.5 : Δομημένα δίκτυα P2P

Οι DHT παρέχουν την δυνατότητα σε οποιοδήποτε ομότιμο κόμβο να μπορεί αποτελεσματικά να ανακτήσει την τιμή που συνδέεται με ένα δεδομένο κλειδί. Επιπλέον, επιτρέπουν την κλιμάκωση του συστήματος σε εξαιρετικά μεγάλο αριθμό κόμβων και έχουν την ικανότητα να διαχειρίζονται αποτελεσματικά συνεχείς αφίξεις, αναχωρήσεις και βλάβες ομότιμων κόμβων.



Σχήμα 1.6 : Κατανεμημένοι Πίνακες Κατακερματισμού

Δύο συστήματα που υιοθετούν την δομημένη p2p αρχιτεκτονική είναι τα Chord και Pastry. Και στις δύο περιπτώσεις χρησιμοποιείται τοπολογία δαχτυλιδιού και μηχανισμοί βασισμένοι στα μαθηματικά εξασφαλίζουν ότι όλοι οι κόμβοι μέσα στο δίκτυο μπορούν να εντοπιστούν σε ορισμένο αριθμό βημάτων πολυπλοκότητας

$O(\log n)$. Εντούτοις, και τα δύο δίκτυα αυτά είναι ερευνητικά συστήματα. Ακόμη η ιδέα των δομημένων p2p συστημάτων δεν έχει προχωρήσει σε εμπορικό επίπεδο.

Αδόμητα P2P δίκτυα

Ένα αδόμητο p2p δίκτυο διαμορφώνεται όταν οι επικαλυπτόμενες συνδέσεις εγκαθίστανται αυθαίρετα. Τέτοια δίκτυα μπορούν να κατασκευαστούν εύκολα, δεδομένου ότι ένας νέος κόμβος που θέλει να συνδεθεί στο δίκτυο μπορεί να αντιγράψει τις υπάρχουσες συνδέσεις ενός άλλου κόμβου και να διαμορφώσει έπειτα τις συνδέσεις του με το πέρασμα του χρόνου. Σε ένα αδόμητο p2p δίκτυο, εάν ένας κόμβος θέλει να βρει ένα δεδομένο, το ερώτημα θα διαδοθεί στο δίκτυο με την μορφή «πλημμύρας», ώστε να βρεθούν όσο γίνεται δυνατόν περισσότεροι κόμβοι που μοιράζονται το συγκεκριμένο δεδομένο. Το βασικό μειονέκτημα σε τέτοια δίκτυα είναι ότι τα queries αυτά δεν επιλύονται πάντα. Αυτό οφείλεται, ουσιαστικά, στο γεγονός ότι μέσα σε ένα μεγάλο σύστημα, το δίκτυο θα μπορούσε να «πλημμυρίσει» με μεγάλο αριθμό από queries με αποτέλεσμα να υπάρχει συνωστισμός και χαμηλή απόδοση αναζήτησης.

Επιπροσθέτως, δεδομένου ότι δεν υπάρχει κανένας συσχετισμός μεταξύ ενός κόμβου και των δεδομένων που μοιράζεται, δεν υπάρχει καμία εγγύηση ότι η «πλημμύρα» θα βρει έναν κόμβο με το ζητούμενο δεδομένο. Τα αδόμητα p2p δίκτυα χωρίζονται σε τρία είδη. Τα αμιγή (pure), τα συγκεντρωμένα και τα υβριδικά p2p δίκτυα.

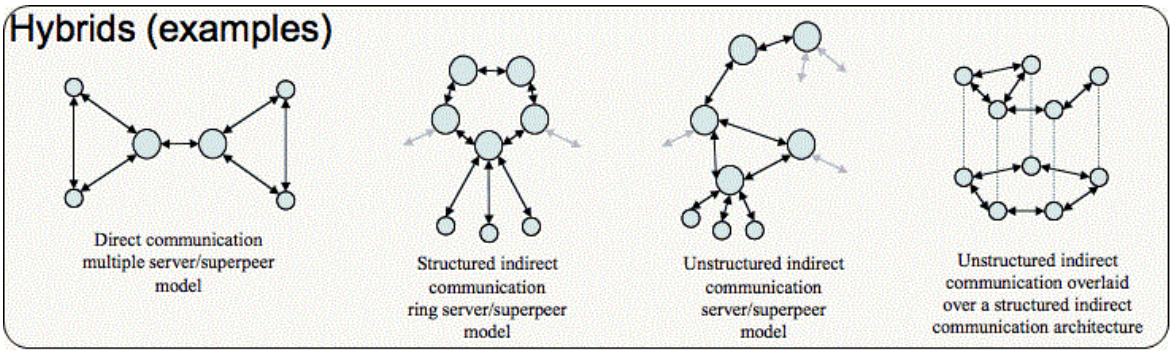
- **Αμιγή p2p:** Κάθε κόμβος που συμμετέχει στο δίκτυο είναι ταυτόχρονα και client και server. Μόλις συνδεθεί με κάποιο p2p client πρόγραμμα κάνει γνωστή την παρουσία του σε ένα μικρό αριθμό υπολογιστών ήδη συνδεδεμένων, οι οποίοι με τη σειρά τους προωθούν τη δήλωση παρουσίας του σε ένα μεγαλύτερο δίκτυο υπολογιστών. Πλέον ο χρήστης έχει τη δυνατότητα να αναζητήσει οποιαδήποτε πληροφορία μεταξύ των διαμοιραζόμενων αρχείων. Οι πιο γνωστές εφαρμογές αμιγών δικτύων p2p είναι το Gnutella και το FreeNet. Οι πιο πρόσφατες εκδόσεις του Gnutella κάνουν χρήση των super-peers που ενεργούν ως hubs στο δίκτυο.

- **Συγκεντρωμένα p2p:** Στα δίκτυα αυτά υπάρχει ένας κεντρικός Index Server στον οποίο αποθηκεύονται οι πληροφορίες για τα περιεχόμενα των καταλόγων, που οι συμμετέχοντες επιθυμούν να μοιράζονται. Οι χρήστες μπορούν να αναζητήσουν

στους Index Servers αυτούς τα αρχεία που ψάχνουν, χρησιμοποιώντας ένα κατάλληλο client πρόγραμμα. Όταν το αρχείο βρεθεί, ανοίγει μια σύνδεση μεταξύ των δύο χρηστών για τη μεταφορά του. Οι πιο γνωστές εφαρμογές συγκεντρωμένων δικτύων p2p είναι το Napster το DC++ και το WinMX.

- **<u>Υβριδικά p2p:</u>** Τα τελευταία χρόνια έχει υπάρξει κινητοποίηση γύρω από τις υβριδικές αρχιτεκτονικές που βασίζονται στη χρήση πολλαπλών τοπολογιών ταυτόχρονα. Με το συνδυασμό των τοπολογιών είναι συχνά δυνατό να αντιμετωπιστούν τα μειονεκτήματα μιας απλής τοπολογίας. Παραδείγματος χάριν, με τη χρησιμοποίηση πολλών κόμβων ευρετηρίου, που συνδέονται με έναν αποκεντρωμένο τρόπο, η πιθανότητα βλάβης σε κεντρικό σημείο απαλείφεται ή με τη χρησιμοποίηση μιας δομημένης τοπολογίας πάνω από μία αδόμητη τοπολογία όλοι οι μηχανισμοί αναζήτησης πόρων μπορούν να χρησιμοποιηθούν ταυτόχρονα με την δυνατότητα εντοπισμού όλων των κόμβων του δικτύου.

Στην κατηγορία των υβριδικών δικτύων ανήκουν και τα δίκτυα ηλεκτρονικού εμπορίου με χρήση δημοπρασίας (e-commerce auctioning), όπως είναι το e-Bay. Στο σχήμα που ακολουθεί φαίνονται μερικά παραδείγματα υβριδικών δικτύων.



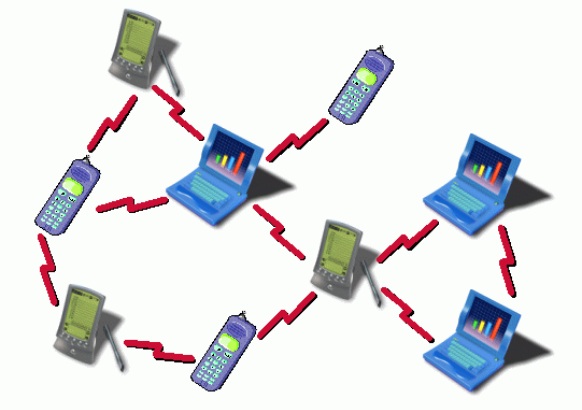
Σχήμα 1.7 : Υβριδικά P2P δίκτυα

1.4.2 Ad-hoc δίκτυα

Ένα ad-hoc δίκτυο είναι μια συλλογή αυτόνομων κόμβων που δεν στηρίζονται σε μια προκαθορισμένη δομή για να κρατάει το δίκτυο σε συνοχή. Οι κόμβοι επικοινωνούν μεταξύ τους χρησιμοποιώντας ασύρματη επικοινωνία και λειτουργούν ακολουθώντας ένα μοντέλο ομότιμων οντοτήτων (p2p). Σε ένα ad-hoc δίκτυο οι κόμβοι προσφέρονται να προωθήσουν δεδομένα σε άλλους κόμβους και επομένως η απόφαση ποιοι κόμβοι θα δρομολογούν δεδομένα γίνεται δυναμικά ανάλογα με τις διασυνδέσεις του δικτύου. Αυτό έρχεται σε αντίθεση με τα μη ασύρματα δίκτυα στα οποία δρομολογητές αναλαμβάνουν το θέμα της δρομολόγησης.

Η αποκεντρωμένη φύση των ad-hoc δικτύων τα κάνει κατάλληλα για ποικίλες εφαρμογές όπου κεντρικές αρχές (κόμβοι) δεν είναι αξιόπιστες. Οι μινιμαλιστικές απαιτήσεις στη διαμόρφωση ενός δικτύου και η γρήγορη και εύκολη επέκταση των ad-hoc συστημάτων τα καθιστούν τα καταλληλότερα σε περιπτώσεις εκτάκτου ανάγκης, όπως φυσικές καταστροφές ή στρατιωτικές συγκρούσεις. Τα πιο γνωστά ad-hoc δίκτυα είναι τα mobile ad-hoc networks γνωστά και ως MANET.

Τα δίκτυα ad-hoc, διαμορφώνονται δυναμικά μεταξύ μιας ομάδας ασύρματων χρηστών και δεν απαιτούν καμία υπάρχουσα υποδομή ή προ-διαμόρφωση, όπως φαίνεται στο Σχήμα 1.8. Κάθε κόμβος λειτουργεί ως ένας ανεξάρτητος δρομολογητής λόγω της έλλειψης κεντρικής διαχείρισης, γεγονός που σημαίνει ότι δεν διαμοιράζεται καμία πληροφορία μεταξύ των κόμβων χωρίς να είναι εν γνώσει.



Σχήμα 1.8 : Ad-hoc δίκτυο

Η ασφάλεια στα ασύρματα δίκτυα και κυρίως στα ad-hoc δίκτυα, είναι ένα πολύ σημαντικό κομμάτι της ευρωστίας των δικτύων και της λειτουργίας τους. Ένα δίκτυο, εκτός από τις λειτουργίες της δημιουργίας και μετάδοσης μηνυμάτων, για να μπορέσει να εκτελέσει την αποστολή του ομαλά, πρέπει να μπορεί να είναι ασφαλές κατά την διάρκεια της λειτουργίας του. Και όταν λέμε ασφαλές, πρέπει να μπορεί κάθε κομμάτι του δικτύου να στέλνει και να δέχεται ασφαλή μηνύματα. Η ασφάλεια στα ad-hoc δίκτυα είναι δύσκολο να επιτευχθεί, λόγω της αδυναμίας των ασύρματων ζεύξεων, της περιορισμένης φυσικής προστασίας των κόμβων του δικτύου, της δυναμικά μεταβαλλόμενης τοπολογίας, της έλλειψης μιας αρχής πιστοποίησης και της έλλειψης ενός κεντρικού σημείου ελέγχου και διαχείρισης. Για να μπορέσουμε να εξασφαλίσουμε ότι ένα δίκτυο είναι ασφαλές, πρέπει να διασφαλίσουμε ότι το κάθε κομμάτι που αποτελεί το δίκτυο είναι ασφαλές. Όπως αναφέραμε και προηγουμένως τα ad-hoc δίκτυα αποτελούνται από ένα μεγάλο αριθμό κόμβων. Η ασφάλεια του δικτύου επιβάλλει να είναι κάθε κόμβος ασφαλής ώστε να μπορεί να ανταλλάσει μηνύματα με τους γειτονικούς κόμβους και με τον σταθμό βάσης.

2 Η εμπιστοσύνη σε P2P δίκτυα

2.1 Γενικά

Η δυνατότητα της συνεχούς επικοινωνίας μέσω του διαδικτύου οδηγεί σταδιακά στη μετανάστευση του εμπορίου και των επιχειρήσεων, από τις άμεσες αλληλεπιδράσεις μεταξύ των ανθρώπων, στις έμμεσες ηλεκτρονικές αλληλεπιδράσεις. Το διαδίκτυο μεταβαίνει από το κεντροποιημένο μοντέλο των κεντρικών οργάνων και μεσαζόντων, σε ένα δίκτυο με ομότιμους και ισόνομους χρήστες, όπως είναι το p2p δίκτυο. Οι ιδιότητες που χαρακτηρίζουν τα p2p δίκτυα είναι οι εξής :

- Είναι περιβάλλοντα ανοιχτού τύπου, με την έννοια ότι οι χρήστες μπορούν να συνδεθούν ή να αποσυνδεθούν από αυτά, όποτε το θελήσουν. Το γεγονός αυτό δημιουργεί στους χρήστες ένα ισχυρό αίσθημα αυτονομίας και ανεξαρτησίας που μπορεί να οδηγήσει σε διάφορες παρεκτροπές. Επιπλέον, η έλλειψη προσωπικής επικοινωνίας ενισχύει τις αρνητικές επιπτώσεις.

- Είναι αποκεντρωμένα περιβάλλοντα, χωρίς κεντρικά σημεία αποτυχίας του συστήματος. Ειδικότερα, είναι απελευθερωμένα από *Trust Third Parties* που θα επιτηρούσαν τις συναλλαγές μεταξύ των χρηστών και θα τιμωρούσαν ή θα απέκλειαν οποιαδήποτε παράνομη συμπεριφορά.

- Είναι παγκόσμια περιβάλλοντα, υπονοώντας ότι οι καθιερωμένοι μηχανισμοί δικαιοσύνης, όπως οι δικαστικοί αγώνες, είναι αναποτελεσματικοί εξαιτίας του μεγάλου κόστους συναλλαγής όταν ξεπεραστούν τα σύνορα δικαιοδοσίας.

Χωρίς να χρειάζεται να επεκταθούμε περισσότερο για την δομή των p2p δικτύων, είναι φανερή η ανάγκη διαχείρισης της εμπιστοσύνης σε τέτοια συστήματα. Για να γίνει πλήρως αποδεκτή η έννοια του ηλεκτρονικού εμπορίου στα p2p δίκτυα πρέπει

να εξαλειφθούν ή τουλάχιστον να ελαχιστοποιηθούν οι κίνδυνοι και οι απειλές από κακόβουλες επιθέσεις.

Η διαχείριση της εμπιστοσύνης σε αποκεντρωμένα συστήματα, όπως τα δίκτυα p2p, είναι ένα πρόβλημα ιδιαίτερης σπουδαιότητας, καθώς έχει να κάνει με την συνύπαρξη πολλών διαφορετικών και ξένων, μεταξύ τους, χρηστών. Άλλωστε, είναι χαρακτηριστικό γνώρισμα τέτοιων κοινοτήτων, ότι σχηματίζονται δυναμικά με μέλη άγνωστα μεταξύ τους. Κατά τη διάρκεια των τελευταίων ετών, κυρίως λόγω της άφιξης νέων δυνατοτήτων για ηλεκτρονική εργασία, οι άνθρωποι άρχισαν να αναγνωρίζουν τη σημασία της διαχείρισης εμπιστοσύνης στις ηλεκτρονικές κοινότητες.

Οι επισκέπτες της ιστοσελίδας Amazon, συνήθως ψάχνουν τις αναθεωρήσεις πελατών προτού αποφασίσουν να αγοράσουν νέα βιβλία. Οι συμμετέχοντες στις δημοπρασίες του eBay μπορούν να αξιολογήσουν ο ένας τον άλλον μετά από κάθε συναλλαγή. Τέλος, οι χρήστες της κοινότητας Gnutella ελέγχουν την αξιολόγηση των αρχείων ή και των ίδιων των χρηστών που τα έχουν ανεβάσει προτού αρχίσουν την μεταφόρτωσή τους. Παρατηρούμε πως η αντιμετώπιση του προβλήματος της εμπιστοσύνης στηρίζεται στην ανάπτυξη στρατηγικών και μοντέλων, που αξιολογούν το βαθμό εμπιστοσύνης κάθε μέλους της κοινότητας. Η *διαχείριση εμπιστοσύνης βασισμένη στην φήμη* (reputation-based trust management) αποτέλεσε την βιώσιμη λύση στο πρόβλημα αυτό.

Οι υπάρχουσες μέθοδοι για τη διαχείριση εμπιστοσύνης, που είναι βασισμένες στη φήμη, εστιάζουν στις σημασιολογικές ιδιότητες του μοντέλου εμπιστοσύνης και είτε στηρίζονται σε έναν κεντρικό κόμβο που λειτουργεί ως βάση δεδομένων, είτε απαιτούν από τον κάθε κόμβο να διατηρεί και να παρέχει στοιχεία όσον αφορά τις προηγούμενες αλληλεπιδράσεις. Τα στοιχεία αυτά, γνωστά και ως feedback, καθορίζουν την φήμη κάθε μέλους της κοινότητας. Παρόλα αυτά, η διαχείριση της εμπιστοσύνης βασισμένη στην φήμη, έχει καθοδηγητικό και συμβουλευτικό χαρακτήρα και όχι προστακτικό. Αυτό έρχεται σε αντίθεση με τα «συγκεντρωτικά» Trusted Third Parties, όπου κεντρικές αρχές επιβάλλουν την εμπιστοσύνη.

Όπως στα περισσότερα συστήματα διαχείρισης δικτύων, κάθε διαχειριζόμενος κόμβος διαθέτει τον αντιπρόσωπο του, τον οποίο από εδώ και στο εξής θα καλούμε πράκτορα (agent). Ένας πράκτορας είναι ουσιαστικά ένα πρόγραμμα το οποίο τρέχει στον διαχειριζόμενο κόμβο και επιτελεί αυτόματα τις λειτουργίες διαχείρισης.

2.2 Εμπιστοσύνη βασισμένη στην φήμη

Δεδομένου ότι είναι πέρα από τις δυνάμεις κάθε ατόμου να αξιολογήσει όλες τις πτυχές μιας δεδομένης κατάστασης, κατά τη λήψη μιας απόφασης εμπιστοσύνης, οι πράκτορες πρέπει να στηριχθούν σε άλλες πηγές πληροφοριών. Πράγματι, εάν η πλήρης γνώση ήταν εφικτή, η έννοια της εμπιστοσύνης θα ήταν αχρείαστη. Στην κοινωνία, λαμβάνουμε τις πληροφορίες από αυτές τις άλλες πηγές, με την διάδοση από στόμα σε στόμα, δηλαδή έναν μηχανισμό διάδοσης της φήμης. Αυτός ο μηχανισμός είναι επίσης μια μορφή κοινωνικού ελέγχου, όπου η συμπεριφορά ενός πράκτορα σε ένα τέτοιο σύστημα επηρεάζεται από άλλους πράκτορες που ενεργούν συνεταιριστικά. Παραδείγματος χάριν, ένα ανέντιμο παντοπωλείο (ιδιοκτήτης) θα αποκτήσει γρήγορα κακή φήμη στην γύρω γειτονιά και θα αναγκαστεί μακροπρόθεσμα να κλείσει το κατάστημα ή να βελτιώσει τη φήμη του. Επιπλέον, μια καλή φήμη λογίζεται ως πλεονέκτημα, καθώς η φήμη είναι και μια μορφή κοινωνικού κεφαλαίου, ειδικά στο εμπόριο. Κατά συνέπεια, οι πληροφορίες φήμης είναι σημαντικές στη λήψη των αποτελεσματικών και ενημερωμένων αποφάσεων εμπιστοσύνης. Από εδώ και στο εξής, κάθε φορά που θα αναφέρουμε την λέξη φήμη για p2p συστήματα θα εννοούμε τον παρακάτω ορισμό :

> Φήμη είναι η προσδοκία για την συμπεριφορά ενός πράκτορα, βασισμένη στις πληροφορίες που υπάρχουν γι' αυτόν ή στις παρατηρήσεις από την παρελθοντική συμπεριφορά του.

Η reputational πληροφορία μπορεί να είναι η γνώμη που προέρχεται από άλλους πράκτορες για έναν συγκεκριμένο πράκτορα ή να βασίζεται στην προσωπική εμπειρία που είχε ένας πράκτορας με αυτόν. Αυτό σημαίνει επί της ουσίας ότι η φήμη ενός πράκτορα είναι ο συνυπολογισμός των γνωμών άλλων πρακτόρων και της δικής μας προσωπικής γνώμης.

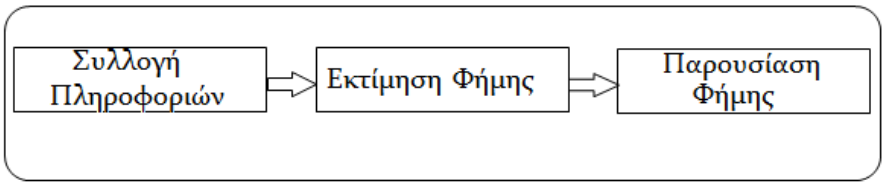
Τα συστήματα φήμης προσφέρουν μια βιώσιμη λύση στην ανάγκη αξιόπιστης συμπεριφοράς στα δίκτυα p2p. Οι βασικές προϋποθέσεις τους είναι ότι οι χρήστες μιας online κοινότητας δεσμεύονται να συμμετέχουν σε επαναλαμβανόμενες αλληλεπιδράσεις και ότι οι πληροφορίες που συγκεντρώνονται από τις προηγούμενες αλληλεπιδράσεις τους είναι ενδεικτικές της μελλοντικής απόδοσής τους και υπό αυτήν την έννοια θα την επηρεάσουν. Κατά συνέπεια, η συλλογή, η επεξεργασία και η διάδοση της reputational πληροφορίας των πρακτόρων έπειτα από κάθε αλληλεπίδραση τους, διαμορφώνουν την αξιοπιστία τους. Αρνητική reputational πληροφορία μειώνει την αξιοπιστία και θετική reputational πληροφορία την αυξάνει.
Τα κύρια στοιχεία που απαρτίζουν ένα μοντέλο εμπιστοσύνης βασισμένο στην φήμη είναι τα ακόλουθα :

1. **Trustee** → Η οντότητα που της δίνεται μια τιμή φήμης για μια υπηρεσία που παρέχει.
2. **Trustor** → Ο ομότιμος κόμβος που πρέπει να εκτιμήσει την φήμη του trustee προκειμένου να πάρει μια απόφαση εμπιστοσύνης σχετικά με αυτόν. Παραδείγματος χάριν, να ξεκινήσει κάποια συναλλαγή μαζί του.
3. **Μάρτυρας** → Ένας κόμβος που παρέχει μία σύσταση για έναν trustee, σύμφωνα με τις προσωπικές τους εμπειρίες με τον τελευταίο.
4. **Πλαίσιο** → Η φήμη ενός κόμβου εξαρτάται από το συγκεκριμένο πλαίσιο για το οποίο αναφέρεται η φήμη του, όπως ειδικές υπηρεσίες που προσφέρει ο trustee, ιδιότητες μια υπηρεσίας κλπ.
5. **Συστάσεις** → Feedback που παρέχουν κόμβοι σχετικό με την αξιοπιστία άλλων κόμβων.
6. **Εμπιστοσύνη ή Φήμη** → Δείκτης της ποιότητας των υπηρεσιών ενός trustee, που βασίζεται στις συστάσεις.

<u>Σχεδίαση p2p συστημάτων φήμης</u>

Γενικά, ένα σύστημα εμπιστοσύνης βασισμένο στην φήμη βοηθά τους κόμβους στην επιλογή ενός αξιόπιστου κόμβου για να πραγματοποιήσουν συναλλαγές. Για να παρέχει αυτήν την λειτουργία, ένα p2p σύστημα φήμης θα πρέπει να :

- Συλλέγει πληροφορίες για την συμπεριφορά στις συναλλαγές κάθε κόμβου. Οι οντότητες που συμμετέχουν στις συναλλαγές κρίνουν η μία την απόδοση της άλλης με τιμές αξιολόγησης, οι οποίες αθροίζονται τοπικά για να διαμορφώσουν την γνώμη μιας οντότητας για άλλες οντότητες. Οι μεμονωμένες αξιολογήσεις ή γνώμες αποτελούν τις συστάσεις, οι οποίες διανέμονται στο p2p δίκτυο. Κάθε κόμβος μπορεί να αποθηκεύσει τέτοιες πληροφορίες, αλλά και να τις προμηθεύσει ύστερα από αίτηση άλλου κόμβου.
- Αθροίζει τις πληροφορίες εμπιστοσύνης που αφορούν τη συμπεριφορά στις συναλλαγές ενός trustee και να παράγει μια τιμή εμπιστοσύνης (ή φήμης) για αυτόν. Δεδομένου ότι είναι αδύνατο ή πάρα πολύ δαπανηρό να ληφθούν οι αξιολογήσεις ή οι γνώμες όλων των αλληλεπιδράσεων με έναν συγκεκριμένο κόμβο, το σκορ της φήμης βασίζεται σε ένα υποσύνολο των αξιολογήσεων.
- Ταξινομεί τους κόμβους σύμφωνα με την εμπιστοσύνη τους ή να συγκρίνει την τιμή εμπιστοσύνης ενός κόμβου με μια τιμή ευαισθησίας προκειμένου να επιτραπεί ή όχι στον trustor η συναλλαγή με έναν επιλεγμένος κόμβο.



Σχήμα 2.1 : P2P reputation-based σύστημα εμπιστοσύνης

Η λειτουργία των p2p συστημάτων φήμης μπορεί να «σπάσει» σε τρία επιμέρους κομμάτια, όπως διακρίνεται και στο σχήμα 2.1. Κατά την κατασκευή κάθε κομματιού θα πρέπει να ληφθούν αποφάσεις σχετικές με σχεδιαστικά ζητήματα τα οποία παρουσιάζονται συνοπτικά στον πίνακα 2.1.

Πολλά μοντέλα που είναι βασισμένα στην φήμη έχουν προταθεί τα τελευταία χρόνια. Ο πρώτος που έκανε μια τέτοια κίνηση ήταν ο Marsh. Σε αυτόν στηρίχθηκε και το επόμενο μοντέλο εμπιστοσύνης που παρουσιάστηκε από τους Alfarez Abdul-Rahman και Stephen Hailes. Στη συνέχεια θα σας παρουσιάσουμε τα μοντέλα EigenTrust, ROCQ και Bayesian που αφορούν καθαρά p2p δίκτυα και το eBay που είναι μοντέλο εμπιστοσύνης για p2p δίκτυα ηλεκτρονικού εμπορίου. Το EigenTrust, το ROCQ και το Bayesian υλοποιούνται σε δομημένα p2p δίκτυα, όπως file sharing εφαρμογές.

Πίνακας 2.1 : Σχεδιαστικές εκτιμήσεις για την σχεδίαση ενός p2p συστήματος φήμης

Συλλογή Πληροφοριών	***Εκτίμηση Φήμης***	***Παρουσίαση Φήμης***
1.Αποθήκευση πληροφορίας εμπιστοσύνης, μηχανισμοί διάδοσης και αναζήτησης.	1. Αρχικοποίηση της τιμής εμπιστοσύνης (τι τιμή πρέπει να δοθεί σε έναν καινούριο κόμβο;).	1. Πεδίο τιμών της εμπιστοσύνης (συνεχείς ή διακριτές τιμές και πόσο μεγάλο σύνολο τιμών).
2.Τοπικός έλεγχος της αποθηκευμένης πληροφορίας εμπιστοσύνης (δυνατότητα επεξεργασίας της πληροφορίας από κόμβους στους οποίους είναι αποθηκευμένη η πληροφορία;).	2. Πεδίο δράσης της πληροφορίας εμπιστοσύνης (καθολική ή τοπική πληροφορία οδηγούν σε αντικειμενική ή υποκειμενική εμπιστοσύνη αντίστοιχα).	2. Ταξινόμηση ή τιμή ευαισθησίας.
3.Αξιοπιστία του κόμβου που δίνει συστάσεις.	3.Μέθοδος υπολογισμού της εμπιστοσύνης (απλές στατιστικές συναρτήσεις, πιθανολογικές μέθοδοι, fuzzy logic κλπ).	3. Δυσπιστία -distrust- (εκτός από την τιμή φήμης, θα υπάρχει και τιμή δυσπιστίας;).
4.Τύπος συμπεριφοράς που λογίζεται για την τιμή φήμης του trustee (θετική, αρνητική ή και τα δύο είδη συμπεριφοράς;).	4. Μεταβατικότητα (αν ο Α εμπιστεύεται τον Β, και ο Β εμπιστεύεται τον Γ, τότε ο Γ εμπιστεύεται τον Α;)	
5.Η εξάρτηση των διαφόρων πλαισίων για την αποτίμηση της φήμης (η φήμη είναι ίδια για όλες τις συναλλαγές ή αξιολογείται ανάλογα το πλαίσιο π.χ ποιότητα ή ταχύτητα;).	5. Πρόσφατα Δεδομένα (οι τελευταίες συναλλαγές θα έχουν μεγαλύτερη επίδραση στην φήμη ενός κόμβου απ' ότι οι πιο παλιές;).	

2.2.1 Μοντέλο EigenTrust

Η ανωνυμία και η ανοιχτή φύση των δικτύων p2p προσφέρουν ένα ιδανικό περιβάλλον για την διάδοση αυτό-αντιγραφούμενων μη αυθεντικών αρχείων. Ο αλγόριθμος EigenTrust έχει ως στόχο την μείωση του αριθμού των "μεταφορτώσεων" μη γνήσιων αρχείων σε ένα σύστημα «διαμοιρασμού-αρχείων».
Το EigenTrust είναι ένα πρωτόκολλο για την διαχείριση της φήμης σε ένα peer-to-peer σύστημα, αναθέτοντας σε κάθε ομότιμο κόμβο μια μοναδική καθολική τιμή εμπιστοσύνης (trust value), η οποία βασίζεται σε προηγούμενες συναλλαγές του κόμβου. Ο υπολογισμός της καθολικής τιμής εμπιστοσύνης κάθε κόμβου, γίνεται με μια ασφαλή και συγκεντρωτική μέθοδο, ώστε ο κάθε χρήστης να μπορεί να επιλέγει με βάση αυτής, τους χρήστες από τους οποίους θέλει να κατεβάσει κάποιο αρχείο. Με αυτόν τον τρόπο, το δίκτυο μπορεί αποτελεσματικά να προσδιορίσει κακόβουλους χρήστες και να τους απομονώσει από το υπόλοιπο σύστημα. Να τονίσουμε πως η συγκέντρωση των τοπικών τιμών εμπιστοσύνης των κόμβων γίνεται χωρίς την χρήση κάποιας κεντρικής μονάδας διαχείρισης και αποθήκευσης, όπως έχουμε αναφέρει προηγουμένως για τα p2p δίκτυα.

Το EigenTrust αναθέτει σε κάθε κόμβο μία μοναδική καθολική τιμή εμπιστοσύνης, η οποία αντικατοπτρίζει την εμπειρία όλων των κόμβων του συστήματος με τον συγκεκριμένο κόμβο. Κάθε κόμβος αποθηκεύει μια local trust value (τοπική τιμή εμπιστοσύνης) για κάθε κόμβο με τον οποίο έχει αλληλεπιδράσει. Ορίζεται ως *τοπική τιμή φήμης* s_{ij} το άθροισμα των τιμών αξιολόγησης, έπειτα από κάθε επιμέρους συναλλαγή, όπου ο κόμβος i μεταφόρτωσε ένα αρχείο από τον κόμβο j :

$$s_{ij} = \sum tr(i, j)$$

Σε ένα κατανεμημένο περιβάλλον p2p, κάθε φορά που ένας ομότιμος κόμβος i κατεβάζει ένα αρχείο από έναν άλλον ομότιμο κόμβο j, ο πρώτος μπορεί να αξιολογήσει την συναλλαγή που έλαβε μέρος με μια τιμή $tr(i, j)$, αναλόγως με την ποιότητα του αρχείου που έλαβε. Εάν πρόκειται για γνήσιο αρχείο τότε αυξάνει την τοπική τιμή εμπιστοσύνης, ενώ εάν πρόκειται για μη γνήσιο αρχείο μειώνει την τοπική τιμή εμπιστοσύνης

Κανονικοποίηση τοπικών τιμών εμπιστοσύνης

Προκειμένου να συγκεντρώσουμε τις τοπικές τιμές εμπιστοσύνης είναι αναγκαίο να τις κανονικοποιήσουμε με κάποιον τρόπο. Διαφορετικά, οι κακόβουλοι χρήστες μπορεί να αναθέτουν αυθαίρετα υψηλές τιμές εμπιστοσύνης σε άλλους κακόβουλους χρήστες και χαμηλές τιμές εμπιστοσύνης σε αξιόπιστους χρήστες, οδηγώντας το σύστημα σε κατάρρευση. Ορίζουμε την *κανονικοποιημένη τοπική τιμή εμπιστοσύνης* c_{ij} ως εξής :

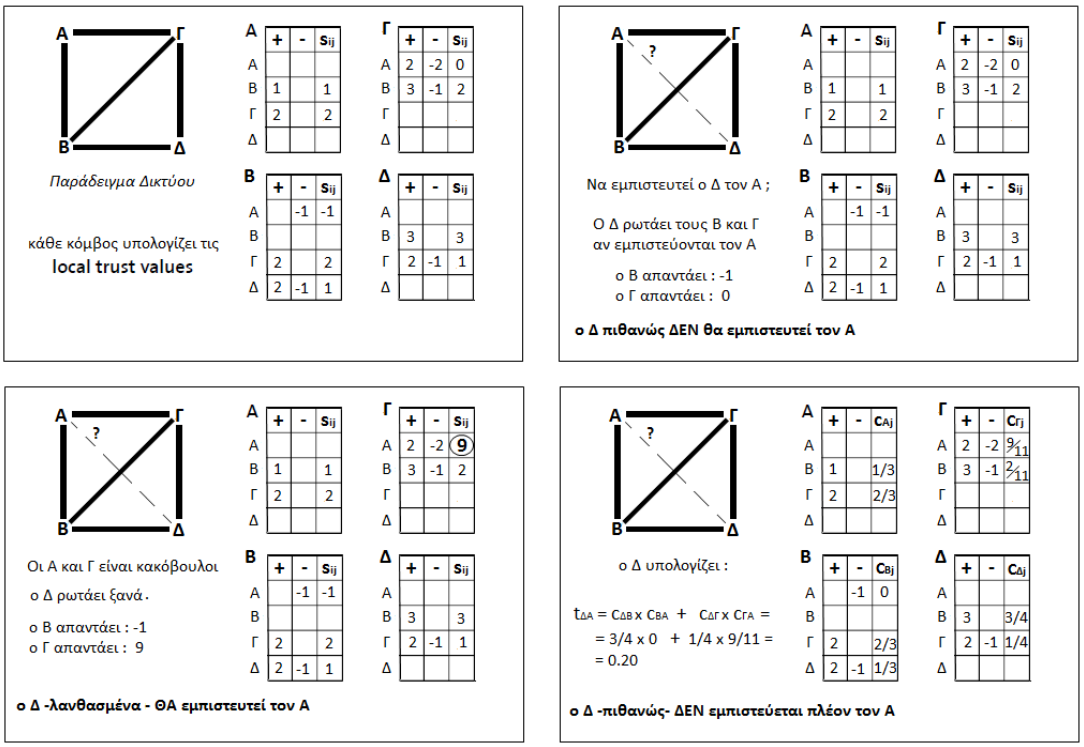
$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)} \qquad (2.1)$$

Το $\sum_j$ υποδεικνύει το άθροισμα των συναλλαγών του κόμβου i με κάθε κόμβο j που έχει αλληλεπιδράσει μαζί του. Η παραπάνω σχέση, διασφαλίζει ότι όλες οι τιμές θα είναι μεταξύ του 0 και 1. Η χρησιμότητα της κανονικοποίησης με αυτόν τον τρόπο, διευκολύνει τους υπολογισμούς στην συνέχεια, καθώς δεν χρειάζεται σε κάθε επανάληψη του αλγορίθμου να κανονικοποιούνται ξανά οι καθολικές τιμές εμπιστοσύνης των κόμβων.

Συλλογή τοπικών τιμών εμπιστοσύνης

Για να συναθροίσουμε τις κανονικοποιημένες τιμές εμπιστοσύνης πρέπει κάθε κόμβος i να ρωτήσει τους γνωστούς σε αυτόν κόμβους, την γνώμη τους για άλλους κόμβους, με τους οποίους ο i δεν έχει αλληλεπιδράσει ακόμη. Παίρνουμε, επομένως, την τοπική τιμή εμπιστοσύνης t_{ik} του κόμβου i για τον κόμβο k, με βάση την γνώμη των «φίλων» του κόμβου i για τον k και η οποία είναι ίση με :

$$t_{ik} = \sum_j c_{ij} c_{jk} \qquad (2.2)$$



Ο τύπος 2.2 μπορεί να γραφεί και σε συμβολισμό μητρών. Εάν ορίσουμε C την μήτρα $[c_{ij}]$ και $\vec{t}_i$ το διάνυσμα που περιέχει τις τιμές t_{ik}, τότε θα ισχύει ότι : $\vec{t}_i = C^T \vec{c}_i$. Όπως γίνεται προφανές ισχύει η επιθυμητή σχέση $\sum_j t_{ij} = 1$.

Παρόλα αυτά, οι τιμές εμπιστοσύνης που αποθηκεύονται σε κάθε κόμβο i αντικατοπτρίζουν μόνο την εμπειρία του κόμβου i και των γνωστών του κόμβων. Για να αποκτήσει μια ευρύτερη εικόνα του δικτύου, ο κόμβος i μπορεί να ρωτήσει τους φίλους των φίλων του ($\vec{t}_i = \left(C^T\right)^2 \vec{c}_i$). Με την υπόθεση ότι η μήτρα C είναι αμείωτη και απεριοδική, το σύστημα θα συγκλίνει έπειτα από n επαναλήψεις και ο κόμβος i αποκτάει την πλήρη εικόνα του δικτύου $\vec{t}_i = \left(C^T\right)^n \vec{c}_i$. Για πολύ μεγάλο n, το διάνυσμα εμπιστοσύνης $\vec{t}_i$ συγκλίνει στο ίδιο διάνυσμα *για κάθε κόμβο* i, το οποίο είναι το αριστερό κύριο ιδιοδιάνυσμα του C. Με άλλα λόγια, το διάνυσμα $\vec{t}$ είναι ένα καθολικό διάνυσμα εμπιστοσύνης.

Pre-Trusted κόμβοι

Ως pre-trusted κόμβοι αναφέρονται οι κόμβοι εκείνοι που θεωρούνται a priori αξιόπιστοι. Τέτοιοι μπορούν να θεωρηθούν οι σχεδιαστές του p2p δικτύου και οι πρώτοι χρήστες του (δηλαδή οι πιο παλιοί χρήστες).

Basic EigenTrust

Αρχικά παρουσιάζεται ο αλγόριθμος Basic EigenTrust, αγνοώντας προς το παρόν την δομημένη φύση των peer-to-peer δικτύων. Υπάρχει ένας κεντρικός εξυπηρετητής ο οποίος γνωρίζει όλες τις τιμές εμπιστοσύνης c_{ij} και εκτελεί τους υπολογισμούς. Ο στόχος του αλγορίθμου αυτού είναι να υπολογίσει την τιμή $\vec{t}_i = \left(C^T\right)^n \vec{e}$, για μεγάλες τιμές του n και όπου το $\vec{e}$ ορίζεται ως το m-διάνυσμα που αναπαριστά την ομοιόμορφη κατανομή πιθανότητας όλων των ομότιμων κόμβων m γενικώς : $\vec{e} = 1/m$.

$\vec{t}^{(0)} = \vec{p}$;
repeat
  $\vec{t}^{(k+1)} = C^T \vec{t}^{(k)}$;
  $\vec{t}^{(k+1)} = (1-a)\vec{t}^{(k+1)} + a\vec{p}$;
  $\delta = ||t^{(k+1)} - t^{(k)}||$;
until $\delta < \epsilon$;

Αλγόριθμος 1 : Basic EigenTrust αλγόριθμος

Το a είναι μια σταθερά μικρότερη του 1 και το διάνυσμα $\vec{p}$ αφορά τους pre-trusted κόμβους. Για τους υπόλοιπους κόμβους ισχύει $\vec{p}_i = 0$. Ο συγκεκριμένος αλγόριθμος με την παρουσία των δύο παραπάνω μεταβλητών, να τονίσουμε πως δίνει λύσεις σε δυσκολίες όπως οι a priori έννοιες της εμπιστοσύνης, το πρόβλημα των ανενεργών χρηστών, καθώς και στους κακόβουλους χρήστες που συνεργάζονται μεταξύ τους, ανταλλάζοντας υψηλές τιμές εμπιστοσύνης.

Distributed EigenTrust

Ο αλγόριθμος αυτός, όπως φανερώνει η ονομασία του, χρησιμοποιείται στα δομημένα δίκτυα p2p και στον οποίο όλοι οι ομότιμοι κόμβοι συνεργάζονται, ώστε να

υπολογίσουν και να αποθηκεύσουν την καθολική τιμή εμπιστοσύνης τους. Τα χαρακτηριστικά του αλγορίθμου αυτού είναι τα εξής :

- ✓ Κάθε ομότιμος κόμβος του δικτύου αποθηκεύει την δική του καθολική τιμή εμπιστοσύνης.
- ✓ Οι ομότιμοι κόμβοι του δικτύου δεν αποθηκεύουν τις καθολικές τιμές εμπιστοσύνης των υπολοίπων ομότιμων κόμβων του δικτύου.
- ✓ Κάθε ομότιμος κόμβος i μπορεί να υπολογίσει την καθολική τιμή εμπιστοσύνης του, σύμφωνα με την σχέση : $\vec{t}_i^{(k+1)} = (1-a)\left(c_{1i}t_1^{(k)} + ... + c_{ni}t_n^{(k)}\right) + ap_i$.

Definitions:

- A_i: set of peers which have downloaded files from peer i
- B_i: set of peers from which peer i has downloaded files

Algorithm:
Each peer i do {
Query all peers $j \in A_i$ for $t_j^{(0)} = p_j$;
repeat
Compute $t_i^{(k+1)} = (1-a)(c_{1i}t_1^{(k)} + c_{2i}t_2^{(k)} + \ldots + c_{ni}t_n^{(k)}) + ap_i$;
Send $c_{ij}t_i^{(k+1)}$ to all peers $j \in B_i$;
Compute $\delta = |t_i^{(k+1)} - t_i^{(k)}|$;
Wait for all peers $j \in A_i$ to return $c_{ji}t_j^{(k+1)}$;
until $\delta < \epsilon$.;
}

Αλγόριθμος 2 : Distributed EigenTrust Αλγόριθμος

Τα σημαντικά στοιχεία του συγκεκριμένου αλγορίθμου είναι δύο. Πρώτον, οι pre-trusted κόμβοι παραμένουν ανώνυμοι ως pre-trusted κόμβοι και δεν έχουν κίνδυνο να αποκαλυφθεί η ταυτότητά τους. Παρότι, περιμένουμε οι κόμβοι αυτοί να έχουν υψηλές τιμές εμπιστοσύνης και να αποκαλυφθεί η ταυτότητά τους, οι προσομοιώσεις δείχνουν ότι σπάνια έχουν τις υψηλότερες τιμές εμπιστοσύνης (προφανώς έχουν μεγαλύτερες τιμές του μέσου όρου).

Δεύτερον, από την στιγμή που κάθε κόμβος i έχει περιορισμένη αλληλεπίδραση με άλλους κόμβους του δικτύου, οι περισσότεροι όροι κανονικοποιημένης εμπιστοσύνης c_{ji} θα είναι μηδενικοί. Επίσης, τα σύνολα A και B θα είναι μικρά. Συνεπώς,

οδηγούμαστε σε έναν απλό κατανεμημένο αλγόριθμο χωρίς πολλούς και εντατικούς υπολογισμούς.

Secure EigenTrust

Το μεγάλο μειονέκτημα του αλγορίθμου που αναλύθηκε προηγουμένως είναι πως κάθε κόμβος i υπολογίζει την δική του καθολική τιμή εμπιστοσύνης. Άρα, κακόβουλοι κόμβοι μπορούν εύκολα να αναφέρουν μια ψεύτικη τιμή εμπιστοσύνης για τους εαυτούς τους, ξεγελώντας το σύστημα και οδηγώντας το σε κατάρρευση.
Την λύση στο πρόβλημα αυτό δίνει ο αλγόριθμος Secure EigenTrust. Καταρχήν, η παρούσα τιμή εμπιστοσύνης κάθε κόμβου δεν θα υπολογίζεται από τον ίδιο κόμβο, ούτε θα παραμένει σε αυτόν, διότι κινδυνεύει να αλλοιωθεί από κακόβουλους χρήστες. Γι' αυτόν τον λόγο ο υπολογισμός της τιμής εμπιστοσύνης ενός κόμβου θα γίνεται από κάποιον άλλο κόμβο. Σε αυτήν την περίπτωση, κάποιος κακόβουλος χρήστης μπορεί να επιστρέψει λανθασμένο αποτέλεσμα για τον κόμβο αυτό, όταν πάει να υπολογίσει την τιμή εμπιστοσύνης του κόμβου. Καταλήγουμε, λοιπόν, στο συμπέρασμα πως η τιμή εμπιστοσύνης ενός χρήστη του δικτύου θα πρέπει να υπολογίζεται από περισσότερους του ενός χρήστη. Οι χρήστες αυτοί ονομάζονται *score managers*.
Για την ανάθεση των score managers σε κάθε κόμβο i γίνεται χρήση ενός κατανεμημένου πίνακα κατακερματισμού -Distributed Hash Table- (DHT) όπως γίνεται στα δίκτυα CAN (Content Addressable Network) και Chord. Οι DHT ουσιαστικά λειτουργούν όπως οι γνωστοί πίνακες κατακερματισμού, στους οποίους αποθηκεύονται ζεύγη κλειδιού-τιμής και κάθε κόμβος του συστήματος μπορεί να ανακτήσει μια τιμή, η οποία συσχετίζεται με κάποιο κλειδί. Οι score managers κάθε κόμβου ορίζονται κάνοντας hash ένα μοναδικό ID του κόμβου, όπως είναι η IP διεύθυνσή και το TCP port του, σε ένα σημείο του χώρου κατακερματισμού του DHT. Ο κόμβος που καλύπτει αυτή την περιοχή του χώρου του DHT ορίζεται ως score manager αυτού του peer. Αν ένας κόμβος θέλει να μάθει την καθολική τιμή εμπιστοσύνης ενός κόμβου γνωρίζοντας το μοναδικό ID αυτού του κόμβου, μπορεί να εντοπίσει τους score managers του και να τους ρωτήσει. Η ακεραιότητα της reputation πληροφορίας εξαρτάται από την αξιοπιστία των κόμβων που υπολογίζουν και αποθηκεύουν τις τιμές εμπιστοσύνης. Ωστόσο η τυχαία επιλογή των score managers για κάθε κόμβο και το γεγονός ότι σε περίπτωση που διαφέρουν οι καθολικές τιμές

εμπιστοσύνης που στέλνουν οι score managers για έναν συγκεκριμένο κόμβο, λαμβάνεται υπόψη η πλειοψηφία των score managers, έχει ως αποτέλεσμα να μειώνεται η πιθανότητα παραποίησης των τιμών εμπιστοσύνης από κακόβουλους score managers.

foreach *peer* i **do**
Submit local trust values $\vec{c_i}$ to all score managers at positions $h_m(pos_i)$, $m = 1 \ldots M-1$;
Collect local trust values $\vec{c_d}$ and sets of acquaintances B_d^i of daughter peers $d \in D_i$;
Submit daughter d's local trust values c_{dj} to score managers $h_m(pos_d)$, $m = 1 \ldots M-1, \forall j \in B_d^i$;
Collect acquaintances A_d^i of daughter peers;
foreach *daughter peer* $d \in D_i$ **do**
Query all peers $j \in A_d^i$ for $c_{jd}p_j$;
repeat
Compute $t_d^{(k+1)} = (1-a)(c_{1d}t_1^{(k)} + c_{2d}t_2^{(k)} + \ldots + c_{nd}t_n^{(k)}) + ap_d$;
Send $c_{dj}t_d^{(k+1)}$ to all peers $j \in B_d^i$;
Wait for all peers $j \in A_i^d$ to return $c_{jd}t_j^{(k+1)}$;
until $|t_d^{(k+1)} - t_d^{(k)}| < \epsilon$.;
end
end

Αλγόριθμος 3 : Secure EigenTrust αλγόριθμος

Κάθε κόμβος i έχει M score managers των οποίων οι συντεταγμένες βρίσκονται με την εφαρμογή στο μοναδικό ID του κόμβου των συναρτήσεων hash : $h_0, h_1, ..., h_{M-1}$. Το διάνυσμα pos_i δίνει τις συντεταγμένες του κόμβου i στον χώρο του DHT. Τέλος, κάθε κόμβος i συμπεριφέρεται και ως score manager, επομένως, χρειάζεται ένα σύνολο D_i που περιέχει τους δείκτες των κόμβων των οποίων οι τιμές εμπιστοσύνης υπολογίζονται από τον κόμβο i. Γι' αυτόν τον λόγο και αναφέρονται ως παιδιά του κόμβου i.

Η χρησιμοποίηση του αλγορίθμου Secure EigenTrust σε κατανεμημένα δίκτυα p2p έχει πολλά πλεονεκτήματα. Κατά πρώτον, διασφαλίζει την ανωνυμία των χρηστών του, καθώς, δεν είναι δυνατόν σε κάποιον κόμβο να βρει το ID του κόμβου του οποίου υπολογίζει την τιμή εμπιστοσύνης. Έτσι, κακόβουλοι κόμβοι δεν μπορούν να αυξήσουν τις τιμές εμπιστοσύνης άλλων κακόβουλων κόμβων. Εν συνεχεία, η τυχαιότητα των συντεταγμένων στις οποίες τοποθετούνται οι καινούριοι κόμβοι, δεν τους επιτρέπουν να βρουν την θέση τους στον DHT και να υπολογίσουν την καθολική τιμή εμπιστοσύνης τους.

Επιπροσθέτως, η παρουσία των pre-trusted χρηστών εγγυάται την σύγκλιση του αλγορίθμου και δεν επιτρέπει τις συνεργασίες κακόβουλων χρηστών που γνωρίζονται μεταξύ τους και ανταλλάζουν υψηλές τιμές τοπικών τιμών εμπιστοσύνης με απώτερο σκοπό να αποκτήσουν υψηλές τιμές καθολικής εμπιστοσύνης.

Η προσέγγιση, όμως, που χρησιμοποιείται στο Secure EigenTrust χαρακτηρίζεται από κάποια σοβαρά μειονεκτήματα.

- Εφαρμόζεται μόνο σε κατανεμημένα p2p δίκτυα και δεν είναι κατάλληλος για δίκτυα p2p business.
- Εξαιτίας της χρήσης κατακερματισμένων πινάκων, το σύστημα είναι ευπαθές σε DHT-threats που έχουν να κάνουν με την δρομολόγηση των πληροφοριών μέσα στους hash tables.
- Η αυθαίρετη επιλογή των pre-trusted χρηστών και η τυχαία κατανομή τους δεν προσφέρει αντικειμενικότητα στο σύστημα.
- Ο αλγόριθμος δεν αποτρέπει τους score managers να δίνουν επίτηδες παραποιημένες πληροφορίες.
- Πολλές άχρηστες και παλιές πληροφορίες αποθηκεύονται και δεν σβήνονται από το σύστημα. Παρότι η εμπιστοσύνη στο EigenTrust είναι δυναμική, θα έπρεπε να λαμβάνονται υπόψη μόνο τα πρόσφατα στατιστικά.

Καθολικές Τιμές Εμπιστοσύνης (Global Trust Values)

Όπως έχει γίνει αντιληπτό, η χρήση των καθολικών τιμών εμπιστοσύνης είναι ιδιαίτερα σημαντική για ένα p2p σύστημα για δύο λόγους. Κατά κύριο λόγο, η χρήση τους μπορεί να απομονώσει τους κακόβουλους χρήστες του συστήματος. Χρήστες με χαμηλές τιμές καθολικής εμπιστοσύνης έχουν χαμηλό δείκτη αξιοπιστίας και οι άλλοι ομότιμοι κόμβοι δεν τους προτιμούν για μεταφόρτωση αρχείων. Με την πάροδο του χρόνου, οι κακόβουλοι χρήστες απομονώνονται από το ίδιο το σύστημα. Επιπροσθέτως, η χρήση των καθολικών τιμών εμπιστοσύνης δίνει επιπλέον κίνητρο στους χρήστες να μοιράζονται με τους υπόλοιπους ομότιμους. Οι αξιόπιστοι χρήστες επιβραβεύονται με το να συνδέονται με πιο πολλούς χρήστες και να έχουν μεγαλύτερο εύρος ζώνης. Έτσι, ακόμα πιο πολλοί χρήστες θα προσπαθούν να είναι αξιόπιστοι κρατώντας καθαρό το δίκτυο από κακόβουλο υλικό.

Extended EigenTrust

Εκτός από τον Secure EigenTrust έχουν παρουσιαστεί και άλλοι εκτεταμένοι αλγόριθμοι του EigenTrust που σαν στόχο έχουν την βελτιστοποίηση του. Θα αναφέρουμε συνοπτικά μερικές από τις προτάσεις-προσεγγίσεις που έχουν γίνει.

- Καινούρια φόρμουλα για τον υπολογισμό των κανονικοποιημένων τοπικών τιμών εμπιστοσύνης η οποία δίνεται από την σχέση : $c_{ij} = \frac{s_{ij}}{\sum_j abs(s_{ij})}$.
- Καινούρια προσέγγιση για την ανάθεση των score managers και την αποθήκευση της reputation πληροφορίας χωρίς την χρήση DHT. Η χρησιμοποίηση bootstrap μηχανισμών και δυναμικών rooting tables οδηγεί σε μια προσέγγιση για αδόμητα p2p δίκτυα.
- Καινούρια τεχνική ανάθεσης pre-trusted κόμβων με στόχο την ομοιόμορφη κατανομή τους.

2.2.2 Μοντέλο ROCQ

Το πλαίσιο ROCQ (Reputation Opinion Credibility Quality) είναι ένα σύστημα διαχείρισης εμπιστοσύνης που στηρίζεται στην φήμη των χρηστών του. Τα αρχικά του ROCQ σημαίνουν Φήμη, Γνώμη, Αξιοπιστία, Ποιότητα. Οι κόμβοι του δικτύου, έπειτα από κάθε συναλλαγή, στέλνουν πληροφορίες για την ποιότητα της συναλλαγής. Οι πληροφορίες αυτές συναθροίζονται για να σχηματίσουν την φήμη κάθε κόμβου. Η συλλογή, η αποθήκευση, η συνάθροιση και η διάδοση των δεδομένων εμπιστοσύνης γίνονται με κατανεμημένο τρόπο.

Opinion

Έπειτα από κάθε συναλλαγή, στην οποία συμμετείχε ένας κόμβος i, καλείται να σχηματίσει μια γνώμη για το πόσο ικανοποιητική ήταν η συναλλαγή αυτή. Ο όρος O_{ij}^k αναφέρεται στην γνώμη του κόμβου i για την k-οστή συναλλαγή του με τον κόμβο j και είναι κανονικοποιημένη ώστε να παίρνει τιμές ανάμεσα στο σύνολο $[0,1]$.

Ένας κόμβος μπορεί να επιλέξει να κρατήσει αρχείο με τις δικές του άμεσες συναλλαγές με άλλους κόμβους, για να υπολογίσει τον μέσο όρο από τις γνώμες που έχει σχηματίσει. Βέβαια, η αποθήκευση κάθε περασμένης συναλλαγής του κόμβου φαντάζει ως μη πρακτική τεχνική. Γι' αυτόν τον λόγο μπορεί ένα κόμβος να αποφασίσει να αποθηκεύει τις n πρόσφατες συναλλαγές ή τις συναλλαγές που έγιναν σε ένα ορισμένο διάστημα χρόνου.
Παρόλα αυτά, σε ένα σύστημα όπως το p2p, όπου υπάρχουν πολλοί κόμβοι και καθημερινά γίνονται πολλές συναλλαγές μεταξύ των κόμβων, η αποθήκευση μεγάλου όγκου δεδομένων είναι σπάταλη και καθόλου πρακτική. Αντί αυτού, λοιπόν, κάθε κόμβος μπορεί να φυλάσσει έναν μέσο όρο από τις γνώμες που έχει σχηματίσει από τις συναλλαγές με κάποιον κόμβο.

Επομένως, ορίζουμε ως *τοπικό μέσο όρο γνώμης* (average local opinion) :

$$O_{ij}^{avg} = \frac{\sum_k O_{ij}^k}{N_{ij}} \qquad (2.3)$$

Ο κόμβος i εκτός από την γνώμη O_{ij}^{avg} για τον κόμβο j, αποθηκεύει και τον αριθμό των συναλλαγών N_{ij} που έχει με τον κόμβο j, καθώς και την διασπορά s_{ij}^2 της συμπεριφοράς του κόμβου j. Να επισημάνουμε πως έπειτα από κάθε συναλλαγή με τον κόμβο j, ο κόμβος i ενημερώνει την γνώμη O_{ij}^{avg}. Από το γεγονός αυτό, γίνεται αντιληπτή και η δυναμικότητα του μοντέλου ROCQ (dynamic trust).

Reputation

Η φήμη (reputation) ενός κόμβου j είναι το τελικό αποτέλεσμα της συνάθροισης των πληροφοριών που στέλνουν διάφοροι άλλοι κόμβοι του δικτύου για τον j. Αναπαριστά την καθολική εικόνα του κατά πόσο ένας κόμβος του συστήματος είναι πιθανόν να επιδιώξει την αλληλεπίδραση με τον κόμβο j. Η φήμη ενός κόμβου είναι και αυτή κανονικοποιημένη μεταβλητή που παίρνει τιμές μεταξύ 0 και 1.
Η φήμη ενός κόμβου j υπολογίζεται στον κόμβο m συνυπολογίζοντας τις γνώμες των υπολοίπων κόμβων για τον j, την τιμή ποιότητας (quality value) που στέλνουν οι

κόμβοι για τον j και την αξιοπιστία των κόμβων αυτών. Η σχέση η οποία δίνει την φήμη $R_{m\,j}$ του κόμβου j είναι :

$$R_{m\,j} = \frac{\sum_i O_{i\,j}^{avg} \cdot C_{mi} \cdot Q_{i\,j}}{\sum_i C_{mi} \cdot Q_{i\,j}} \qquad (2.4)$$

όπου $R_{m\,j}$ είναι η φήμη του κόμβου j, C_{mi} είναι η αξιοπιστία του κόμβου i σύμφωνα με τον κόμβο m, $O_{i\,j}^{avg}$ είναι η μέση τιμή της γνώμης του κόμβου i για τον κόμβο j και $Q_{i\,j}$ είναι η συσχετισμένη με τον j τιμή ποιότητας του κόμβου i. Κατά συνέπεια, ο κόμβος m δίνει περισσότερη βαρύτητα στις εκτιμήσεις που θεωρούνται υψηλής ποιότητας και προέρχονται από κόμβους με μεγάλο βαθμό αξιοπιστίας στα μάτια του κόμβου m.

Credibility

Η ύπαρξη μιας μεταβλητής όπως είναι η αξιοπιστία (credibility) των πληροφοριοδοτών κόμβων του συστήματος είναι πολύ σημαντική για το ίδιο το δίκτυο. Η απουσία της θα καθιστούσε το δίκτυο αδύναμο να διακρίνει τις αληθινές από τις ψευδείς πληροφορίες. Κακόβουλοι κόμβοι θα μπορούσαν να αξιολογήσουν λανθασμένα είτε από αντιδικία είτε για άλλους σκοπούς άλλους ομότιμους κόμβους. Επιπρόσθετα, κακόβουλοι κόμβοι θα μπορούσαν να συνεργαστούν μεταξύ τους, ώστε να ανεβάσουν την δική τους φήμη και να μειώσουν την φήμη άλλων κόμβων.
Η αξιοπιστία είναι μια ξεχωριστή μεταβλητή από τις υπόλοιπες του αλγορίθμου ROCQ και η οποία είναι ανεξάρτητη της φήμης ενός κόμβου του συστήματος, καθώς ένας κόμβος μπορεί να είναι τίμιος στις συναλλαγές του με άλλους κόμβους, αλλά να ενεργεί κακόβουλα όταν αξιολογεί την συμπεριφορά άλλων κόμβων. Στον μηχανισμό του ROCQ, η αξιοπιστία ενός κόμβου χρησιμοποιείται ως δείκτης βάρους των πληροφοριών που στέλνει πίσω στο σύστημα. Στην περίπτωση που κάποιος κόμβος δώσει λανθασμένες πληροφορίες για άλλους κόμβους, ο δείκτης αξιοπιστίας του ελαττώνεται και οι μεταγενέστερες αναφορές του θα έχουν μειωμένο αντίκτυπο στην φήμη άλλων κόμβων. Οι δείκτες αξιοπιστίας στηρίζονται στις άμεσες εμπειρίες μεταξύ των κόμβων και εν αντιθέσει με τις γνώμες (opinions) δεν μοιράζονται μεταξύ των κόμβων. Τέλος, κάθε δείκτης αξιοπιστίας κανονικοποιείται ώστε να παίρνει τιμές μεταξύ 0 και 1.

Η αξιοπιστία C_{mi} του κόμβου i σύμφωνα με τον κόμβο m, μας δείχνει την εμπιστοσύνη που έχει ο κόμβος m για τις γνώμες (opinions) τού κόμβου i σχετικά με άλλους κόμβους. Κάθε γνώμη που εκφράζει ο κόμβος i για έναν άλλον κόμβο αξιολογείται ανάλογα με τον δείκτη αξιοπιστίας που έχει ο m για τον i. Κάθε φορά που ο κόμβος i διαδίδει στο σύστημα μία γνώμη, ο κόμβος m ενημερώνει την αξιοπιστία του C_{mi}. Επιπλέον, όταν ο κόμβος m ενημερώνει την αξιοπιστία του κόμβου i, χρησιμοποιεί την τιμή ποιότητας που του προμηθεύει ο i, ώστε να αποφασίσει το ποσό της μεταβολής. Την πρώτη φορά που καλείται ένας κόμβος να εκθέσει μία γνώμη για κάποιον κόμβο ο δείκτης αξιοπιστίας του τίθεται ίσος με 0.5. Έκτοτε, σε κάθε επόμενη γνώμη, η αξιοπιστία ρυθμίζεται σύμφωνα με την ακόλουθη φόρμουλα.

$$C_{mi}^{k+1} = \begin{cases} C_{mi}^{k} + \dfrac{(1 - C_{mi}^{k}) \cdot Q_{ij}}{2} & \alpha\nu \left| R_{mj} - O_{ij}^{avg} \right| < s_{mj} \\ C_{mi}^{k} - \dfrac{C_{mi}^{k} \cdot Q_{ij}}{2} & \alpha\nu \left| R_{mj} - O_{ij}^{avg} \right| > s_{mj} \end{cases} \qquad (2.5)$$

όπου C_{mi}^{k} η αξιοπιστία του κόμβου i έπειτα από k αναφορές στον κόμβο m, O_{ij}^{avg} η παρούσα γνώμη που έχει αναφέρει ο κόμβος i για τον κόμβο j, Q_{ij} η συσχετισμένη τιμή ποιότητας, R_{mj} η υπολογισμένη από τον κόμβο m φήμη του κόμβου j και s_{mj} η τυπική απόκλιση των γνωμών που έχουν αναφερθεί στον κόμβο m για τον κόμβο j.
Εάν κάποιος κόμβος-πληροφοριοδότης (reporting peer) είναι κακόβουλος, τότε ο δείκτης αξιοπιστίας του μειώνεται σταδιακά, καθώς οι γνώμες που εκθέτει δεν ταιριάζουν με αυτές των υπολοίπων κόμβων. Συνεπώς, ένας κόμβος με χαμηλό δείκτη αξιοπιστίας συμβάλλει λιγότερο στον συναθροιστικό υπολογισμό της φήμης.

Quality

Σε αντίθεση με άλλα συστήματα διαχείρισης εμπιστοσύνης, το πλαίσιο ROCQ επιτρέπει σε έναν κόμβο να αξιολογήσει τις πληροφορίες με τις οποίες ανατροφοδοτεί το σύστημα. Καταρχήν, στην περίπτωση που δώσει λανθασμένες πληροφορίες, ο δείκτης αξιοπιστίας του ελαττώνεται. Γι' αυτό, ο κόμβος μπορεί να μειώσει τον

δείκτη ποιότητας για τις γνώμες που δίνει και δεν είναι απόλυτα σίγουρος, ρισκάροντας λιγότερο την αξιοπιστία του σε περίπτωση που η κρίση σου είναι λανθασμένη. Κατά δεύτερον, οι συναλλαγές ποικίλουν ως προς την σπουδαιότητα τους. Ένας κόμβος θα μπορούσε να συμπεριφερθεί τίμια σε κάμποσες μικρές συναλλαγές και να εξαπατήσει σε κάποια μεγαλύτερη. Όπως και οι υπόλοιπες παράμετροι, ο δείκτης ποιότητας είναι κανονικοποιημένος μεταξύ των τιμών 0 και 1.
Κάθε φορά που ο κόμβος i στέλνει την ενημερωμένη μέση τιμή γνώμης για τον δείκτη j O_{ij}^{avg}, στέλνει ακόμη και τον δείκτη ποιότητας Q_{ij}. Ο δείκτης ποιότητας Q_{ij} αναπαριστά την ποιότητα που προσδίδει ο κόμβος i στην πληροφορία που στέλνει (την γνώμη του για τον κόμβο j). Ο δείκτης Q_{ij} επιτρέπει στον κόμβο i να εκφράσει την δύναμη που έχει η γνώμη του. Υποθέτουμε ότι η συμπεριφορά εμπιστοσύνης του j είναι μια τυχαία κατανεμημένη μεταβλητή. Το δείγμα της τυχαίας αυτής μεταβλητής, ύστερα από αλληλεπιδράσεις του κόμβου i με τον j, προσεγγίζει μια μέση τιμή και μια τυπική απόκλιση. Η μέση τιμή είναι η O_{ij}^{avg} και η τυπική απόκλιση η s_{ij}. Ο δείκτης ποιότητας της γνώμης ορίζεται ως το επίπεδο εμπιστοσύνης, όπου η πραγματική μέση τιμή εμπιστοσύνης για έναν κόμβο βρίσκεται μέσα στο διάστημα εμπιστοσύνης :

$$O_{ij}^{avg} \cdot \left(1 \pm \frac{r}{100}\right) \qquad (2.6)$$

όπου r είναι μια παράμετρος του συστήματος που δηλώνει το μέγεθος του διαστήματος εμπιστοσύνης ως ποσοστό της μέσης τιμής του δείγματος και παίρνει τιμές από 5 έως 30. Για να υπολογίσουμε τα επίπεδα εμπιστοσύνης με άγνωστες παραμέτρους, την πραγματική μέση τιμή και την τυπική απόκλιση, κάνουμε χρήση της t-κατανομής (*Student's t-distribution)*. Η τιμή t της παραπάνω κατανομής δίνεται από την εξίσωση :

$$t = \frac{r}{100} \cdot \frac{O_{ij}^{avg} \cdot \sqrt{N_{ij}}}{s_{ij}} \qquad (2.7)$$

Έτσι ο δείκτης ποιότητας μπορεί να υπολογιστεί από την παρακάτω σχέση, όπου B είναι η *Ατελής Συνάρτηση* B και ορίζεται ως $B(z;a,b) \equiv \int_0^z u^{a-1} \cdot (1-u)^{b-1} du$.

$$Q_{ij} = 1 - B\left(\frac{(N_{ij}-1)}{(N_{ij}-1)+t^2}; \frac{1}{2}\cdot(N_{ij}-1), \frac{1}{2}\right) \qquad (2.8)$$

Στην περίπτωση που κάποιος κόμβος έχει μόνο μια συναλλαγή με έναν άλλον κόμβο οι δύο προηγούμενες εξισώσεις δεν μπορούν να επιλυθούν. Τότε, υπάρχει ως προεπιλογή ο δείκτης ποιότητας να παίρνει την τιμή 1.

Η Αρχιτεκτονική του ROCQ

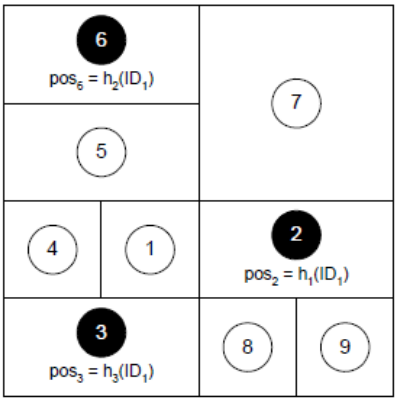
Το μοντέλο εμπιστοσύνης ROCQ εξαρτάται ξεκάθαρα από την αρχιτεκτονική του συστήματος και από την υλοποίηση του ROCQ. Από την στιγμή που δεν υπάρχει κάποιο είδος κεντροποιημένης αρχής, οποιαδήποτε υλοποίηση του χρειάζεται να συλλέξει, να αποθηκεύσει και να διασπείρει τις πληροφορίες εμπιστοσύνης με κατανεμημένο τρόπο. Επιπλέον, όλη αυτή η διαδικασία πρέπει να επιτευχθεί όσο γίνεται πιο αποτελεσματικά και οικονομικά. Η ακόλουθη υλοποίηση του ROCQ προϋποθέτει ένα επικαλυπτόμενο δίκτυο το οποίο θα παρέχει έναν ασφαλή, αξιόπιστο και ντετερμινιστικό τρόπο δρομολόγησης των μηνυμάτων (πληροφοριών). Τέτοια δίκτυα είναι τα CAN, Chord και Pastry.

Τα συγκεκριμένα δίκτυα κάνουν χρήση των κατανεμημένων πινάκων κατακερματισμού (DHT), με παρόμοιο τρόπο όπως και στο μοντέλο EigenTrust (βλ. Ενότητα 3.2.1), για να αντιστοιχήσουν αντικείμενα σε συγκεκριμένο χώρο στον πίνακα κατακερματισμού. Οι DHT παρέχουν την επιθυμητή λειτουργία των score managers της παρούσας υλοποίησης. Ο score manager ενός κόμβου είναι ένας άλλος κόμβος του δικτύου που αποθηκεύει όλη την πληροφορία εμπιστοσύνης σχετική με αυτόν τον κόμβο. Όλες οι γνώμες που αναφέρονται γι' αυτόν τον κόμβο και αιτήσεις για την φήμη του κόμβου αυτού δρομολογούνται στον score manager. Ως εκ τούτου, οι score managers λειτουργούν ως η αποκεντρωμένη βάση δεδομένων εμπιστοσύνης.

- **Score Managers:** Η επικαλυπτόμενη δομή των DHT προσδιορίζει τυχαία και ομοιόμορφα M score managers για κάθε κόμβο του δικτύου. Αυτό επιτυγχάνεται, κατακερματίζοντας ένα ευρέως γνωστό αναγνωριστικό κάθε κόμβου, όπως είναι η διεύθυνση IP. Ο πλησιέστερος κόμβος στον χώρο του κατακερματισμένου αναγνωριστικού ID, ορίζεται ως score manager για τον κόμβο αυτό. Πολλαπλοί

score managers ορίζονται είτε κάνοντας χρήση πολλαπλών συναρτήσεων κατακερματισμού είτε ορίζοντας τους M πλησιέστερους κόμβους στο κατακερματισμένο ID του χώρου κλειδιού (key space) του DHT. Έπειτα από κάθε αλληλεπίδραση, οι δύο κόμβοι που συμμετείχαν στη συναλλαγή εκθέτουν στους score managers τους τις ενημερωμένες γνώμες τους, μαζί με τους συσχετισμένους δείκτες ποιότητας. Οι score managers με την σειρά τους συνυπολογίζουν τις τιμές αυτές ώστε να εξάγουν την καθολική τιμή εμπιστοσύνης ή, όπως αναφέρεται στο ROCQ, τον δείκτη φήμης.

Οι score managers απαντούν, επίσης, σε queries που αιτούνται από άλλους κόμβους, που θέλουν να αλληλεπιδράσουν με έναν κόμβο για τον οποίο είναι υπεύθυνοι. Μαζί με τον δείκτη φήμης R_{mj} για έναν κόμβο, ο score manager m αποθηκεύει τον αριθμό των γνωμών N_{mj} που έχει λάβει για τον j και την διασπορά s_{mj}^2. Οι πληροφορίες αυτές χρειάζονται για τον υπολογισμό του δείκτη ποιότητας που ο score manager επισυνάπτει στον δείκτη φήμης.



Σχήμα 2.2 : Δισδιάστατος hash space στο CAN

Στο σχήμα 2.2 φαίνεται ένα στιγμιότυπο hash στο CAN. Το μοναδικό ID του κόμβου 1 αντιστοιχίζεται στα σημεία του χώρου του DHT που καλύπτονται από τους κόμβους 2, 3 και 6 αντίστοιχα με συναρτήσεις κατακερματισμού h_1, h_2 και h_3. Αυτοί οι κόμβοι ορίζονται ως score managers του κόμβου 1. Όταν ένας score manager εγκαταλείπει το σύστημα, μεταφέρει όλες τις πληροφορίες του στον γείτονα κόμβο στο χώρο συντεταγμένων του DHT. Έτσι εξασφαλίζεται η ευρωστία του συστήματος DHT.

- **Ανάκληση Δεδομένων Εμπιστοσύνης:** Όταν ένας κόμβος θελήσει να μάθει την φήμη κάποιου κόμβου προτού αλληλεπιδράσει μαζί του, εντοπίζει τους M score managers του κόμβου χρησιμοποιώντας το υπόστρωμα του DHT και τους ρωτάει την φήμη του κόμβου αυτού. Κάθε score manager, απαντάει με έναν δείκτη φήμης και τον συσχετισμένο δείκτη ποιότητας. Ο αιτών κόμβος υπολογίζει την μέση τιμή φήμης για τον εν λόγω κόμβο, R_{ij}^{avg}, χρησιμοποιώντας τους δείκτες ποιότητας και αξιοπιστίας των score managers, καθώς ένας score manager μπορεί να είναι κακόβουλος και να αποστείλει ψευδείς τιμές φήμης. Ο υπολογισμός της μέσης τιμής φήμης γίνεται ομοίως με την σχέση 2.4, μόνο που αντί για τις γνώμες O_{ij}^{avg} έχουμε τους δείκτες φήμης R_{mj}.

$$R_{ij}^{avg} = \frac{\sum_y R_{yj} \cdot C_{iy} \cdot Q_{yj}}{\sum_y C_{iy} \cdot Q_{yj}} \qquad (2.9)$$

όπου R_{yj} ο δείκτης φήμης που λήφθηκε από τον score manager y για τον κόμβο j.

Συμπεράσματα

Το ROCQ είναι ένας απλός μηχανισμός στηριζόμενος στη φήμη των χρηστών του, που επιδιώκει να επιβάλλει την σωστή συμπεριφορά σε ένα δίκτυο. Παρόλα αυτά, το μοντέλο ROCQ δεν μπορεί να αποτρέψει όλους τους πιθανούς κινδύνους από κακόβουλες επιθέσεις των οντοτήτων ενός αυτόνομου συστήματος και αυτό διότι, πολλές επιθέσεις γίνονται στο επικαλυπτόμενο δίκτυο, το οποίο χρησιμοποιείται για την επικοινωνία μεταξύ των χρηστών και την δρομολόγηση πληροφοριών. Τέτοιοι κίνδυνοι, βέβαια, δεν θα εξεταστούν εδώ, καθώς βρίσκονται έξω από το πεδίο του ROCQ.

Η ύπαρξη δύο επιπλέον μεταβλητών, του δείκτη αξιοπιστίας και του δείκτη ποιότητας, στον αλγόριθμο του ROCQ, προσδίδει σε ένα αυτόνομο σύστημα μεγαλύτερη ασφάλεια. Αυτό έχει διαπιστωθεί και πειραματικά, όπου σε προσομοιώσεις χωρίς την παρουσία των δύο παραπάνω δεικτών, η απόδοση του αλγορίθμου μειώνεται κατά 30%. Το μοντέλο ROCQ παρότι θεωρείται αρκετά αξιόπιστο στην γενική περίπτωση, όταν οι κακόβουλοι κόμβοι αποτελούν την

πλειοψηφία σε ένα δίκτυο p2p, τότε η απόδοση του μειώνεται δραματικά και επιστρέφει σωστά αποτελέσματα μόνο στο 25-30% των περιπτώσεων.

2.2.3 Μοντέλο Bayesian

Το μοντέλο Bayesian *(Bayesian Network-Based Trust Model)* αναφέρεται σε καθεαυτό p2p συστήματα, όπου κάθε οντότητα είναι πάροχος και χρήστης των αγαθών. Η ασφάλεια σε τέτοια συστήματα είναι περισσότερο αναγκαία από τα κεντροποιημένα δίκτυα όπου υπάρχει τουλάχιστον ένας κόμβος διαχειριστής. Σε κάθε οντότητα ενός p2p δικτύου προσκολλάται ένας πράκτορας (agent) ο οποίος παρέχει τις πληροφορίες εμπιστοσύνης και φήμης της οντότητας (trust and reputation).
Οι περισσότερες εφαρμογές και πειράματα για την εμπιστοσύνη και την φήμη, εστιάζονται μόνο στο ένα από τα δύο στοιχεία. Θεωρητικά, βέβαια, ο συνδυασμός των δύο, εμπιστοσύνης και φήμης δεν είναι άγνωστος. Στα p2p δίκτυα ένας πράκτορας χτίζει δύο είδη εμπιστοσύνης προς έναν άλλο πράκτορα.
Πρώτον, την εμπιστοσύνη σε έναν άλλο πράκτορα να παρέχει υπηρεσίες και δεύτερον την εμπιστοσύνη σε έναν άλλο πράκτορα να παρέχει συστάσεις για κάποιον τρίτο πράκτορα. Το δεύτερο είδος εμπιστοσύνης συχνά αναφέρεται και ως αξιοπιστία. Στο κεφάλαιο αυτό θα αναπτύξουμε μια εφαρμογή peer-to-peer διαμοιρασμού αρχείων (file sharing) βασισμένη στο Bayesian μοντέλο.
Εντούτοις, η μέθοδος είναι αρκετά γενική και μπορεί να εφαρμοστεί σε υπηρεσίες web, ηλεκτρονικού εμπορίου ή ακόμα και σε κατανεμημένα περιβάλλοντα peer-to-peer computing.

Μηχανισμός Εμπιστοσύνης και Φήμης

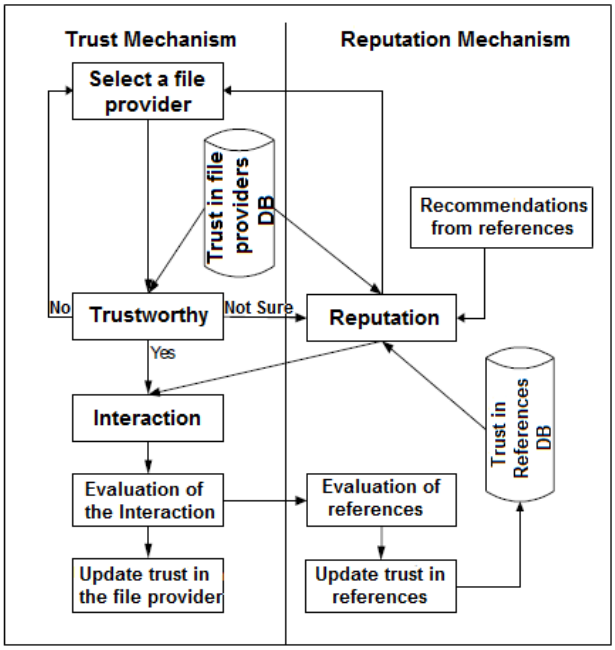
Το συγκεκριμένο μοντέλο κατά Bayes εξειδικεύεται σε file sharing εφαρμογές όπως είναι το Gnutella. Οι πράκτορες δεν παρέχουν ομοιόμορφες υπηρεσίες σε ένα τέτοιο δίκτυο. Παραδείγματος χάριν, κάποιος ομότιμος κόμβος μπορεί να είναι συνδεδεμένος σε ευρυζωνικό δίκτυο και να μοιράζεται αρχεία με άλλους σε υψηλές ταχύτητες, σε κάποιον άλλο κόμβο μπορεί να του αρέσει η μουσική και να μοιράζεται τα αρχεία μουσικής του κοκ. Όπως τονίσαμε και προηγουμένως, κάθε πράκτορας αναπτύσσει δύο είδη εμπιστοσύνης, την εμπιστοσύνη στην ικανότητα του παρόχου

(να μοιράζεται αρχεία) και στην αξιοπιστία των πρακτόρων να προτείνουν τρίτους κόμβους για διαμοιρασμό αρχείων.

Ένα αίτημα αναζήτησης σε file sharing εφαρμογές, οδηγεί συνήθως σε έναν μακρύ κατάλογο προμηθευτών για το ίδιο αρχείο. Εάν ένας κόμβος συμβεί να επιλέξει έναν προμηθευτή με κακή ποιότητα αρχείων ή χαμηλές ταχύτητες μεταφόρτωσης, τότε ο κόμβος θα σπαταλήσει χρόνο και προσπάθεια, γεγονός που μπορεί να οδηγήσει στην απογοήτευση του χρήστη και εγκατάλειψη του συστήματος.

Προκειμένου να λυθεί το πρόβλημα αυτό, χρησιμοποιούμε το μηχανισμό της εμπιστοσύνης και της φήμης όπως φαίνεται στο σχήμα 2.3. Μόλις λάβει ένας κόμβος έναν κατάλογο προμηθευτών αρχείων για μια δεδομένη αναζήτηση, μπορεί να διευθετήσει τον κατάλογο σύμφωνα με την εμπιστοσύνη του σε αυτούς τους προμηθευτές αρχείων. Κατόπιν, ο κόμβος επιλέγει έναν από τους προμηθευτές αρχείων της λίστας. Εάν ο προμηθευτής αρχείων είναι αξιόπιστος, σύμφωνα με τις προηγούμενες εμπειρίες του κόμβου, τότε ο κόμβος θα αλληλεπιδράσει με τον προμηθευτή αρχείων (μεταφόρτωση αρχείων). Εάν ο προμηθευτής αρχείων δεν είναι αξιόπιστος, ο κόμβος θα επιλέξει έναν άλλο προμηθευτή αρχείων για να αλληλεπιδράσει μαζί του. Εάν ο κόμβος δεν είναι βέβαιος για την αξιοπιστία του προμηθευτή αρχείων, παραδείγματος χάριν, ο κόμβος δεν έχει καμία αλληλεπίδραση ή μόνο μερικές αλληλεπιδράσεις με τον προμηθευτή αρχείων, τότε μπορεί να ζητήσει από άλλους κόμβους να του κάνουν συστάσεις.

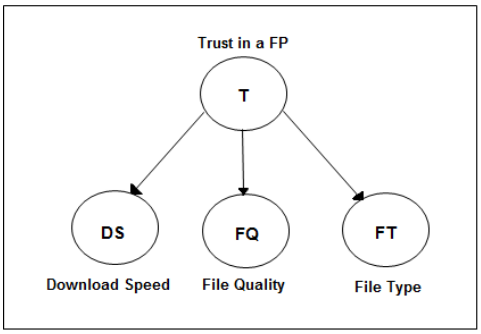
Μετά από κάθε αλληλεπίδραση, ο κόμβος ενημερώνει την τιμή εμπιστοσύνης του για τον προμηθευτή αρχείων σύμφωνα με την αξιολόγησή του για την αλληλεπίδραση αυτή. Εάν η αλληλεπίδραση είναι ικανοποιητική, θα αυξήσει την τιμή εμπιστοσύνης του για τον προμηθευτή αρχείων, ενώ εάν η αλληλεπίδραση δεν είναι ικανοποιητική, θα μειώσει την τιμή εμπιστοσύνης του για τον προμηθευτή αρχείων. Στην περίπτωση που η απόφαση για αλληλεπίδραση είναι βασισμένη σε συστάσεις άλλων κόμβων, ο κόμβος θα ενημερώσει επιπλέον την τιμή εμπιστοσύνης του για κάθε έναν από τους κόμβους αυτούς (καλούμε αυτούς τους κόμβους «κριτές»).



Σχήμα 2.3 : Ο μηχανισμός εμπιστοσύνης-φήμης

Εμπιστοσύνη κατά Bayes

Σε ένα δίκτυο peer-to-peer, οι ικανότητες των προμηθευτών αρχείων δεν είναι ομοιόμορφες. Παραδείγματος χάριν, μερικοί προμηθευτές αρχείων μπορούν να συνδεθούν μέσω ενός ευρυζωνικού δικτύου, ενώ άλλοι συνδέονται μέσω ενός αργού διαποδιαμορφωτή (modem). Μερικοί προμηθευτές αρχείων ενδιαφέρονται για τη μουσική, έτσι μοιράζονται πολλά αρχεία μουσικής. Σε άλλους μπορεί να τους ενδιαφέρουν οι ταινίες και να μοιραστούν περισσότερες ταινίες. Κάποιοι θεωρούν σημαντική την ποιότητα των αρχείων, έτσι κρατούν και μοιράζονται μόνο αρχεία με υψηλή ποιότητα. Επομένως, οι δυνατότητες των προμηθευτών αρχείων μπορούν να παρουσιαστούν σε διάφορες κατηγορίες, όπως η ταχύτητα μεταφόρτωσης, η ποιότητα αρχείου και ο τύπος αρχείου (δείτε το σχήμα 2.4).



Σχήμα 2.4 : Μοντέλο δικτύου Bayesian

Οι ανάγκες κάθε πράκτορα είναι επίσης διαφορετικές στις διάφορες καταστάσεις. Μερικές φορές, μπορεί να θελήσει να ξέρει την συνολική ικανότητα του προμηθευτή αρχείων. Άλλες φορές μπορεί μόνο να ενδιαφερθεί για την ικανότητα του προμηθευτή αρχείων σε κάποια ιδιαίτερη κατηγορία. Οι πράκτορες πρέπει επίσης να αναπτύξουν τη διαφοροποιημένη εμπιστοσύνη στις δυνατότητες των προμηθευτών αρχείων. Παραδείγματος χάριν, ο πράκτορας που θέλει να μεταφορτώσει ένα αρχείο μουσικής από έναν προμηθευτή αρχείων, φροντίζει για το εάν ο προμηθευτής αρχείων είναι σε θέση να παρέχει ένα αρχείο μουσικής με καλή ποιότητα και σε γρήγορη ταχύτητα. Δηλαδή, περιλαμβάνει την δυνατότητα του προμηθευτή αρχείων σε δύο κατηγορίες, την ποιότητα και την ταχύτητα. Και στο σημείο αυτό γεννιέται το ερώτημα, πως ο πράκτορας είναι σε θέση να συνδυάσει τις διάφορες τιμές εμπιστοσύνης για να καταλήξει εάν ο προμηθευτής αρχείων είναι αξιόπιστος. Το Bayesian δίκτυο δίνει λύση στο πρόβλημα αυτό με μια ευέλικτη μέθοδο.

Ένα Bayesian δίκτυο είναι ένα σχεσιακό δίκτυο το οποίο χρησιμοποιεί στατιστικές μεθόδους για να παρουσιάσει τις σχέσεις πιθανοτήτων μεταξύ των διαφορετικών πρακτόρων. Το θεωρητικό του υπόβαθρο στηρίζεται στον κανόνα του Bayes ο οποίος δίνει την υπό συνθήκη πιθανότητα $p(h \mid e)$ μέσω της σχέσης :

$$p(h \mid e) = \frac{p(e \mid h) \cdot p(h)}{p(e)} \qquad (2.10)$$

όπου $p(h)$ είναι η προγενέστερη (δεν λαμβάνει υπόψη καμία πληροφορία για το e) πιθανότητα της υπόθεσης h, $p(e)$ είναι η προγενέστερη πιθανότητα του στοιχείου e και $p(e \mid h)$ είναι η πιθανότητα να συμβεί το e δεδομένου του h.

Ένα naive Bayesian δίκτυο είναι ένα απλό Bayesian δίκτυο. Αποτελείται από έναν κόμβο ρίζα και διάφορους κόμβους φύλλα. Θα χρησιμοποιήσουμε ένα naive Bayesian δίκτυο για να παρουσιάσουμε την εμπιστοσύνη μεταξύ ενός πράκτορα και ενός προμηθευτή αρχείων. Κάθε πράκτορας αναπτύσσει ένα naive Bayesian δίκτυο για κάθε έναν προμηθευτή αρχείων με τον οποίο έχει αλληλεπιδράσει. Κάθε Bayesian δίκτυο αποτελείται έναν κόμβο ρίζα T, που έχει δύο τιμές, *satisfying* (ικανοποιητική)

και *unsatisfying* (μη ικανοποιητική), δηλωμένες με τις αριθμητικές τιμές 1 και 0. Η $p(T=1)$ αντιπροσωπεύει την τιμή της συνολικής εμπιστοσύνης του πράκτορα, στην ικανότητα του προμηθευτή να παρέχει αρχεία. Ουσιαστικά είναι το ποσοστό των αλληλεπιδράσεων που είναι ικανοποιητικές, δηλαδή ο αριθμός των αλληλεπιδράσεων m που είναι ικανοποιητικές, διαιρούμενο με το συνολικό αριθμό αλληλεπιδράσεων n. Η $p(T=0)$ είναι το ποσοστό των μη ικανοποιητικών αλληλεπιδράσεων.

$$p(T=1)=\frac{m}{n} \qquad (2.11)$$
$$p(T=1)+p(T=0)=1$$

Τα φύλλα που βρίσκονται κάτω από την ρίζα αντιπροσωπεύουν τις δυνατότητες των προμηθευτών αρχείων σε διαφορετικές κατηγορίες. Κάθε κόμβος φύλλο συνδέεται με έναν υποθετικό πίνακα πιθανοτήτων (CPT). Ο κόμβος, που ορίζεται με FT (file type), περιλαμβάνει το σύνολο των διαφορετικών τύπων αρχείων. Υποθέτουμε ότι περιλαμβάνει πέντε τιμές, Μουσική, Ταινίες, Έγγραφα, Εικόνες και Λογισμικό. Ο CPT του κόμβου FT φαίνεται στον πίνακα 2.2. Το άθροισμα των πιθανοτήτων κάθε στήλης είναι ίσο με 1.

	T=1	**T=0**
Μουσική	p(FT ="Music"|T = 1)	p(FT ="Music"|T = 0)
Ταινίες	p(FT ="Movie"| T = 1)	p(FT ="Movie"| T = 0)
Έγγραφα	p(FT ="Docu"|T = 1)	p(FT ="Docu"|T = 0)
Εικόνες	p(FT ="Image"|T =1)	p(FT ="Image"|T = 0)
Λογισμικό	p(FT ="Soft"|T = 1)	p(FT ="Soft"|T = 0)

Πίνακας 2.2 : Ο πίνακας CPT του κόμβου FT

Η $p(\text{FT}=\text{"Music"}\,|\,T=1)$ είναι η υπό συνθήκη πιθανότητα που υπολογίζει την πιθανότητα το αρχείο του συγκεκριμένου προμηθευτή να είναι αρχείο μουσικής, δεδομένου ότι η αλληλεπίδραση είναι ικανοποιητική. Μπορεί να υπολογιστεί σύμφωνα με τον ακόλουθο τύπο:

$$p(\text{FT} = \text{"Music"} \mid T = 1) = \frac{p(\text{FT} = \text{"Music"},\ T = 1)}{p(T = 1)} \qquad (2.12)$$

όπου $p(\text{FT} = \text{"Music"},\ T = 1)$ η πιθανότητα οι αλληλεπιδράσεις να είναι ικανοποιητικές και τα αρχεία που περιλαμβάνονται να είναι αρχεία μουσικής.

$$p(\text{FT} = \text{"Music"},\ T = 1) = \frac{m1}{n} \qquad (2.13)$$

όπου m1 είναι ο αριθμός των ικανοποιητικών αλληλεπιδράσεων, όταν τα αρχεία που περιλαμβάνονται είναι αρχεία μουσικής.

Η $p(\text{FT} = \text{"Music"} \mid T = 0)$ υπολογίζει την υπό συνθήκη πιθανότητα τα αρχεία να είναι αρχεία μουσικής, δεδομένου ότι οι αλληλεπιδράσεις δεν είναι ικανοποιητικές. Οι πιθανότητες για άλλους τύπους αρχείων του πίνακα 2.2 υπολογίζονται με παρόμοιο τρόπο.

Ο κόμβος DS υποδηλώνει την κατηγορία ταχύτητα μεταφόρτωσης (download speed). Αποτελείται από τρία αντικείμενα, Γρήγορη (Fast), Μέτρια (Medium) και Αργή (Slow) κάθε ένα από τα οποία καλύπτει μια περιοχή ταχυτήτων. Ο κόμβος FQ υποδηλώνει την κατηγορία ποιότητα αρχείων (file quality). Έχει επίσης τρία αντικείμενα, Υψηλή (High), Μέτρια (Medium) και Χαμηλή (Low). Οι πίνακες τους CPT είναι παρόμοιοι με αυτόν του πίνακα 2.2.

Από την στιγμή που οι πίνακες CPT ενσωματωθούν στο σύστημα, κάθε πράκτορας μπορεί να υπολογίσει την πιθανότητα κάποιος συγκεκριμένος προμηθευτής αρχείων να είναι αξιόπιστος σε διάφορες πτυχές, με τη χρησιμοποίηση του κανόνα του Bayes. Παραδείγματος χάριν, έστω ένας πράκτορας i και ένας προμηθευτής αρχείων j. Ο πράκτορας i θέλει να μάθει αν ο προμηθευτής j είναι αξιόπιστος για να κατεβάσει ένα αρχείο ταινίας με υψηλή ποιότητα. Η πιθανότητα αυτή είναι η εξής : $p(\text{T} = 1 \mid \text{FT} = \text{"Movie"},\ \text{FQ} = \text{"}High\text{"})$. Ανατρέχοντας στους αντίστοιχους πίνακες CPT και με χρήση του κανόνα του Bayes ο πράκτορας παίρνει απάντηση στο ερώτημα του. Οι πράκτορες μπορούν να θέσουν σύμφωνα με τις ανάγκες τους τις κατάλληλες μεταβλητές. Έπειτα από κάθε αλληλεπίδραση, οι πράκτορες ενημερώνουν τα αντίστοιχα Bayesian δίκτυά τους.

Αξιολόγηση αλληλεπιδράσεων

Οι πράκτορες ενημερώνουν τα αντίστοιχα Bayesian δίκτυά τους μετά από κάθε αλληλεπίδραση. Εάν μια αλληλεπίδραση είναι ικανοποιητική, τα m και n του τύπου 2.11, αυξάνονται κατά 1. Εάν δεν είναι ικανοποιητική, μόνο το n αυξάνεται κατά 1. Δύο βασικοί παράγοντες εξετάζονται όταν αξιολογούν οι πράκτορες μια αλληλεπίδραση. Ο βαθμός ικανοποίησής τους, για την ταχύτητα μεταφόρτωσης s_{ds} και για την ποιότητα του μεταφορτωμένου αρχείου s_{fq}. Ο συνολικός βαθμός ικανοποίησης s του πράκτορα σε μια αλληλεπίδραση υπολογίζεται ως εξής : $s = w_{ds} \times s_{ds} + w_{fq} \times s_{fq}$ όπου $w_{ds} + w_{fq} = 1$. Τα w_{ds} και w_{fq} υποδηλώνουν βάρη, που δείχνουν τη σημασία της ταχύτητας μεταφόρτωσης και της ποιότητας των αρχείων σε έναν συγκεκριμένο πράκτορα. Κάθε πράκτορας έχει ένα κατώφλι βαθμού ικανοποίησης s_t. Εάν $s < s_t$, η αλληλεπίδραση δεν είναι ικανοποιητική.

Διαχείριση συστάσεων άλλων πρακτόρων

Στην παρούσα file sharing p2p εφαρμογή, οι χρήστες βρίσκουν τα αρχεία με τη χρησιμοποίηση της λειτουργίας αναζήτησης. Στις περισσότερες των περιπτώσεων, παίρνουν έναν μακρύ κατάλογο προμηθευτών για το ίδιο αρχείο. Προκειμένου να λυθεί το πρόβλημα της επιλογής κάποιο αναξιόπιστου προμηθευτή, χρησιμοποιούμε το μηχανισμό της εμπιστοσύνης και της φήμης. Εάν ο πράκτορας δεν έχει καμία εμπειρία με τον προμηθευτή αρχείων, μπορεί να ζητήσει από άλλους πράκτορες να του κάνουν συστάσεις. Ο πράκτορας μπορεί να στείλει τα διάφορα αιτήματα σύστασης ανάλογα με τις ανάγκες του. Όταν οι άλλοι πράκτορες λαμβάνουν αυτά τα αιτήματα, θα ελέγξουν στους πίνακες CPT των Bayesian δικτύων τους για να δουν εάν μπορούν να απαντήσουν στα συγκεκριμένα ερωτήματα. Ο αιτών πράκτορας μπορεί να λάβει διάφορες τέτοιες συστάσεις συγχρόνως, οι οποίες μπορεί να προέρχονται από αξιόπιστους, αναξιόπιστους ή και άγνωστους πράκτορες. Οι αναξιόπιστοι πράκτορες απορρίπτονται αυτόματα.

Επόμενο βήμα είναι ο υπολογισμός της συνολικής τιμή σύστασης (recommendation) για τον προμηθευτή αρχείων από τον συνδυασμό των αξιόπιστων και άγνωστων αναφορών.

$$r_{ij} = w_t \times \frac{\sum_{l=1}^{k} tr_{il} \cdot t_{lj}}{\sum_{l=1}^{k} tr_{il}} + w_s \times \frac{\sum_{z=1}^{g} t_{zj}}{g} \qquad (2.14)$$

όπου $w_t + w_s = 1$, r_{ij} είναι η συνολική τιμή σύστασης για τον προμηθευτή αρχείων j που παίρνει ο πράκτορας i, τα k και g είναι οι αριθμοί των αξιόπιστων και άγνωστων αναφορών αντίστοιχα, tr_{il} είναι η εμπιστοσύνη που έχει ο χρήστης i στην αξιόπιστη αναφορά l, t_{lj} είναι η εμπιστοσύνη που έχει η αξιόπιστη αναφορά l στον προμηθευτή αρχείων j, t_{zj} είναι η εμπιστοσύνη που έχει η άγνωστη αναφορά z στον προμηθευτή αρχείων j. Τέλος, τα w_t και w_s είναι τα βάρη που υποδηλώνουν το πόσο πολύ ο χρήστης λαμβάνει υπόψη τις συστάσεις από τις αξιόπιστες αναφορές και από τις άγνωστες αναφορές αντίστοιχα.

Δεδομένης μιας τιμής κατωφλίου θ, εάν η συνολική τιμή σύστασης r_{ij} είναι μεγαλύτερη από το θ, ο πράκτορας θα αλληλεπιδράσει με τον προμηθευτή αρχείων, διαφορετικά, όχι.

Εάν ο πράκτορας αλληλεπιδράσει με τον προμηθευτή αρχείων, όχι μόνο θα ενημερώσει την τιμή εμπιστοσύνης του για τον προμηθευτή αρχείων, δηλαδή το αντίστοιχο Bayesian δίκτυό του, αλλά επιπλέον θα ενημερώσει τις τιμές εμπιστοσύνης του για τους πράκτορες που του έδωσαν τις συστάσεις, σύμφωνα με τον ακόλουθο τύπο :

$$tr_{ij}^{n} = a \cdot tr_{ij}^{o} + (1-a) \cdot e_a \qquad (2.15)$$

όπου η tr_{ij}^{n} είναι η νέα τιμή εμπιστοσύνης που ο πράκτορας i έχει στον κριτή j μετά από την ενημέρωση, η tr_{ij}^{o} είναι η προηγούμενη τιμή εμπιστοσύνης, ο a είναι ο ρυθμός εκμάθησης και είναι ένας πραγματικός αριθμός στο διάστημα $[0,1]$ και η e_a είναι η νέα τιμή στοιχείων (evidence) και μπορεί να είναι -1 ή 1. Εάν η τιμή της σύστασης είναι μεγαλύτερη από το θ και η κατόπιν αλληλεπίδραση με τον προμηθευτή αρχείων είναι έγκυρη, τότε τίθεται e_a ίσο με 1. Σε αντίθετη περίπτωση, τίθεται e_a ίσο με -1.

Ένας άλλος τρόπος για να δούμε εάν ένας πράκτορας είναι αξιόπιστος στο να δίνει σωστές συστάσεις είναι η σύγκριση μεταξύ των Bayesian δικτύων δύο πρακτόρων σχετικών με έναν κοινό προμηθευτή αρχείων. Μετά από κάθε σύγκριση, οι πράκτορες ενημερώνουν τις τιμές εμπιστοσύνης τους σύμφωνα με τον τύπο :

$$tr_{ij}^{n} = \beta \cdot tr_{ij}^{o} + (1-\beta) \cdot e_{\beta} \qquad (2.16)$$

όπου οι μεταβλητές είναι αντίστοιχες του τύπου 2.15 και επιπλέον ισχύει ο περιορισμός $\beta > \alpha$ και ότι η μεταβλητή e_{β} παίρνει τιμές στο διάστημα $[0,1]$.

Συμπεράσματα

Όπως αναφέρθηκε και προηγουμένως, η ανάλυση για file sharing εφαρμογές που παρουσιάσαμε στην ενότητα αυτή δεν είναι απαγορευτική για επέκταση του μοντέλου σε άλλες peer-to-peer εφαρμογές. Η χρήση του κανόνα του Bayes και των δικτύων naive Bayesian προσδίδουν στο σύστημα μεγαλύτερη ασφάλεια και καλύτερη απόδοση σε σχέση με τα απλά συστήματα γενικής εμπιστοσύνης. Στην υποενότητα 2.4 θα αναφερθούμε εκτενέστερα στα πλεονεκτήματα και στα μειονεκτήματα του συγκεκριμένου αλγορίθμου σε σύγκριση πάντα με τους υπόλοιπους αλγόριθμους.

2.2.4 Η εμπιστοσύνη στο eBay

Με την εξάπλωση του διαδικτύου στην καθημερινή ζωή των ανθρώπων ήταν αναμενόμενη και η εισβολή του στο εμπόριο και τις συναλλαγές. Η νέα μορφή αγοραπωλησίας προϊόντων άρχισε να γίνεται γνωστή από την δεκαετία του 1980 και περιελάμβανε κυρίως προσφερόμενες υπηρεσίες πάνω σε ηλεκτρονικά συστήματα και το διαδίκτυο. Με την πάροδο των χρόνων το ηλεκτρονικό εμπόριο (e-commerce) πήρε την σημερινή του μορφή. Παρόλα αυτά, πολλοί χρήστες του ηλεκτρονικού εμπορίου, παραμένουν αβέβαιοι για την αξιοπιστία του, καθώς πάρα πολλοί προμηθευτές είναι κακόφημοι. Όπως γίνεται αντιληπτό, η ανάπτυξη της εμπιστοσύνης σε τέτοια συστήματα όπως το ηλεκτρονικό εμπόριο φαντάζει επιτακτική.

Οι κοινότητες ηλεκτρονικού εμπορίου, όπως είδαμε και στην ενότητα 1.4.1, ανήκουν στα κατανεμημένα δίκτυα p2p. Ειδικότερα ανήκουν στα p2p δίκτυα τρίτης γενιάς,

όπου γίνεται χρήση υβριδικών τεχνικών. Στα e-commerce δίκτυα χρησιμοποιείται μια πλατφόρμα client-server, όπου υπάρχει μια κεντρική αρχή που αποθηκεύει την reputation πληροφορία. Η εμπιστοσύνη όμως κερδίζεται με βάση το ιστορικό των συναλλαγών ενός δεδομένου κόμβου και σύμφωνα με την γνώμη των υπολοίπων. Τα πιο γνωστά παραδείγματα e-commerce κοινοτήτων είναι το eBay και το Amazon. Στη συνέχεια του κεφαλαίου θα παρουσιάσουμε τον τρόπο με τον οποίο επιτυγχάνεται η ανάπτυξη της εμπιστοσύνης μεταξύ των χρηστών του eBay.

Το eBay είναι ένα reputation-based σύστημα διαχείρισης της εμπιστοσύνης για online δημοπρασίες. Χρησιμοποιεί απλές βαθμονομημένες κλίμακες για κάθε χρήστη. Έπειτα από κάθε επιτυχή συναλλαγή οι χρήστες , δηλαδή, ο αγοραστής και ο πωλητής καλούνται να βαθμολογήσουν ο ένας τον άλλον με βάση το 1. Οι πωλητές μπορούν να δώσουν θετική αξιολόγηση και να αφήσουν ένα σχόλιο για τον αγοραστή, ενώ οι αγοραστές μπορούν να δώσουν θετική, αρνητική ή ουδέτερη αξιολόγηση για τον πωλητή. Έχουμε λοιπόν : -1 για αρνητική αξιολόγηση, 0 για ουδέτερη αξιολόγηση και 1 για θετική αξιολόγηση. Οι εκτιμήσεις των τελευταίων 6 μηνών λαμβάνονται υπόψη για κάθε χρήστη και έτσι υπολογίζεται η φήμη του (reputation). Οι αγοραστές μπορούν επιπλέον να αξιολογήσουν τους πωλητές με επιπρόσθετα κριτήρια, όπως είναι η συνέπεια της συναλλαγής (προϊόντα πληρώνονται αλλά δεν παραδίδονται ή το αντίστροφο), η επικοινωνία, η ποιότητα σε αναλογία με την τιμή του αντικειμένου συναλλαγής[3], τα μεταφορικά τέλη και τα τέλη παράδοσης. Αυτές οι λεπτομερείς αξιολογήσεις του πωλητή δεν συνυπολογίζονται στην τιμή φήμης του κάθε χρήστη και είναι ανώνυμες.

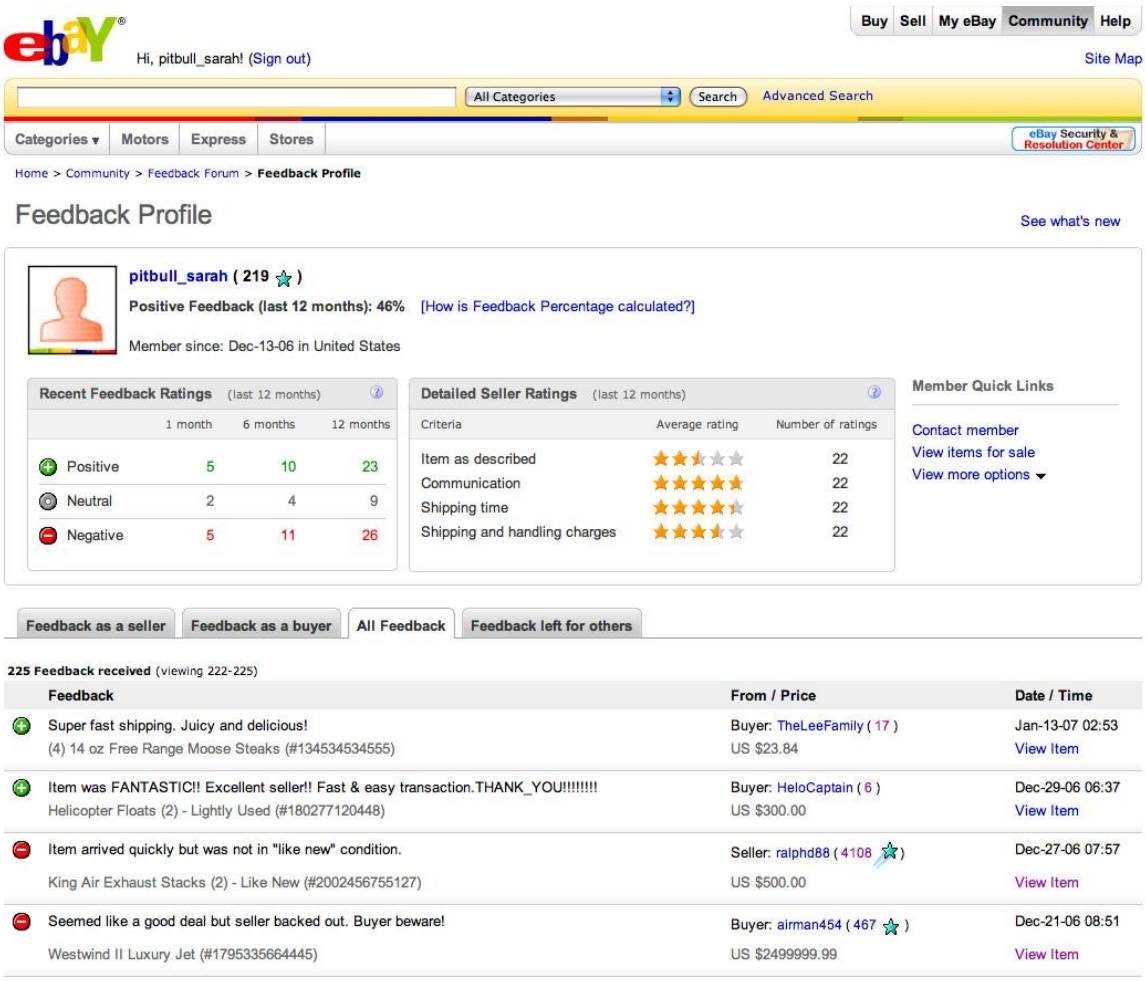
Από το Feedback Score που παρουσιάσαμε παραπάνω εξάγεται το αντίστοιχο Feedback star[4] που είναι το σύμβολο της εμπιστοσύνης και της εμπειρίας κάθε χρήστη στην κοινότητα του eBay. Το Feedback star ενός χρήστη φαίνεται στο προφίλ του. Κάθε άλλος χρήστης του eBay που επιθυμεί να ξεκινήσει μια συναλλαγή μαζί του (είτε ως πωλητή είτε ως αγοραστή) μπορεί να το συμβουλευτεί.

Εκτός όμως από το Feedback star υπάρχουν και άλλοι τρόποι αξιολόγησης για κάθε χρήστη που μπορούν να εντοπιστούν στην σελίδα του προφίλ του. Ένας τρόπος είναι η αξιολόγηση ενός πωλητή με βάση τα κριτήρια που περιγράψαμε προηγουμένως. Τέλος, οι χρήστες μπορούν να αφήνουν σχόλια στο προφίλ ενός χρήστη που έχουν

[3] Στο κριτήριο αυτό συμπεριλαμβάνεται και το κατά πόσο το αντικείμενο που παρέλαβε ο αγοραστής συμφωνεί με την περιγραφή που έδωσε ο πωλητής.

[4] Λεπτομερώς το Feedback star στην ιστοσελίδα http://pages.ebay.ie/help/feedback/scores-reputation.html

συναλλαχθεί μαζί του. Τα σχόλια αυτά μπορεί να αποδειχθούν χρήσιμα για επόμενους χρήστες που πρόκειται να ξεκινήσουν μια συναλλαγή με αυτόν. Στο σχήμα 2.5 φαίνεται η σελίδα του προφίλ ενός χρήστη του eBay. Μπορούμε ξεκάθαρα να διακρίνουμε τα τέσσερα σημεία αξιολόγησης που περιγράφονται στο σχήμα 2.6.



Σχήμα 2.5 : Feedback profile

Yellow star ( ) = 10 to 49 ratings

Blue star ( ) = 50 to 99 ratings

Turquoise star ( ) = 100 to 499 ratings

Purple star ( ) = 500 to 999 ratings

Red star ( ) = 1,000 to 4,999 ratings

Green star ( ) = 5,000 to 9,999 ratings

Yellow shooting star ( ) = 10,000 to 24,999 ratings

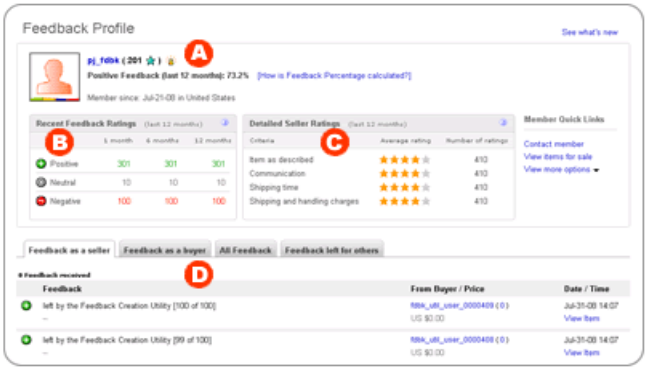
Turquoise shooting star ( ) = 25,000 to 49,999 ratings

Purple shooting star ( ) = 50,000 to 99,999 ratings

Red shooting star ( ) = 100,000 to 499,000 ratings

Green shooting star ( ) = 500,000 to 999,999 ratings

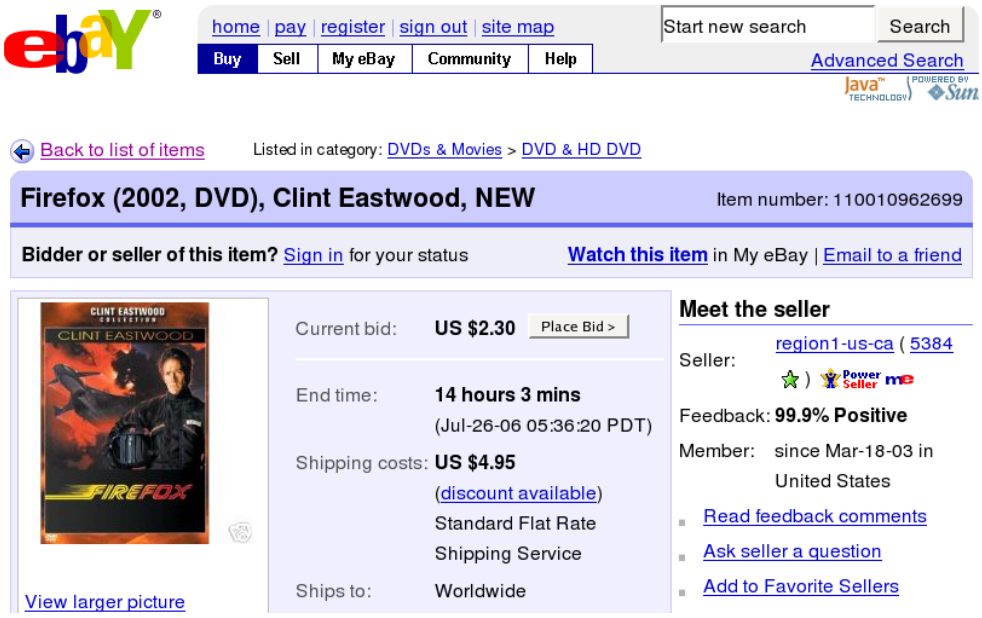
Silver shooting star ( ) = 1,000,000 ratings or more



Τι είναι ;
A: Θετικές αξιολογήσεις και Feedback star
B: Πρόσφατες αξιολογήσεις
C: Λεπτομερείς αξιολογήσεις πωλητή
D: Σχόλια άλλων χρηστών

Σχήμα 2.6 : Feedback score

Ουσιαστικά, στο e-bay εκτός από τους συμμετέχοντες (buyers and sellers) υπάρχει και μια υπηρεσία δημοπρασίας (auction service). Η υπηρεσία αυτή λειτουργεί ως ο μεσάζων ανάμεσα στον πωλητή και τον αγοραστή και πρόκειται για μια εφαρμογή λογισμικού που είναι τοποθετημένη στην ιστοσελίδα και «τρέχει» κατά την διάρκεια της πώλησης/αγοράς ενός αντικειμένου. Στο σχήμα 2.7 που ακολουθεί φαίνεται ένα στιγμιότυπο της υπηρεσίας δημοπρασίας του eBay κατά την δημοπρασία ενός αντικειμένου. Ο αγοραστής έχει την δυνατότητα να δει την αξιοπιστία του πωλητή προτού κάνει προσφορά.



Σχήμα 2.7 : Η υπηρεσία δημοπρασίας του eBay

Το eBay δίνει την δυνατότητα στους ίδιους τους αγοραστές να διαμορφώνουν την αξιοπιστία των πωλητών και των προϊόντων τους. Παρέχει, επίσης, την δυνατότητα στους πωλητές να δίνουν reviews και ενημέρωση για τα αντικείμενα που δημοπρατούν. Οι αγοραστές από την άλλη, μπορούν να ζητήσουν διευκρινήσεις από τους πωλητές σχετικά με το προϊόν που επιθυμούν να αγοράσουν, από τη στιγμή που αρχίσει η δημοπρασία και καθ' όλη τη διάρκεια της. Άλλωστε η σωστή επικοινωνία είναι το κλειδί για την ανάπτυξη της εμπιστοσύνης σε online δημοπρασίες. Η εμπιστοσύνη στο eBay στηρίζεται σε μεγάλο ποσοστό στις αξιολογήσεις που στέλνουν οι χρήστες του, γι' αυτό και ενθαρρύνει όλους τους αγοραστές και τους πωλητές να δίνουν feedback.

Η reputation πληροφορία κάθε χρήστη αποθηκεύεται σε κεντρικό server του eBay και απεικονίζεται στην σελίδα του προφίλ του, στην οποία έχουν πρόσβαση όλα τα εγγεγραμμένα μέλη. Όλοι οι αγοραστές και οι πωλητές ενθαρρύνονται να έχουν λογαριασμούς Pay Pal. Το Pay Pal είναι ο πιο κοινός και ασφαλής τρόπος πληρωμής. Η μέθοδος αυτή αποτρέπει αγνώστους να έχουν πρόσβαση σε τραπεζικούς λογαριασμούς άλλων χρηστών του eBay.

Το μοναδικό μειονέκτημα του eBay σε θέμα ασφάλειας είναι ότι σε αντίθεση με άλλες παρόμοιες υπηρεσίες δεν διαθέτει sniping πολιτική. Αυτό σημαίνει ότι οι αγοραστές μπορούν με την χρήση ενός *auction sniping* προγράμματος να κάνουν την υψηλότερη προσφορά την τελευταία δυνατή στιγμή και να κλέψουν τη δημοπρασία. Σύμφωνα, βέβαια, με το eBay, όλο αυτό είναι μέρος της εμπειρίας.

2.3 Κακόβουλες επιθέσεις σε P2P δίκτυα

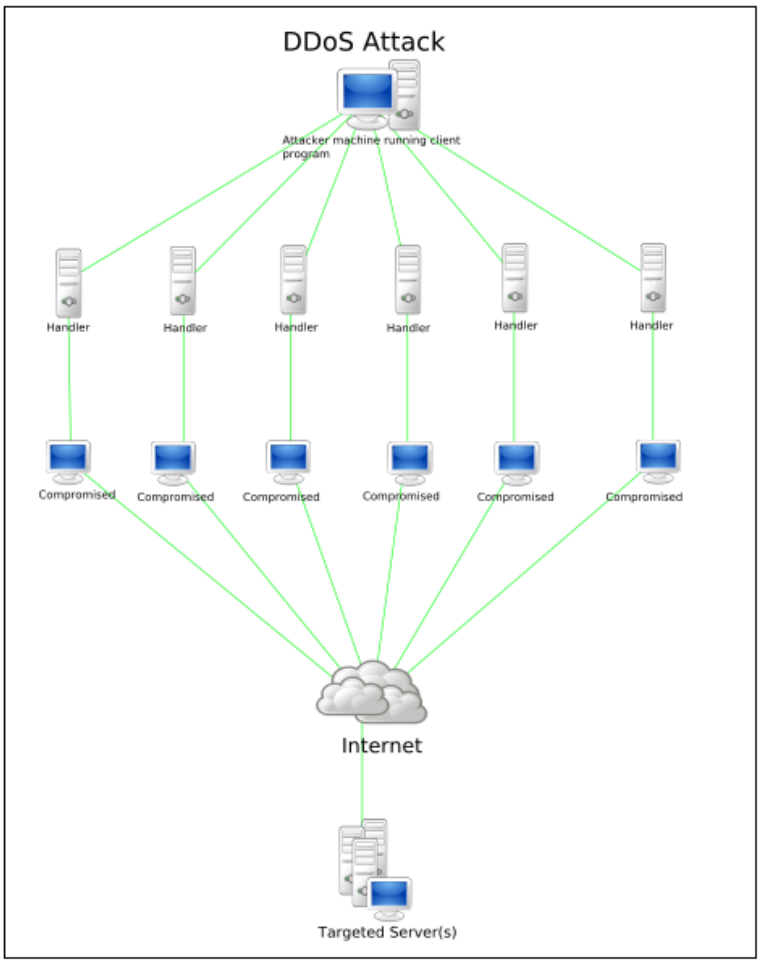
Η ευκολία χρήσης, η πληθώρα επιλογών και η χαμηλή τιμή (συχνά δωρεάν) είναι οι λόγοι που τα p2p δίκτυα εξαπλώνονται ραγδαία τα τελευταία χρόνια. Η έκρηξη δημοτικότητας των δικτύων αυτών, σε συνδυασμό με το γεγονός ότι αναπτύσσονται σε εχθρικά περιβάλλοντα, τα έχουν βάλει στον στόχο κακόβουλων χρηστών. Τα p2p δίκτυα αντιμετωπίζουν κάθε στιγμή τον κίνδυνο να δεχτούν επιθέσεις από hackers. Οι κακόβουλες επιθέσεις σε p2p δίκτυα μπορούν να κατηγοριοποιηθούν με πολλούς τρόπους. Οι επιθέσεις με βάση την προέλευση χωρίζονται σε εξωτερικές επιθέσεις, που είναι επιθέσεις που προέρχονται από κόμβους που δεν ανήκουν στο δίκτυο και σε εσωτερικές επιθέσεις, που είναι οι επιθέσεις που προέρχονται από κόμβους που ανήκουν στο ίδιο το δίκτυο. Οι επιθέσεις με βάση την φύση τους, ταξινομούνται σε ενεργητικές και παθητικές. Οι παθητικές επιθέσεις είναι εκείνες στις οποίες κάποιος ωτακουστής παρακολουθεί τις συναλλαγές μεταξύ δύο κόμβων του δικτύου, αλλά δεν υφίσταται αλλοίωση των δικτυακών δεδομένων και συστημάτων. Στις ενεργητικές επιθέσεις υπάρχει ενεργή αλληλεπίδραση με το θύμα, δηλαδή ο εισβολέας επικοινωνεί είτε διαγράφοντας δεδομένα, είτε αλλοιώνοντας δεδομένα ή ακόμα και μπλοκάροντας την επικοινωνία στο δίκτυο. Τέλος οι επιθέσεις χωρίζονται σε επιθέσεις στο επίπεδο δεδομένων και σε επιθέσεις στο επίπεδο ελέγχου, ανάλογα με τον στόχο της επίθεσης. Μια επίθεση στο επίπεδο δεδομένων έχει ως σκοπό να «δηλητηριάσει» τα αρχεία που διακινούνται στο δίκτυο ή ακόμα και να τα καταστήσει

μη διαθέσιμα. Μια επίθεση στο επίπεδο ελέγχου έχει ως στόχο την λειτουργικότητα του συστήματος με σαφή προσανατολισμό να καταστήσει το δίκτυο αργό και αναποτελεσματικό και με γνώμονα κυρίως το πρωτόκολλο δρομολόγησης. Όπως γίνεται αντιληπτό οι παραπάνω κατηγοριοποιήσεις δεν είναι ανεξάρτητες μεταξύ τους, αλλά μια επίθεση μπορεί να ανήκει ταυτόχρονα σε δύο ή και παραπάνω κατηγορίες. Στη συνέχεια αναλύουμε τις πιο σημαντικές επιθέσεις που μπορεί να δεχτεί ένα p2p δίκτυο, καθώς και προτεινόμενες άμυνες.

2.3.1 Επιθέσεις άρνησης υπηρεσίας

Οι επιθέσεις άρνησης υπηρεσίας (Denial of Service, DoS attacks) έχουν ως στόχο να αποτρέψουν τους χρήστες να χρησιμοποιήσουν το σύστημα, εκμεταλλευόμενες αδυναμίες του λογισμικού (software exploits). Η πιο κοινή μορφή επίθεσης DoS σε δίκτυα p2p είναι η επίθεση «πλημμύρας», όπου ο επιτιθέμενος πλημμυρίζει το δίκτυο με ψευδή πακέτα ή με queries ώστε να εξαντλήσει τους υπολογιστικούς ή δικτυακούς πόρους. Οι επιθέσεις DoS είναι πιο επικίνδυνες όταν οι επιτιθέμενοι χρήστες είναι πολλαπλοί. Τότε αναφέρονται ως κατανεμημένες επιθέσεις άρνησης υπηρεσίας (Distributed Denial-of-Service, DDoS). Σε μια DDoS επίθεση, όπως φαίνεται στο σχήμα 2.8, γίνεται χρήση, από τον επιτιθέμενο, πολλών ελεγχόμενων υπολογιστών. Οι υπολογιστές αυτοί (compromised) εκτίθενται σε κάποιον ιό ή trojan και τους εγκαθίσταται ένας zombie agent. Έπειτα τα τερματικά που χειρίζεται ο επιτιθέμενος (handlers ή bots) καθοδηγούν τους zombie υπολογιστές σε μια κατανεμημένη επίθεση στο στοχοθετημένο εξυπηρετητή ή και δίκτυο. Σε κάθε περίπτωση οι handlers κάνουν IP spoofing, ώστε να κρύβουν την IP διεύθυνση (και συνεπώς την προέλευση) των zombie υπολογιστών. Άλλες γνωστές DoS επιθέσεις είναι οι : Ping-of-Death (POD), Teardrop, SYN attack, Smurf attack και Amplification Attack.

Τα p2p δίκτυα είναι στόχοι πολλών εισβολέων που θέλουν να ξεκινήσουν μια επίθεση DOS σε κάποια ιστοσελίδα παραδείγματος χάριν. Ο επιτιθέμενος μπορεί να αποσυνδέσει τους χρήστες ενός μεγάλου p2p file sharing hub και να τους συνδέσει στην ιστοσελίδα του θύματος προκαλώντας υψηλή συμφόρηση.



Σχήμα 2.8 : DDoS επίθεση

Προτεινόμενες άμυνες

Εξαιτίας του ότι οι υπολογιστές που εκτελούν μια επίθεση DoS κρύβουν την IP διεύθυνση τους και ενδέχεται να είναι διαφορετικοί μεταξύ τους, οι επιθέσεις DoS είναι πολύ δύσκολο να αντιμετωπιστούν. Μια ευρέως γνωστή μέθοδος για την αποφυγή επιθέσεων DoS είναι το *pricing*. Ο host προτού προχωρήσει στην περάτωση των αιτημάτων, ζητάει από τους χρήστες να λύσουν κάποιον εύκολο γρίφο ή να πληκτρολογήσουν ένα συνθηματικό που τους δίνεται. Επιθέσεις μικρής εμβέλειας μπορούν να αποτραπούν με αυτή την τεχνική, όμως επιθέσεις μεγάλης εμβέλειας όπως οι DDoS καταφέρνουν τον στόχο τους.

2.3.2 Σιβυλλικές επιθέσεις

Η δημιουργία της σιβυλλικής επίθεσης (Sybil attack) έγινε με σκοπό να νικήσει τους μηχανισμούς πλεονασμού των κατανεμημένων συστημάτων αποθήκευσης δεδομένων στα peer-to-peer δίκτυα. Η ελλοχεύουσα ιδέα της επίθεσης αυτή είναι πως ένας κακόβουλος κόμβος μπορεί να παρουσιάσει πολλαπλές ταυτότητες (είτε καινούριες είτε κλεμμένες) και έτσι να κερδίσει τον έλεγχο ενός μέρους του δικτύου. Μόλις το επιτύχει αυτό, ο επιτιθέμενος μπορεί να πάρει στην δικαιοδοσία του αρχεία και να αποφασίσει είτε να τα μολύνει είτε να τα αλλοιώσει. Εάν ο επιτιθέμενος μπορεί να τοποθετήσει τις πολλαπλές ταυτότητές του με στρατηγικό τρόπο, μπορεί να καταφέρει μεγάλη βλάβη στο σύστημα.

Προτεινόμενες άμυνες

Ένας τρόπος αντιμετώπισης των σιβυλλικών επιθέσεων είναι, όπως αναφέραμε και στις DoS επιθέσεις, το pricing. Με αυτόν τον τρόπο, ο κακόβουλος χρήστης δεν μπορεί να δημιουργήσει αυτόματα και αυθαίρετα πολλαπλές ταυτότητες, γεγονός χρονοβόρο και επίπονο για κάποιον επιτιθέμενο που θέλει να δημιουργήσει χιλιάδες ταυτότητες χρηστών.

Μια ακόμα λύση είναι η ύπαρξη μιας κεντρικής αρχής με την οποία θα πρέπει να έρθει σε επικοινωνία ο χρήστης ώστε να πιστοποιήσει την ταυτότητα του. Ενώ μια τέτοια περίπτωση φαντάζει ως ιδανική λύση για το πρόβλημα αυτό, παρόλα αυτά είναι μη ικανοποιητική, καθώς η ύπαρξη οποιουδήποτε κεντροποιημένου κόμβου αναιρεί αυτόματα το μοντέλο p2p.

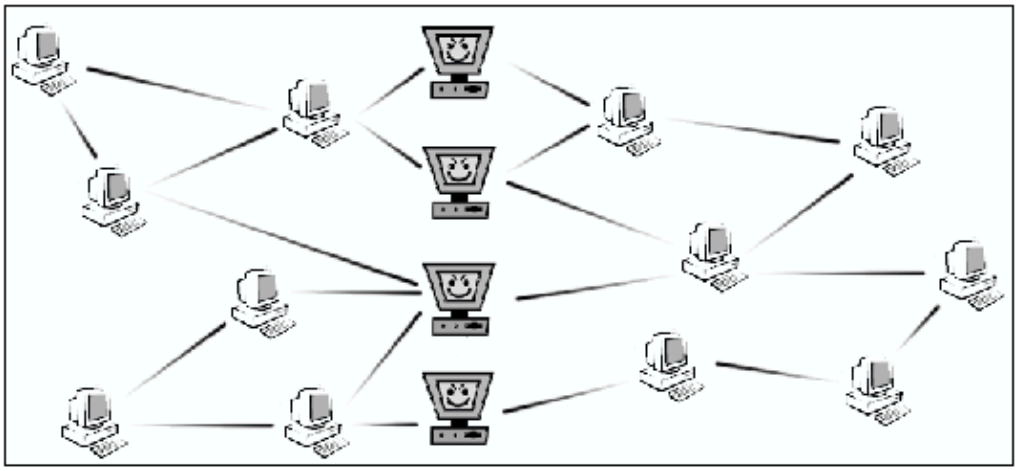
Τέλος, η καλύτερη άμυνα για να αποτραπούν οι σιβυλλικές επιθέσεις είναι με το να καταστεί αδύνατο στον επιτιθέμενο να τοποθετήσει τις πολλαπλές ταυτότητες του σε στρατηγικές θέσεις. Γι' αυτόν τον λόγο τα δομημένα p2p δίκτυα είναι ο καλύτερος αμυντικός μηχανισμός απέναντι στις σιβυλλικές επιθέσεις. Δομημένα δίκτυα όπως τα Chord, CAN, Pastry και πιο σύγχρονα όπως το Freenet δεν μπορούν να αποτρέψουν κάποιον να δημιουργεί πολλαπλές ταυτότητες. Όμως κάνουν χρήση αποδοτικών και τυχαία δυναμικών αλγορίθμων για την εύρεση των χρηστών του δικτύου στηριζόμενοι στους κατανεμημένους πίνακες κατακερματισμού DHT. Έτσι δεν αφήνουν κακόβουλους χρήστες να κατακτήσουν στρατηγικές θέσεις και να απομονώσουν το υπόλοιπο δίκτυο. Οι τυχαία κατανεμημένες ταυτότητες περιορίζουν τις επιπτώσεις μιας σιβυλλικής επίθεσης.

2.3.3 Man-in-the-Middle επιθέσεις

Στις Man-in-the-Middle επιθέσεις, ο επιτιθέμενος παρεμβάλλεται χωρίς να ανιχνευθεί στην επικοινωνία δύο κόμβων και μπορεί είτε να υποκλέψει στοιχεία είτε να υποκριθεί ότι είναι κάποιος τρίτος. Ευτυχώς, τέτοιες επιθέσεις δεν έχουν επιπτώσεις σε p2p δίκτυα, όπου όλοι οι κόμβοι είναι ομότιμοι και τα περιεχόμενα των συναλλαγών μοιράζονται έτσι κι αλλιώς. Σκοπός τέτοιων επιθέσεων θα μπορούσε να είναι η μόλυνση αρχείων και η διάδοσή τους στο δίκτυο εξ' ονόματος έμπιστων οντοτήτων ή η αναμετάδοσή τους από κάποιον supernode. Παραδείγματος χάριν, ARP poisoning, TCP session hijacking, DNS poisoning, URL redirection.

2.3.4 Επιθέσεις Eclipse

Προτού ο επιτιθέμενος ξεκινήσει μια επίθεση Eclipse θα πρέπει να πάρει υπό τον έλεγχο του έναν συγκεκριμένο αριθμό κόμβων του δικτύου κατά μήκος στρατηγικών μονοπατιών δρομολόγησης (όπως ακριβώς και στην Sybil επίθεση). Αφού επιτύχει αυτό, μπορεί να χωρίσει το δίκτυο σε διάφορα υποδίκτυα. Κατά συνέπεια, όταν ένας κόμβος θέλει αν επικοινωνήσει με έναν κόμβο άλλου υποδικτύου, το μήνυμά του θα πρέπει να δρομολογηθεί μέσω ενός κακόβουλου κόμβου. Οι επιθέσεις Eclipse είναι μια κλίμακα υψηλότερα από τις Man-in-the-Middle επιθέσεις και κατά κάποιο τρόπο αποτελούν συνέχεια των σιβυλλικών επιθέσεων.



Σχήμα 2.9 : Επίθεση Eclipse

Το να καταφέρει κάποιος κακόβουλος χρήστης μια επίθεση Eclipse είναι εξαιρετικά δύσκολο. Αν όμως το επιτύχει, τότε μπορεί να βλάψει το δίκτυο με ιδιαίτερα αποτελεσματικό τρόπο :

- Ο επιτιθέμενος μπορεί να επιτεθεί στο επίπεδο ελέγχου δρομολογώντας ανεπαρκώς κάθε μήνυμα.
- Ο επιτιθέμενος μπορεί να απορρίψει όλα τα μηνύματα, ώστε να αποκόψει κάθε επικοινωνία μεταξύ των υποδικτύων.
- Ο επιτιθέμενος μπορεί να επιτεθεί στο επίπεδο δεδομένων μολύνοντας αρχεία ή μεταφορτώνοντας εκ μέρους αθώων κόμβων μολυσμένα ή αλλοιωμένα αρχεία.

2.3.5 Κακόβουλο λογισμικό

Όταν αναφερόμαστε σε κακόβουλο λογισμικό εννοούμε κυρίως ιούς, δούρειους ίππους (trojans) και σκουλήκια (worms). Τα πρώτα δύο προσαρτώνται σε εκτελέσιμα αρχεία και συνήθως έχουν αργή μετάδοση. Τα σκουλήκια είναι αυτόματα διαδιδόμενοι ιοί που χρησιμοποιούν το δίκτυο για να στέλνουν αντίγραφα του εαυτού τους σε άλλους κόμβους του δικτύου και συνεπώς δεν χρειάζονται την μεσολάβηση κάποιου χρήστη. Σε αντίθεση με τους ιούς, τα σκουλήκια δεν χρειάζεται να επισυναφτούν σε κάποιο εκτελέσιμο πρόγραμμα.
Όπως καταλαβαίνουμε, τα σκουλήκια αποτελούν έναν από τους μεγαλύτερους κινδύνους στο διαδίκτυο. Η διάδοση των σκουληκιών μέσω ενός p2p δικτύου είναι καταστροφική. Τα σκουλήκια όπως φαίνεται προσελκύονται από τα p2p δίκτυα για τους εξής λόγους :

- Τα p2p δίκτυα αποτελούνται από υπολογιστές που τρέχουν παρόμοιο λογισμικό. Έτσι ένας επιτιθέμενος μπορεί να ελέγξει όλο το δίκτυο εκμεταλλευόμενος μόνο μία τρύπα ασφαλείας.

- Οι ομότιμοι κόμβοι ενός p2p δικτύου διασυνδέονται με πολλούς διαφορετικούς κόμβους. Ένα σκουλήκι που τρέχει σε μια p2p εφαρμογή δεν θα χάνει χρόνο να ψάχνει για άλλα θύματα. Απλά θα παίρνει την λίστα με τους γείτονες του θύματος του και θα εξαπλώνεται.

- Οι p2p εφαρμογές συνήθως χρησιμοποιούνται για την ανταλλαγή μεγάλων αρχείων. Τα σκουλήκια συνήθως είναι τόσο μικρά ώστε να χωράνε σε ένα TCP πακέτο. Συνεπώς τα p2p σκουλήκια δεν θα έχουν περιορισμό μεγέθους, άρα θα μπορούν να επιτελούν πιο πολύπλοκες λειτουργίες.

- Τα p2p προγράμματα τρέχουν κυρίως σε τερματικά και όχι σε εξυπηρετητές. Συνεπώς, είναι πιο πιθανό για έναν επιτιθέμενο να αποκτήσει πρόσβαση σε ευαίσθητα αρχεία, όπως αριθμοί πιστωτικών καρτών, κωδικούς και βιβλία διευθύνσεων.

Προτεινόμενες άμυνες

Όλα τα προβλήματα ξεκινούν από έναν αφύλακτο υπολογιστή σε μια ευρυζωνική σύνδεση στο διαδίκτυο, χωρίς εγκατεστημένο κάποιο τείχος προστασίας και κάποιο πρόγραμμα antivirus. Μια λύση θα ήταν οι υπεύθυνοι για την ανάπτυξη p2p λογισμικού να μην γράφουν λογισμικά που να περιέχουν bugs. Από μόνο του αυτό είναι ανέφικτο, αλλά ένα πρώτο βήμα θα ήταν να ευνοηθούν γλώσσες με ισχυρό σύστημα τύπων όπως η Java και η C# αντί της C ή της C++, όπου οι υπερχειλίσεις του buffer είναι πολύ ευκολότερο να υπολογιστούν.
Μια άλλη ενδιαφέρουσα παρατήρηση είναι ότι τα υβριδικά p2p συστήματα εμφανίζουν μια ευπάθεια, ενώ τα αμιγή p2p συστήματα όχι. Η ύπαρξη supernodes στο δίκτυο Gnutella για παράδειγμα, δίνει την δυνατότητα στον επιτιθέμενο να στοχεύει αυτούς τους στρατηγικούς κόμβους πρώτα, προκειμένου να διαδοθεί το σκουλήκι πιο αποτελεσματικά.
Τέλος, οι υπεύθυνοι για την ανάπτυξη λειτουργικών συστημάτων προσφέρουν επίσης μερικές λύσεις. Το λειτουργικό OpenBSD 3.8 επιστρέφει ψευδοτυχαίες διευθύνσεις μνήμης. Αυτό κάνει τις υπερχειλίσεις του buffer αδύνατες, δεδομένου ότι ένας επιτιθέμενος δεν μπορεί να ξέρει ποιο τμήμα μνήμης πρέπει να αντιγράψει.

2.4 Ανασκόπηση και σύγκριση μοντέλων εμπιστοσύνης

Από τα μοντέλα που παρουσιάσαμε στο προηγούμενο κεφάλαιο αυτό που ξεχωρίζει, με την έννοια του διαφορετικού, είναι το eBay. Καταρχήν, έχει εφαρμογή σε e-commerce κοινότητες, όπου υπάρχει κεντρική αποθήκευση των πληροφοριών φήμης. Επίσης, η υπηρεσία δημοπρασίας που λειτουργεί ως ο μεσάζων μεταξύ πωλητή και αγοραστή, βρίσκεται με την μορφή λογισμικού σε κεντρικό εξυπηρετητή και «τρέχει» τοπικά σε κάθε χρήστη κάθε φορά που ο τελευταίος επιθυμεί να συμμετάσχει σε μια συναλλαγή. Σε θέματα ασφαλείας και αξιοπιστίας, το eBay είναι αψεγάδιαστο σε σύγκριση με παρόμοια μοντέλα για p2p δίκτυα ηλεκτρονικού εμπορίου με δημοπρασία. Οι αγορές γίνονται με ασφάλεια κάτι που το εγγυάται το ίδιο το σύστημα, ενώ κακόβουλοι χρήστες απομονώνονται από τους υπόλοιπους χρήστες.

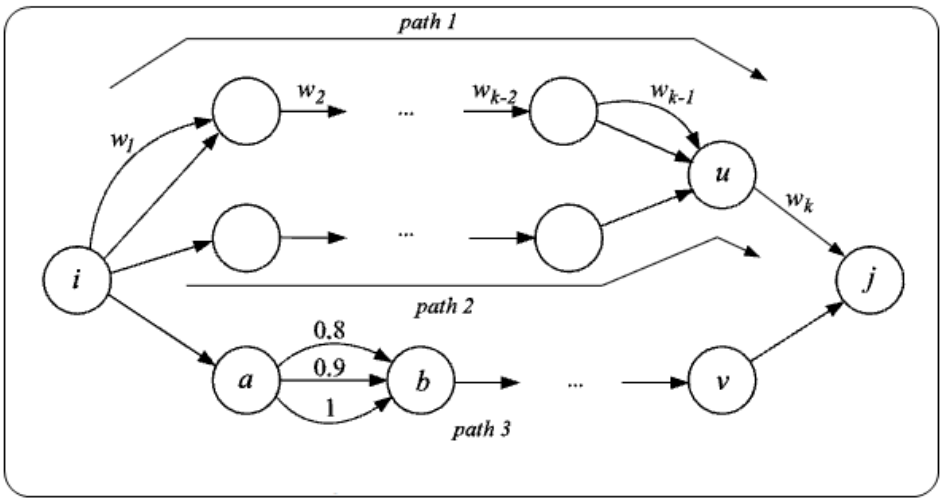
Το EigenTrust είναι ένα μοντέλο διαχείρισης της εμπιστοσύνης βασισμένο στην φήμη, για δομημένα p2p δίκτυα. Ο αλγόριθμος του μοντέλου παρέχει σε κάθε χρήστη του δικτύου μια μοναδική καθολική τιμή εμπιστοσύνης που βασίζεται στο ιστορικό των μεταφορτώσεων και με αυτόν τον τρόπο στοχεύει στην ελάττωση του αριθμού των μη γνήσιων αρχείων σε ένα p2p δίκτυο. Οι καθολικές τιμές εμπιστοσύνης έχουν διπλό στόχο στο EigenTrust. Πρώτον, να απομονώσουν του κακόβουλους κόμβους και δεύτερον να δώσουν κίνητρο στους νέους χρήστες να μοιράζονται τα αρχεία τους. Οι δημιουργοί του EigenTrust, είχαν υλοποιήσει και προσομοιώσει τον αλγόριθμο κάτω από διάφορες κακόβουλες απειλές. Τα συμπεράσματα που βγήκαν ήταν πολύ ενθαρρυντικά, καθώς για διάφορες τιμές κακόβουλων uploads, τα downloads μη γνήσιων αρχείων ήταν κατά 10% ελαττωμένα σε σύγκριση με ένα μοντέλο που δεν κάνει χρήση καθολικών τιμών εμπιστοσύνης.

Το ROCQ είναι και αυτό ένα μοντέλο διαχείρισης εμπιστοσύνης βασισμένο στην φήμη, για δομημένα p2p δίκτυα. Ο μηχανισμός ROCQ υπολογίζει την αξιοπιστία των κόμβων με βάση το ιστορικό των συναλλαγών τους (feedback). Γι' αυτόν τον σκοπό συνδυάζει τέσσερις μεταβλητές : την *φήμη ή καθολική τιμή εμπιστοσύνης* του κόμβου, την *γνώμη* που σχηματίζεται από τις άμεσες αλληλεπιδράσεις του κόμβου, την *αξιοπιστία* ενός κόμβου που δίνει συστάσεις και την *ποιότητα ή εγκυρότητα* που δίνει στις συστάσεις του ένας κόμβος. Τα συμπεράσματα που βγαίνουν από τις προσομοιώσεις του αλγόριθμου ROCQ είναι καταπληκτικά. Όταν οι κακόβουλοι κόμβοι είναι μειοψηφία στο σύστημα, τότε ο αλγόριθμος σε σχεδόν 100% των περιπτώσεων κάνει σωστή επιλογή (επιτρέπει τις αλληλεπιδράσεις με τίμιους

κόμβους, ενώ αποτρέπει τις αλληλεπιδράσεις με κακόβουλους κόμβους). Όταν οι κακόβουλοι κόμβοι αποτελούν την πλειοψηφία, τότε ο αλγόριθμος επιστρέφει σωστά αποτελέσματα σε 25-30% των περιπτώσεων. Το ποσοστό αυτό υπερβαίνει την απόδοση άλλων αλγόριθμων στις ίδιες συνθήκες.

Τέλος, είδαμε ένα Bayesian μοντέλο διαχείρισης εμπιστοσύνης βασισμένο στην φήμη. Το μοντέλο Bayesian που παρουσιάσαμε αφορούσε file sharing εφαρμογές. Τα Bayesian δίκτυα κάνουν χρήση πιθανολογικής εκτίμησης της εμπιστοσύνης, σύμφωνα με τον κανόνα του Bayes. Επιπλέον, παρέχουν ελαστικές μεθόδους για την παρουσίαση της διαφοροποιημένης εμπιστοσύνης και τον συνδυασμό των πολλών διαφορετικών πτυχών της εμπιστοσύνης. Όπως και τα υπόλοιπα μοντέλα, έτσι αυτό δεν μπορεί να επεκταθεί στα αδόμητα p2p δίκτυα που χρησιμοποιούνται στο εμπόρια κατά κόρον σήμερα, αλλά «χτίστηκε» με βάση τα ερευνητικά δομημένα p2p δίκτυα. Σύμφωνα με διάφορες υλοποιήσεις του αλγορίθμου που έχουν γίνει από τους δημιουργούς του, το πιο σημαντικό στοιχείο του Bayesian μοντέλου είναι οι συστάσεις που παρέχουν οι κόμβοι. Η διάδοσή τους στο σύστημα, αυξάνει τον αριθμό των επιτυχημένων αλληλεπιδράσεων. Κλείνοντας, να επισημάνουμε, ότι το μοντέλο Bayesian είναι άριστα εφαρμόσιμο σε μικρά δίκτυα, όπου η πιθανότητα αγοραστή και πωλητή να έχουν συνεχόμενες αλληλεπιδράσεις είναι μεγάλη (φαινόμενο μικρού κόσμου). Αντίθετα, σε μεγάλα δίκτυα, όπως το eBay, όπου η πιθανότητα ένας πωλητής να συναντήσει τον ίδιο αγοραστή είναι μικρή, το Bayesian μοντέλο δεν δίνει ικανοποιητικά αποτελέσματα.

Από την παρουσίαση όλων των παραπάνω συστημάτων διαχείρισης της εμπιστοσύνης που είναι βασισμένα στην φήμη, μπορούμε να εξάγουμε την γενική μορφή ενός p2p συστήματος εμπιστοσύνης και τον θεωρητικό ορισμό ενός reputational συστήματος εμπιστοσύνης. Ας θεωρήσουμε τον πολύγραφο (multigraph) εμπιστοσύνης του σχήματος 2.10. Πρόκειται για έναν κατευθυνόμενο γράφο που αναπαριστά τα βάρη W των δεδομένων feedback από τις αλληλεπιδράσεις των κόμβων. Οι κόμβοι του γράφου είναι οι ομότιμοι κόμβοι του δικτύου και οι γραμμές, οι αλληλεπιδράσεις του κάθε κόμβου. Ο καταναλωτής είναι ο κόμβος από τον οποίο ξεκινάει η γραμμή και ο προμηθευτής ο κόμβος στον οποίο καταλήγει. Παραδείγματος χάριν, ο κόμβος a έχει αξιολογήσει τον κόμβο b στις τρεις αλληλεπιδράσεις που είχαν με 0.8, 0.9 και 1 αντίστοιχα (από το σύνολο τιμών [0,1]).



Σχήμα 2.10 : Πολύγραφος εμπιστοσύνης ενός p2p δικτύου

Για την αποτίμηση της εμπιστοσύνης ενός κόμβου χρειαζόμαστε έναν αλγόριθμο A που ενεργεί στον διαμορφωμένο γράφο, συναθροίζει τα διαθέσιμα feedback για κάθε κόμβο και εξάγει μια τιμή εμπιστοσύνης $t \in T$. Η τετράδα (G, W, A, T) αναπαριστούν ένα p2p σύστημα φήμης, όπου G ένας κατευθυνόμενος γράφος με βάρη (P, V) με P το σύνολο των ομότιμων κόμβων και V το σύνολο των αλληλεπιδράσεων.

3 Η εμπιστοσύνη σε συγκεντρωμένα δίκτυα

3.1 Γενικά

Όταν αναφερόμαστε στην εμπιστοσύνη ή στην ασφάλεια σε συγκεντρωμένα δίκτυα υπολογιστών, το πρώτο πράγμα που μας έρχεται στο μυαλό είναι η κεντρική αρχή. Η κεντρική αρχή είναι μια κεντροποιημένη οντότητα ή αλλιώς ένας εξυπηρετητής (server) που έχει αναλάβει να αποκαταστήσει με ασφάλεια την επικοινωνία δύο οντοτήτων. Παραδείγματος χάριν, οι δύο οντότητες αυτές μπορεί να είναι δύο εφαρμογές e-mail που θέλουν να ανταλλάξουν με ασφάλεια e-mail. Η κεντρική αρχή στην περίπτωση αυτή είναι ένας mail server ή όπως καλείται mail transfer agent (MTA) που αναλαμβάνει να δρομολογήσει με ασφάλεια τα e-mails από τον έναν χρήστη στον άλλον, με βάση τα MX records που έχει αποθηκευμένα.

Όπως και στα κατανεμημένα δίκτυα p2p, έτσι και στα συγκεντρωμένα δίκτυα, η ανάγκη για επικοινωνία μεταξύ των κόμβων που τα απαρτίζουν είναι επιτακτική. Όπως είδαμε και στο Κεφάλαιο 1, η βασική διαφορά είναι ότι στα μεν p2p δίκτυα, οι κόμβοι μπορούν να αναπτύξουν κατευθείαν επικοινωνία μεταξύ τους, ενώ στα συγκεντρωμένα δίκτυα κυριαρχούν ένας ή περισσότεροι κεντρικοί κόμβοι. Για να αποκτήσει πρόσβαση ένας κόμβος στους πόρους ενός άλλου κόμβου πρέπει να πάρει την έγκριση της κεντρικής αρχής. Παρότι το μοντέλο αυτό φαντάζει ευκολότερα υλοποιήσιμο και πιο ασφαλές, πλέον με την πάροδο των χρόνων εγκαταλείπεται από πολλές εφαρμογές του διαδικτύου, καθώς η ύπαρξη κεντρικών σημείων βλάβης το καθιστά ιδιαίτερα ευάλωτο. Ειδικότερα, κακόβουλες επιθέσεις, όπως για παράδειγμα επιθέσεις άρνησης υπηρεσίας (denial-of-service, DoS) στοχεύουν στις κεντρικές αρχές, ώστε να επιτύχουν ολική κατάρρευση του συστήματος.

Το κυριότερο χαρακτηριστικό για την ανάπτυξη εμπιστοσύνης σε ένα συγκεντρωμένο δίκτυο είναι η ύπαρξη ασφαλούς επικοινωνίας μεταξύ των κόμβων του. Οι επιθυμητές ιδιότητες ασφαλούς επικοινωνίας είναι οι εξής :

- **Εμπιστευτικότητα (Confidentiality) :** Η πληροφορία του μεταδιδόμενου μηνύματος είναι προσβάσιμη μόνο από τα εξουσιοδοτημένα μέλη. Δηλαδή από τον αποστολέα και τον σκοπούμενο δέκτη. Τα περιεχόμενα του μηνύματος δεν πρέπει να είναι κατανοητά σε τρίτους, όπως ωτακουστές που μπορεί να κάνουν παρεμβολές στο μήνυμα. Είναι φανερή, λοιπόν, η ανάγκη κρυπτογράφησης και αποκρυπτογράφησης των δεδομένων που στέλνονται κατά την επικοινωνία των δύο οντοτήτων. Οι τεχνικές κρυπτογράφησης που εξασφαλίζουν εμπιστευτικές επικοινωνίες βασίζονται σε ένα ή περισσότερα κλειδιά και παρουσιάζονται στην Ενότητα 3.2.

- **Ακεραιότητα (Integrity) :** Τα μεταδιδόμενα δεδομένα από την ώρα που ο αποστολέας τα δημιούργησε και τα υπέγραψε, μπορεί να τροποποιηθούν ακούσια ή εκούσια. Γι' αυτόν τον λόγο υπάρχουν τεχνικές ανίχνευσης αλλοιώσεων, παρόμοιες με τις τεχνικές αθροίσματος ελέγχου (CRC), που συναντάμε συχνά στα πρωτόκολλα ζεύξης δεδομένων. Οι τεχνικές αυτές ονομάζονται αλγόριθμοι σύνοψης μηνυμάτων (message digest) και παρέχουν την επιθυμητή ακεραιότητα της πληροφορίας.

- **Μη απάρνηση (Non-repudiation) :** Μη απάρνηση είναι η γενική ιδέα ότι ένα συμβόλαιο δεν μπορεί να αμφισβητηθεί από κανένα από τα συμβαλλόμενα μέρη. Στα δίκτυα η μη απάρνηση σημαίνει ότι μπορεί να πιστοποιηθεί πως αποστολέας και παραλήπτης ήταν όντως εκείνοι που έστειλαν και έλαβαν το μήνυμα αντίστοιχα. Κανείς από τους δύο δεν μπορεί να αρνηθεί την αυθεντικότητα της μετάδοσης ή την δημιουργία της.

- **Πιστοποίηση (Authentication) :** Στις πρόσωπο με πρόσωπο ανθρώπινες επικοινωνίες τα δύο επικοινωνούντα μέρη μπορούν με την οπτική επαφή να αναγνωρίσουν το ένα το άλλο. Όταν όμως αναφερόμαστε σε δίκτυα, όπου οι επικοινωνούσες οντότητες χωρίζονται από ένα μέσο αυτό είναι δύσκολο. Έτσι,

αποστολέας και παραλήπτης πρέπει να είναι σε θέση να εξακριβώσουν τις ταυτότητές τους. Στην Ενότητα 3.3 θα δούμε όλα τα γνωστά πρωτόκολλα ταυτοποίησης.

- **Διαθεσιμότητα και έλεγχος προσπέλασης :** Όταν αναφερόμαστε στον όρο διαθεσιμότητα εννοούμε κατά πόσο είναι δυνατόν να επιτευχθεί η επικοινωνία και ότι κακόβουλοι χρήστες δεν θα αποτρέψουν τη χρήση του δικτύου υποδομής για την επικοινωνία έγκυρων χρηστών. Τα τελευταία χρόνια αυξάνονται ολοένα οι επιθέσεις άρνησης υπηρεσίας (DoS) που σκοπό έχουν να καταστήσουν άχρηστο το δίκτυο για επικοινωνία των έγκυρων χρηστών. Η ύπαρξη έγκυρων και κακόβουλων χρηστών οδήγησε και στην ιδέα του ελέγχου προσπέλασης, δηλαδή την επιβεβαίωση των δικαιωμάτων του χρήστη ότι επιτρέπεται να προσπελάσει τους πόρους και ότι η προσπέλαση αυτή γίνεται με τον σωστά καθορισμένο τρόπο. Η αντιμετώπιση τέτοιων προβλημάτων γίνεται με την χρήση τειχών προστασίας (firewalls). Το firewall είναι μια συσκευή που τοποθετείται ανάμεσα στο δίκτυο που θέλουμε να προστατεύσουμε και στον υπόλοιπο κόσμο. Η μελέτη τέτοιων θεμάτων ξεφεύγει από το ζήτημα μας και δεν θα αναλυθεί περαιτέρω.

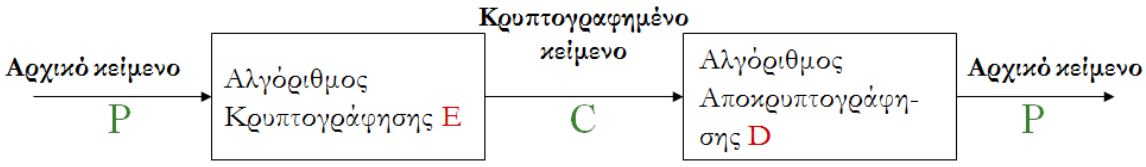
Για να αναπτυχθεί, λοιπόν, εμπιστοσύνη μεταξύ δύο οντοτήτων χρειάζονται τρία στοιχεία. Κατά πρώτον, να υπάρχει *εμπιστευτικότητα,* κάτι που επιτυγχάνεται με την κρυπτογράφηση των δεδομένων. Δεύτερον, η *πιστοποίηση* των δύο συμβαλλόμενων οντοτήτων, δηλαδή η εξακρίβωση των ταυτοτήτων τους. Τέλος, να υπάρχει *ακεραιότητα* των δεδομένων που ανταλλάσσουν. Στη συνέχεια αναλύουμε ένα προς ένα όσα αναφέρθηκαν παραπάνω, με σκοπό στο τέλος του κεφαλαίου να παρουσιάσουμε το σύστημα Kerberos που αναπτύχθηκε το 1994 στο MIT και σχεδιάστηκε με απώτερο σκοπό την ασφαλή και έμπιστη σύνδεση ενός χρήστη στους πόρους μιας υπηρεσίας. Ο χρήστης αυτός έχει πιστοποιηθεί στην κεντρική αρχή του Kerberos. Επομένως, σε ένα τέτοιο σύστημα η κεντρική αρχή αναλαμβάνει να καθορίσει την εμπιστοσύνη στο δίκτυο και δεν την κερδίζει ο χρήστης, όπως συμβαίνει στα p2p δίκτυα.

3.2 Αρχές κρυπτογραφίας

Η κρυπτογραφία είναι ένας κλάδος της επιστήμης της κρυπτολογίας, η οποία ασχολείται με την μελέτη της ασφαλούς επικοινωνίας. Ο κύριος στόχος της είναι να παρέχει μηχανισμούς για δύο ή περισσότερα μέλη να επικοινωνήσουν, χωρίς κάποιος άλλος να είναι ικανός να διαβάσει την πληροφορίας εκτός από τα συμβαλλόμενα μέρη. Συνεπώς, ένας αποστολέας θα μπορεί να κρυπτογραφήσει τα δεδομένα προς μετάδοση και ένας δέκτης θα πρέπει να είναι σε θέση να μπορεί να αποκρυπτογραφήσει τα «μεταμφιεσμένα» δεδομένα.
Κρυπτογράφηση ονομάζεται η διαδικασία μετασχηματισμού ενός **αρχικού κειμένου** (plaintext ή cleartext) σε μια ακατάληπτη μορφή με την χρήση κάποιου **αλγόριθμου κρυπτογράφησης**, ούτως ώστε να μην μπορεί να διαβαστεί από κανέναν εκτός του νόμιμου παραλήπτη. Το ακατανόητο κείμενο που προκύπτει είναι το κρυπτογραφημένο μήνυμα, γνωστό σαν **κρυπτοκείμενο** (ciphertext). Ένα ενδιαφέρον στοιχείο είναι πως οι αλγόριθμοι κρυπτογράφησης που χρησιμοποιούνται στα περισσότερα σύγχρονα συστήματα (και αυτών του Διαδικτύου) έχουν εκδοθεί, έχουν προτυποποιηθεί και είναι διαθέσιμοι στον καθέναν. Εφόσον όλοι ξέρουν την μέθοδο κρυπτογράφησης θα πρέπει να υπάρχει μια μυστική πληροφορία που να αποτρέπει έναν εισβολέα να κατανοήσει το περιεχόμενο του κρυπτογραφημένου μηνύματος. Αυτή η μυστική πληροφορία είναι τα περιβόητα κλειδιά. Ένα **κλειδί** (key) είναι ένας αριθμός αρκετών bit που χρησιμοποιείται ως είσοδος στην συνάρτηση κρυπτογράφησης.

Ένα κρυπτοσύστημα (σύνολο διαδικασιών κρυπτογράφησης-αποκρυπτογράφησης) αποτελείται από μία πεντάδα (P, C, k, E, D) :

- Το P είναι ο χώρος όλων των δυνατών μηνυμάτων ή αλλιώς ανοικτών κειμένων.
- Το C είναι ο χώρος όλων των δυνατών κρυπτογραφημένων μηνυμάτων ή αλλιώς κρυπτοκειμένων.
- Το k είναι ο χώρος όλων των δυνατών κλειδιών ή αλλιώς κλειδοχώρος.
- Η E είναι ο κρυπτογραφικός μετασχηματισμός ή κρυπτογραφική συνάρτηση.
- Η D είναι η αντίστροφη συνάρτηση ή μετασχηματισμός αποκρυπτογράφησης.

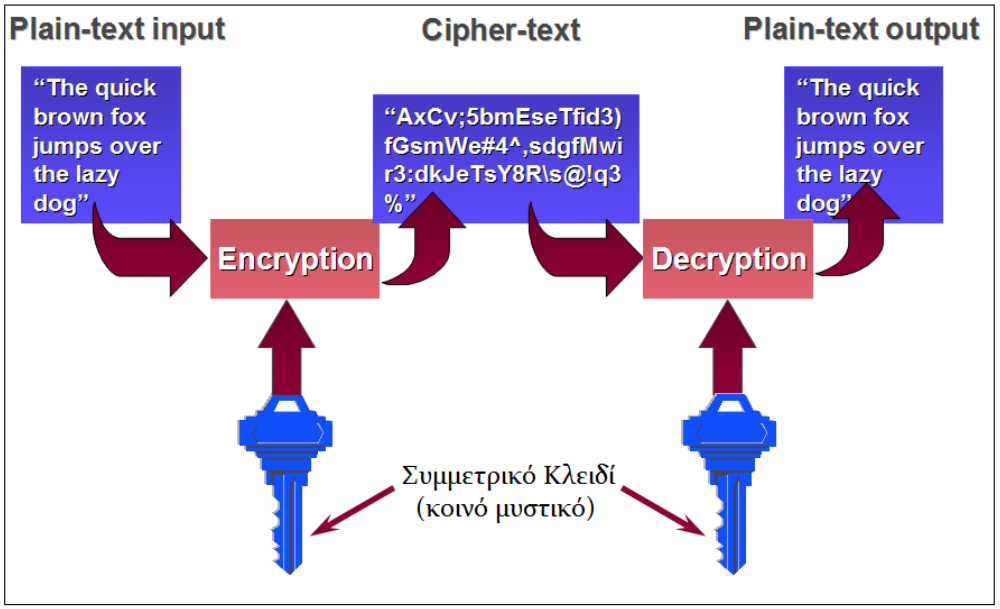


Σχήμα 3.1 : Τυπικό σύστημα κρυπτογράφησης

Στις μέρες μας υπάρχουν δύο είδη γνωστών αλγόριθμων κρυπτογράφησης. Οι συμμετρικοί αλγόριθμοι ή αλγόριθμοι ιδιωτικού κλειδιού και οι ασύμμετροι αλγόριθμοι ή αλγόριθμοι δημόσιου κλειδιού.

3.2.1 Κρυπτογραφία συμμετρικού κλειδιού

Ένα συμμετρικό κρυπτοσύστημα χρησιμοποιεί κατά την διαδικασία κρυπτογράφησης-αποκρυπτογράφησης ένα κοινό κλειδί K. Η ασφάλεια αυτών των αλγορίθμων βασίζεται στην μυστικότητα του κλειδιού. Τα συμμετρικά κρυπτοσυστήματα προϋποθέτουν την ανταλλαγή του κλειδιού μέσα από ένα ασφαλές κανάλι επικοινωνίας.



Σχήμα 3.2 : Κρυπτογραφία συμμετρικού κλειδιού

Έστω ότι δύο μέρη, η Alice και ο Bob θέλουν να επικοινωνήσουν χρησιμοποιώντας κρυπτογραφία συμμετρικού κλειδιού. Η Alice παρέχει ένα κλειδί K_A ως είσοδο στον αλγόριθμο κρυπτογράφησης. Ο αλγόριθμος κρυπτογράφησης με χρήση του κλειδιού και του αρχικού κειμένου m, παράγει το κρυπτοκείμενο $K_A(m)$[5]. Ο Bob από το μέρος του παρέχει ένα κλειδί K_B στον αλγόριθμο αποκρυπτογράφησης, ο οποίος από το κρυπτοκείμενο και το κλειδί του Bob παράγει το αρχικό κείμενο m. Δηλαδή, ο Bob αν δεχθεί ένα κρυπτογραφημένο κείμενο της μορφής $K_A(m)$, μπορεί να το αποκρυπτογραφήσει υπολογίζοντας το $K_B(K_A(m)) = m$. Στα συστήματα συμμετρικού κλειδιού, τα κλειδιά της Alice και του Bob είναι πανομοιότυπα και μυστικά, όπως φαίνεται και στο σχήμα 3.2. Ισχύει επομένως ότι $K_A \equiv K_B \equiv K$.

Στην εποχή μας, ένας από τους γνωστούς αλγόριθμους κρυπτογράφησης είναι το **Πρότυπο Κρυπτογράφησης Δεδομένων** (Data Encryption Standard, DES). Το DES κρυπτογραφεί το αρχικό κείμενο σε κομμάτια 64-bit (**block cipher**), χρησιμοποιώντας ένα κλειδί 64-bit. Κάθε ένα από τα 8 bytes του κλειδιού έχουν ένα bit ισοτιμίας. Οπότε το κλειδί DES έχει ουσιαστικά μήκος 56 bits. Σύμφωνα με το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας NIST, σκοπός του DES είναι να περιπλέξει τελείως τα δεδομένα και το κλειδί, ώστε κάθε bit του κρυπτοκειμένου να εξαρτάται από κάθε bit των δεδομένων και από κάθε bit του κλειδιού. Η λειτουργία του DES ξεφεύγει από τα όρια του θέματος της διπλωματικής εργασίας και δεν θα αναφερθούμε περαιτέρω.

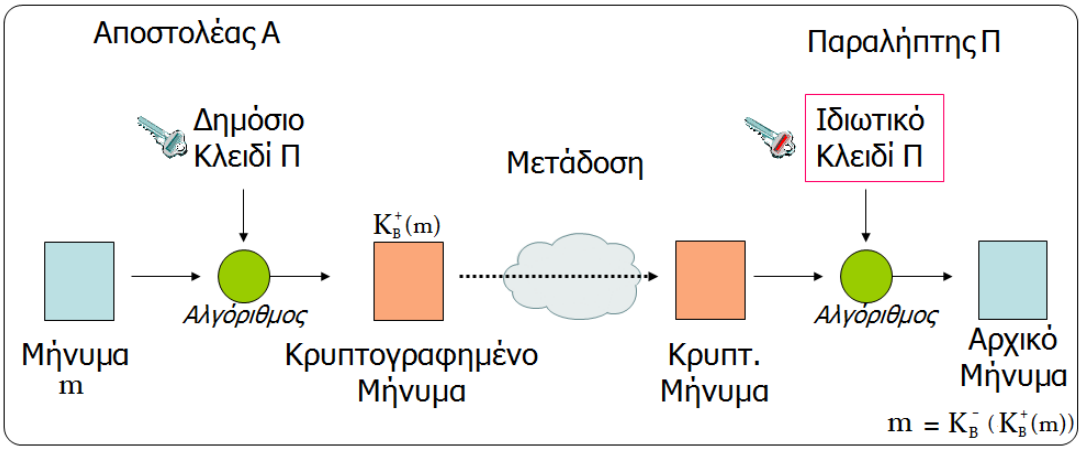
Το DES παρόλα αυτά δεν είναι ασφαλές. Εκτιμάται ότι μια μηχανή μπορεί να σπάσει το DES 56-bit σε ένα δευτερόλεπτο, δοκιμάζοντας 2^{55} κλειδιά. Ο διάδοχος του DES είναι το **Πρότυπο Προχωρημένης Κρυπτογράφησης** (Advanced Encryption Standard, AES), που είναι ένας αλγόριθμος συμμετρικού κλειδιού ο οποίος επεξεργάζεται δεδομένα σε block των 128-bit και μπορεί να λειτουργήσει με κλειδιά μήκους 128, 198 και 256 bits. Άλλοι γνωστοί αλγόριθμοι κρυπτογράφησης είναι οι : triple DES (3DES), RC2, RC4, RC5, IDEA.

Οι αλγόριθμοι κρυπτογράφησης συμμετρικού κλειδιού είναι γενικά πολύ γρήγοροι, αλλά έχουν ένα σοβαρό μειονέκτημα. Αυτό σχετίζεται με τα προβλήματα στην ασφάλεια διανομής του κοινού μυστικού κλειδιού.

[5] Ο συμβολισμός $K_A(m)$ θα αναφέρεται ως το κρυπτογραφημένο κείμενο του m με χρήση του κλειδιού K_A.

3.2.2 Κρυπτογραφία δημόσιου κλειδιού

Ένα ασύμμετρο κρυπτοσύστημα χρησιμοποιεί κατά την διαδικασία κρυπτογράφησης-αποκρυπτογράφησης ένα ζεύγος κλειδιών, όπως βλέπουμε και στο σχήμα 3.3. Ένα από τα κλειδιά (δημόσιο κλειδί) είναι γνωστό και στα δύο μέρη. Το άλλο κλειδί είναι γνωστό μόνο στον έναν από τους δύο (ιδιωτικό κλειδί). Υποθέτουμε ότι η Alice θέλει να επικοινωνήσει με τον Bob. Αντί να μοιράζονται η Alice και ο Bob ένα μυστικό κλειδί, ο Bob, δηλαδή ο παραλήπτης των μηνυμάτων, έχει δύο κλειδιά. Το δημόσιο κλειδί K_B^+, που είναι γνωστό σε όλο τον υπόλοιπο κόσμο και το ιδιωτικό του κλειδί K_B^-, που γνωρίζει μόνο ο ίδιος. Για να επικοινωνήσει η Alice (η αποστολέας των μηνυμάτων) με τον Bob, θα πρέπει να πάρει πρώτα το δημόσιο κλειδί του Bob. Στη συνέχεια κρυπτογραφεί το αρχικό μήνυμα m με χρήση ενός γνωστού αλγόριθμου κρυπτογράφησης και του δημόσιου κλειδιού του Bob, υπολογίζοντας το $K_B^+(m)$. Ο Bob λαμβάνει το κρυπτοκείμενο και χρησιμοποιεί έναν γνωστό αλγόριθμο αποκρυπτογράφησης με είσοδο το ιδιωτικό του κλειδί, για να αποκρυπτογραφήσει την κρυπτογραφημένη πληροφορία που του έστειλε η Alice. Υπολογίζει δηλαδή το $K_B^-(K_B^+(m))$.



Σχήμα 3.3 : Κρυπτογραφία δημόσιου κλειδιού

Αν και υπάρχουν πολλοί αλγόριθμοι κρυπτογράφησης δημόσιου κλειδιού, όπως οι DSA (Digital Signature Algorithm), Diffie-Hellman κλπ, ο αλγόριθμος RSA έχει γίνει συνώνυμος με την κρυπτογραφία δημόσιου κλειδιού. Ο RSA βασίζεται στην δυσκολία παραγοντοποίησης μεγάλων αριθμών (σήμερα, συνήθως της τάξης των

1024 με 2048 bits). Χρησιμοποιούνται δυο κλειδιά, ένα δημόσιο κατά την διάρκεια της κρυπτογράφησης και ένα κρυφό για την αποκρυπτογράφηση.

Δημιουργία των κλειδιών

1. Επιλογή δυο τυχαίων (μεγάλων) πρώτων αριθμών p και q έτσι ώστε $p \neq q$.
2. Υπολογίζουμε το $n = p \cdot q$.
3. Υπολογίζουμε την συνάρτηση του Euler : $\varphi = (p-1)(q-1)$.
4. Επιλογή ενός αριθμού $e > 1$ έτσι ώστε να μην έχει κοινούς διαιρέτες με τον φ παρά μόνο το 1, δηλαδή $e \wedge \varphi(n) = 1$.
5. Υπολογίζουμε τον αριθμό d έτσι ώστε το de-1 να διαιρείται ακριβώς με το φ. $d.e \equiv 1 \bmod \varphi(n)$.

Για την εύρεση πρώτων αριθμών χρησιμοποιούνται πιθανολογικοί αλγόριθμοι, ενώ συνηθισμένες επιλογές για το e είναι το 3, 7 και 2^{16} + 1. Γενικώς, πρέπει να προσέχουμε ότι μικροί αριθμοί οδηγούν σε ταχύτερους υπολογισμούς αλλά και σε πιο αδύνατη ασφάλεια. Τα κλειδιά είναι τα εξής:

- ✓ Δημόσιο : (n, e)
- ✓ Ιδιωτικό : (n, d)

Μπορούμε τώρα να δημοσιεύσουμε το πρώτο κλειδί, δίνοντας έτσι την δυνατότητα σε οποιονδήποτε να μας στείλει κρυπτογραφημένα μηνύματα που μόνο εμείς (χάρη στο κρυφό ιδιωτικό κλειδί) μπορούμε να αποκρυπτογραφήσουμε.

Κρυπτογράφηση

Το κρυπτογραφημένο μήνυμα c υπολογίζεται με τον εξής τρόπο:

$$c = m^e \bmod n$$

Αποκρυπτογράφηση

Αφού ληφθεί ένα κρυπτογραφημένο μήνυμα c, για να διαβάσουμε το αρχικό μήνυμα προβαίνουμε στον ακόλουθο υπολογισμό:

$$m = c^d \bmod n$$

Με την χρήση δημόσιου κλειδιού, το πρόβλημα διανομής ενός μυστικού κλειδιού εξαλείφεται. Πλέον δεν χρειάζεται να μεταδοθεί ή να αποκαλυφθεί κάποιο ιδιωτικό κλειδί με κίνδυνο την αντιγραφή του. Επιπλέον, ο αλγόριθμος δημόσιου κλειδιού βρίσκει εφαρμογή στις ψηφιακές υπογραφές, όπου κάποιος μπορεί να επιβεβαιώσει την ταυτότητά του μόνο με το ιδιωτικό του κλειδί. Λεπτομέρειες θα δούμε στην Ενότητα 3.4. Όπως το νόμισμα έχει δύο όψεις έτσι και στην κρυπτογραφία δημόσιου κλειδιού υπάρχουν αδύνατα σημεία. Καταρχήν, η κρυπτογραφία τέτοιου είδους είναι αργή στην εκτέλεση. Χαρακτηριστικά το DES είναι τουλάχιστον 100 φορές πιο γρήγορο σε λογισμικό και έως 10.000 φορές πιο γρήγορο σε υλικό από τον RSA. Επιπροσθέτως, υπάρχουν αμφισβητήσεις σχετικά με τα δημόσια κλειδιά. Συνίσταται επομένως η εγκατάσταση αρχών πιστοποίησης και οργανωμένων υποδομών δημόσιου κλειδιού.

Κλείνοντας την ενότητα να αναφέρουμε πως υπάρχουν και μεικτά σχήματα κρυπτογραφίας δημόσιου-ιδιωτικού κλειδιού (υβριδικά κρυπτοσυστήματα), όπου ο αποστολέας με χρήση του δημόσιου κλειδιού του παραλήπτη, του στέλνει το μυστικό κλειδί της συνόδου (session key). Πλέον ο παραλήπτης γνωρίζει το κλειδί της συνόδου που θα χρησιμοποιήσει ο αποστολέας για την κρυπτογραφημένη πχ κατά DES μεταφορά δεδομένων του.

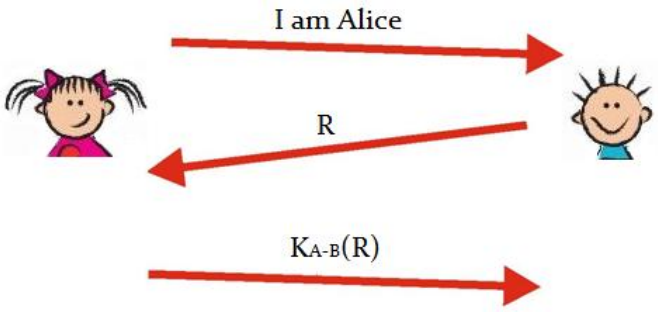
3.3 Ταυτοποίηση

Υποθέτουμε πάλι ότι η Alice θέλει να επικοινωνήσει με τον Bob. Προτού ξεκινήσει η μετάδοση δεδομένων ο Bob θέλει να σιγουρευτεί ότι η Alice είναι όντως αυτή που λέει ότι είναι. Το πρόβλημα αυτό είναι γνωστό ως ταυτοποίηση. Η ταυτοποίηση ουσιαστικά είναι η διαδικασία απόδειξης ταυτότητας κάποιου σε κάποιον άλλον. Όταν δύο μέρη επικοινωνούν μέσω ενός δικτύου η διαδικασία της ταυτοποίησης είναι περισσότερο πολύπλοκη απ' ότι στην πραγματική ζωή. Όταν μιλάμε για ταυτοποίηση πάνω σε ένα δίκτυο δεν υπάρχει η οπτική εμφάνιση ή η φωνή. Σε αυτήν την

περίπτωση, η ταυτοποίηση στηρίζεται αποκλειστικά στα μηνύματα και στα δεδομένα που ανταλλάσσονται σαν κομμάτι ενός **πρωτοκόλλου ταυτοποίησης**. Στηριζόμενοι στα δύο είδη κρυπτογραφίας που μελετήσαμε στην προηγούμενη ενότητα, θα παρουσιάσουμε στην συνέχεια δύο πρωτόκολλα ταυτοποίησης τα οποία ονομάζουμε αυθαίρετα **ap** (authentication protocol). Το πρωτόκολλο ταυτοποίησης ap1.0 βασίζεται στην κρυπτογραφία συμμετρικού κλειδιού και το πρωτόκολλο ταυτοποίησης ap2.0 στην κρυπτογραφία δημόσιου κλειδιού.

Πρωτόκολλο ταυτοποίησης ap1.0

Μια κλασική προσέγγιση ταυτοποίησης είναι η Alice να χρησιμοποιήσει έναν μυστικό κωδικό πρόσβασης. Ο κωδικός αυτός για να απαγορευτεί σε άλλους να τον κλέψουν, κρυπτογραφείται. Στο πρωτόκολλο ταυτοποίησης ap1.0 η Alice και ο Bob μοιράζονται ένα συμμετρικό μυστικό κλειδί K_{A-B}. Επιπλέον υπάρχει και ο nonce R. Ένας nonce είναι ένας αριθμός που θα χρησιμοποιήσει το πρωτόκολλο μόνο μια φορά κατά την διάρκεια της ζωής του. Με αυτόν τον τρόπο ο Bob προστατεύεται από το λεγόμενο *playback attack*, δηλαδή τον κίνδυνο κάποιος εισβολέας να καταγράψει την κρυπτογραφημένη έκδοση του κωδικού πρόσβασης και ύστερα να την αναπαράγει στον Bob προσποιούμενος ότι είναι η Alice. Παρόμοια τεχνική χρησιμοποιείται και στο πρωτόκολλο τριμερούς χειραψίας TCP.



Σχήμα 3.4 : Το πρωτόκολλο ap1.0

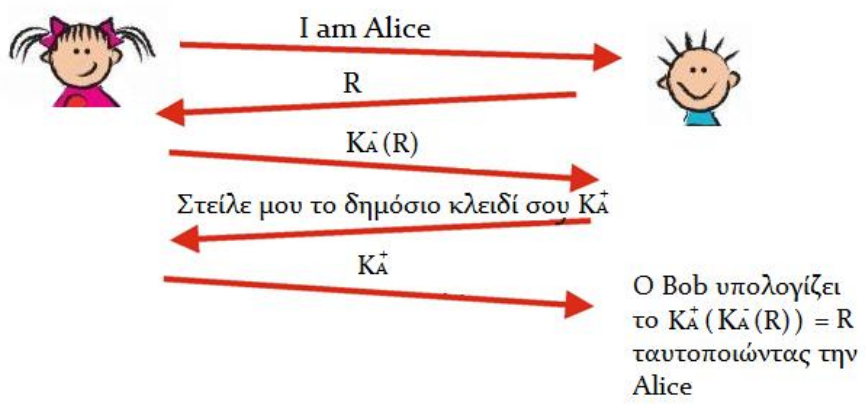
Τα βήματα που ακολουθούνται για την ταυτοποίηση της Alice στον Bob όπως παρουσιάζεται στο σχήμα 3.4 είναι τα εξής :

1. Η Alice στέλνει το μήνυμα “I am Alice” στον Bob, λέγοντας του ότι είναι η ίδια.
2. Ο Bob στέλνει στην Alice έναν τυχαίο αριθμό nonce R.
3. Η Alice με την βοήθεια του συμμετρικού μυστικού κλειδιού K_{A-B} κρυπτογραφεί τον nonce R, και τον στέλνει πίσω στον Bob.
4. Ο Bob αποκρυπτογραφεί το μήνυμα $K_{A-B}(R)$ που του έστειλε η Alice. Αν προκύψει ο nonce R που έστειλε στην Alice, τότε η ταυτοποίηση της Alice είναι έγκυρη.

Ο Bob μπορεί να είναι σίγουρος ότι η Alice είναι όντως αυτή που λέει ότι είναι εφόσον μόνο οι δυο τους γνωρίζουν το συμμετρικό μυστικό κλειδί.

<u>Πρωτόκολλο ταυτοποίησης ap2.0</u>

Στο σημείο αυτό γεννιέται η απορία : “ Η χρήση nonce και κρυπτογραφίας δημόσιου κλειδιού θα ήταν καλύτερη; ”. Με μια πρώτη σκέψη θα λέγαμε πως η αποφυγή της δυσκολίας ενός συστήματος διαμοιρασμού κλειδιού είναι ένα πλεονέκτημα. Ας αναλύσουμε πως δουλεύει το πρωτόκολλο ap2.0, που φαίνεται στην ακόλουθη εικόνα.



Σχήμα 3.5 : Το πρωτόκολλο ap2.0

1. Η Alice στέλνει το μήνυμα “I am Alice” στον Bob, λέγοντας του ότι είναι η ίδια.
2. Ο Bob στέλνει στην Alice έναν τυχαίο αριθμό nonce R.
3. Η Alice με την βοήθεια του ιδιωτικού της μυστικού κλειδιού K_{A}^{-} κρυπτογραφεί τον nonce R, και τον στέλνει πίσω στον Bob.
4. Ο Bob αποκρυπτογραφεί το μήνυμα $K_{\text{A}}^{-}(R)$ που του έστειλε η Alice εφαρμόζοντας το δημόσιο κλειδί της Alice K_{A}^{+} υπολογίζοντας το $K_{A}^{+}(K_{\text{A}}^{-}(R))$. Αν προκύψει ο nonce R που έστειλε στην Alice, τότε η ταυτοποίηση της Alice είναι έγκυρη.

Παρόλα αυτά το πρωτόκολλο ap2.0 δεν είναι το ίδιο ασφαλές με το πρωτόκολλο ap1.0. Στην πραγματικότητα το πρωτόκολλο ap2.0 είναι τόσο ασφαλές όσο και η διανομή δημόσιων κλειδιών. Στην Ενότητα 3.5 θα δούμε ασφαλείς μεθόδους διανομής δημόσιων κλειδιών.

3.4 Ακεραιότητα

Σε πολλά σενάρια ο αποστολέας και ο δέκτης ενός μηνύματος θέλουν να έχουν την σιγουριά ότι το μήνυμα αυτό δεν έχει υποστεί αλλοιώσεις είτε επιτηδευμένα, είτε κατά λάθος. Ενώ η κρυπτογράφηση «μεταμφιέζει» το περιεχόμενο του μηνύματος, δεν μπορεί να εγγυηθεί ότι αυτό θα παραμείνει αναλλοίωτο. Στο σημείο αυτό είναι που μπαίνουν στο παιχνίδι οι ψηφιακές υπογραφές. Στον κόσμο του διαδικτύου όταν κάποιος θέλει να δηλώσει τον ιδιοκτήτη ενός εγγράφου ή να υποδηλώσει την συμφωνία του με τα περιεχόμενα ενός εγγράφου κάνει χρήση της ψηφιακής του υπογραφής. Μια ψηφιακή υπογραφή έχει ακριβώς τις ίδιες ιδιότητες με την κανονική υπογραφή (είναι επαληθεύσιμη, δεν μπορεί να πλαστογραφηθεί, δεν μπορεί να ανακληθεί κλπ) και έχει ως βάση την κρυπτογραφία δημόσιου κλειδιού.
Στην συνέχεια της ενότητας παρουσιάζουμε τον τρόπο με τον οποίο παράγονται οι ψηφιακές υπογραφές, καθώς και τον τρόπο με τον οποίο επαληθεύονται, δηλαδή ότι η υπογραφή ανήκει όντως σε αυτόν που υπέγραψε το ψηφιακό έγγραφο.

3.4.1 Ψηφιακές υπογραφές

Μια ψηφιακή υπογραφή είναι ένας τύπος ασύμμετρου συστήματος κρυπτογραφίας. Έστω πάλι ότι έχουμε τους γνωστούς μας Alice και Bob, οι οποίοι έχουν επικοινωνήσει και έχουν ανταλλάξει μηνύματα και έγγραφα. Πώς η Alice είναι σίγουρη ότι κάποιο έγγραφο που έχει λάβει από τον Bob ανήκει όντως σε εκείνον; Ασφαλώς από την ψηφιακή υπογραφή του Bob με την οποία υπέγραψε το έγγραφο.
Ας δούμε τώρα τον τρόπο παραγωγής μιας ψηφιακής υπογραφής. Υποθέτουμε ότι ο Bob θέλει να υπογράψει ψηφιακά ένα έγγραφο e. Το μόνο που έχει να κάνει είναι να κρυπτογραφήσει το έγγραφο χρησιμοποιώντας το ιδιωτικό του κλειδί K_{B}^{-}. Υπολογίζει δηλαδή το $K_B^-(e)$. Σκοπός βέβαια δεν είναι να κρυπτογραφήσει το έγγραφο για να το προστατέψει από τρίτους, αλλά να υπογράψει το έγγραφο με τέτοιο τρόπο ώστε να είναι επαληθεύσιμη η ταυτότητά του και να μην μπορεί να παραποιηθεί ή να ανακληθεί. Τώρα το μόνο που έχει να κάνει η Alice είναι να ελέγξει αν το ιδιωτικό κλειδί του Bob φτάνει για την παραγωγή της ψηφιακής του υπογραφής.
Η Alice έχει το έγγραφο e και το $K_B^-(e)$. Μπορεί να προμηθευτεί όμως και το δημόσιο κλειδί του Bob K_B^+ (λεπτομέρειες στην Ενότητα 3.5). Αν εφαρμόσει το δημόσιο κλειδί του Bob στο κρυπτογραφημένο έγγραφο $K_B^-(e)$, τότε θα πάρει το $K_B^+(K_B^-(e))$. Όμως ισχύει ότι $e = K_B^+(K_B^-(e))$. Καταλήγει συνεπώς στο αρχικό έγγραφο που της έστειλε ο Bob. Άρα μόνο ο Bob θα μπορούσε να είχε υπογράψει το έγγραφο αυτό, καθώς το μόνο άτομο που γνωρίζει το ιδιωτικό κλειδί K_{B}^{-} είναι ο ίδιος ο Bob. Σε περίπτωση που το έγγραφο e είχε τροποποιηθεί ή παραποιηθεί σε κάποια εναλλακτική μορφή e', η υπογραφή που είχε δημιουργήσει ο Bob για το e δεν θα ήταν έγκυρη για το e'.

<u>Συνόψεις μηνυμάτων</u>

Η υπογραφή δεδομένων μέσω πλήρους κρυπτογράφησης/αποκρυπτογράφησης μπορεί να θεωρηθεί υπερβολή λόγω των επιπρόσθετων βαρών. Μια πιο αποδοτική προσέγγιση είναι η χρήση σύνοψης μηνύματος (fingerprint ή message digest) που είναι παρόμοια τεχνική με ένα άθροισμα ελέγχου. Στη διαδικασία της δημιουργίας και επαλήθευσης της υπογραφής εμπλέκεται και η έννοια της συνάρτησης κατακερματισμού (one way hash). Με την εφαρμογή της συνάρτησης

κατακερματισμού, από ένα μήνυμα ανεξαρτήτως του μεγέθους του, παράγεται η σύνοψή του, η οποία είναι μία σειρά από bits συγκεκριμένου μεγέθους (π.χ. 128 ή 160 bits). Η σύνοψη του μηνύματος είναι μία ψηφιακή αναπαράσταση του μηνύματος, είναι μοναδική για το μήνυμα και το αντιπροσωπεύει.
Η συνάρτηση κατακερματισμού είναι μονόδρομη, διότι από την σύνοψη που δημιουργεί, είναι υπολογιστικά αδύνατον κάποιος να εξάγει το αρχικό μήνυμα. Η πιθανότητα δύο μηνύματα να έχουν την ίδια σύνοψη είναι εξαιρετικά μικρή. Αυτό σημαίνει ότι αν το μήνυμα του αποστολέα έχει κάποια συγκεκριμένη σύνοψη και το μήνυμα που λάβει ο παραλήπτης (χρησιμοποιώντας την ίδια συνάρτηση κατακερματισμού) παράγει διαφορετική σύνοψη, τότε το μήνυμα κατά την μετάδοσή του έχει αλλοιωθεί (μη ακεραιότητα). Οποιαδήποτε αλλαγή σε ένα μήνυμα συνεπάγεται και τη δημιουργία διαφορετικής σύνοψης.

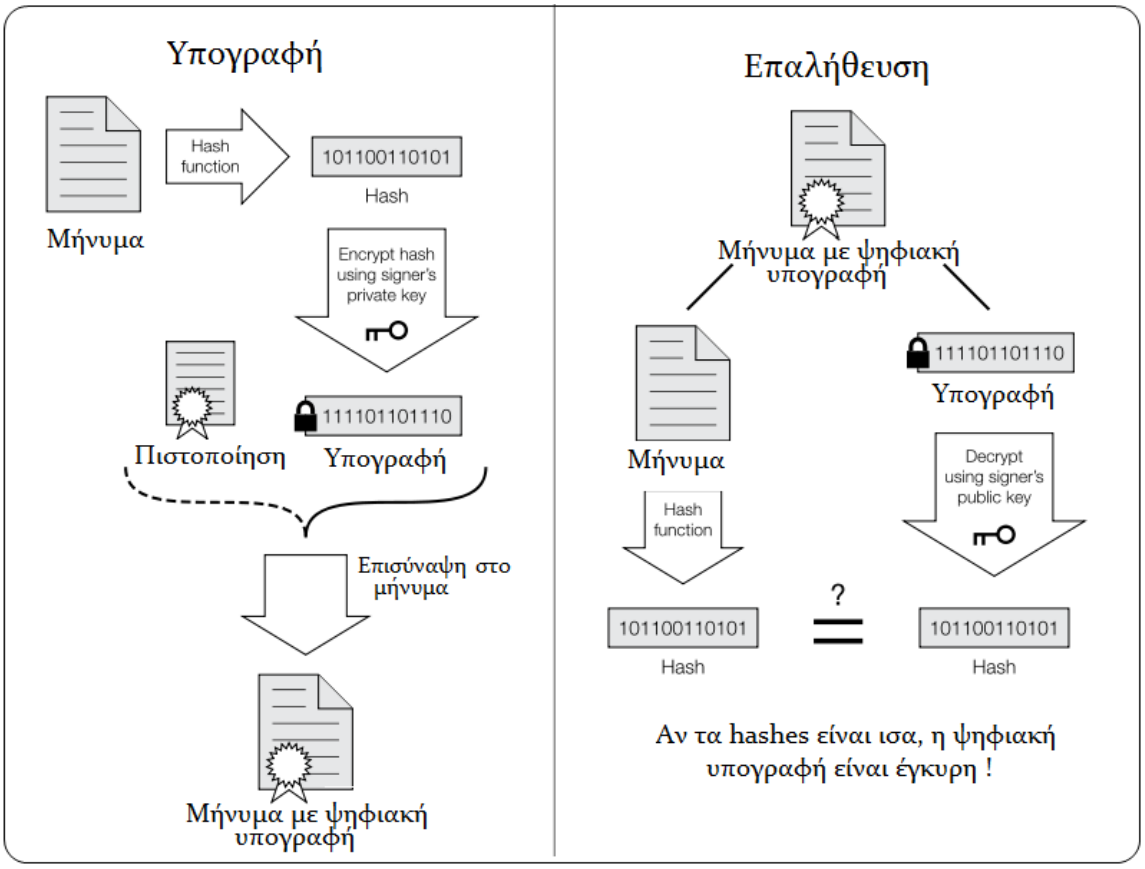
Δημιουργία υπογραφής

1. Ο αποστολέας χρησιμοποιώντας κάποιον αλγόριθμο κατακερματισμού δημιουργεί τη σύνοψη του μηνύματος που θέλει να στείλει. Ανεξάρτητα από το μέγεθος του μηνύματος, αυτό που θα παραχθεί θα είναι μία συγκεκριμένου μήκους σειρά ψηφίων.
2. Με το ιδιωτικό του κλειδί, ο αποστολέας κρυπτογραφεί τη σύνοψη. Αυτό που παράγεται είναι η ψηφιακή υπογραφή. Η υπογραφή είναι ουσιαστικά μία σειρά ψηφίων συγκεκριμένου πλήθους.
3. Η κρυπτογραφημένη σύνοψη (ψηφιακή υπογραφή) προσαρτάται στο κείμενο και το μήνυμα με τη ψηφιακή υπογραφή μεταδίδονται μέσω του δικτύου.

Επαλήθευση υπογραφής

1. Ο παραλήπτης αποσπά από το μήνυμα την ψηφιακή υπογραφή (κρυπτογραφημένη με το ιδιωτικό κλειδί του αποστολέα σύνοψη).
2. Εφαρμόζοντας στο μήνυμα που έλαβε τον ίδιο αλγόριθμο κατακερματισμού, ο παραλήπτης δημιουργεί τη σύνοψη του μηνύματος.
3. Στη συνέχεια, αποκρυπτογραφεί με το δημόσιο κλειδί του αποστολέα, την κρυπτογραφημένη σύνοψη του μηνύματος (ψηφιακή υπογραφή).
4. Συγκρίνονται οι δύο συνόψεις και αν βρεθούν ίδιες, αυτό σημαίνει ότι το μήνυμα που έλαβε ο παραλήπτης είναι ακέραιο. Αν το μήνυμα έχει μεταβληθεί, η σύνοψη

που θα παράγει ο παραλήπτης θα είναι διαφορετική από την σύνοψη που έχει κρυπτογραφηθεί.



Σχήμα 3.6 : Παραγωγή και επαλήθευση ψηφιακής υπογραφής

3.5 Διανομή κλειδιού και πιστοποίηση

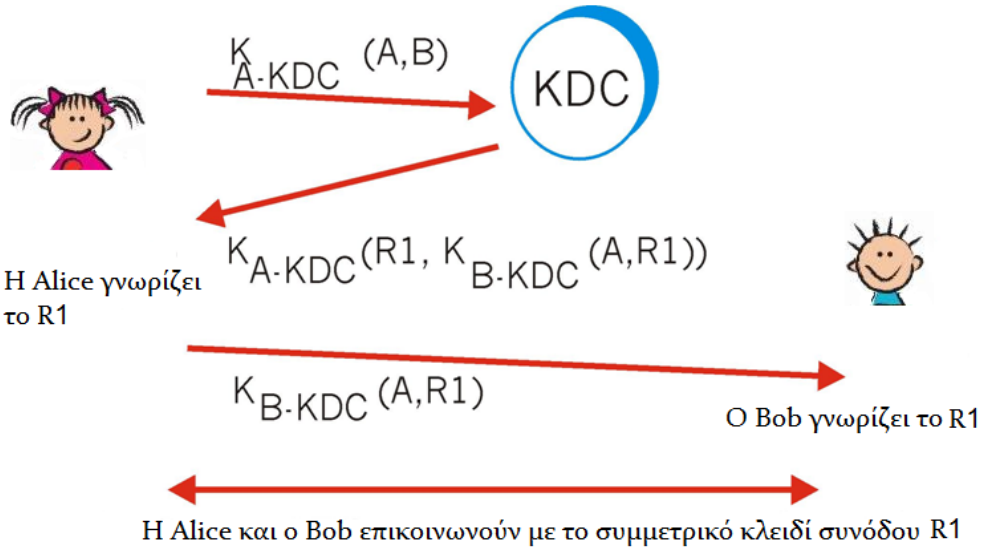
Τα δύο είδη κρυπτογραφίας που αναλύσαμε νωρίτερα στο κεφάλαιο αυτό παρουσιάζουν το καθένα από ένα σοβαρό μειονέκτημα. Η κρυπτογραφία συμμετρικού κλειδιού έχει το πρόβλημα τής από κοινού συμφωνίας ενός μυστικού διαμοιρασμένου κλειδιού. Από την άλλη, η κρυπτογραφία δημόσιου κλειδιού ενώ λύνει το πρόβλημα του συμμετρικού κλειδιού έχει πρόβλημα με την δυσκολία ασφαλούς λήψης του δημόσιου κλειδιού.

Η λύση στις παραπάνω δυσκολίες είναι η χρήση ενός *έμπιστου μεσολαβητή* (Trusted Third Party). Στην κρυπτογραφία συμμετρικού κλειδιού ο έμπιστος μεσολαβητής καλείται *κέντρο διανομής κλειδιού* (key distribution center, KDC) και στην

κρυπτογραφία δημόσιου κλειδιού ο έμπιστος μεσολαβητής καλείται αρχή πιστοποίησης (certification authority, CA).

3.5.1 Κέντρο διανομής κλειδιού KDC

Το KDC είναι ένας εξυπηρετητής που μοιράζεται με κάθε εγγεγραμμένο του χρήστη ένα ιδιωτικό κλειδί. Μπορούμε να το φανταστούμε σαν έναν κωδικό που δίνει ο χρήστης όταν εγγράφεται στο KDC και ο οποίος αποθηκεύεται εκεί. Όταν η Alice θελήσει να επικοινωνήσει με τον Bob (και οι δυο τους είναι εγγεγραμμένοι στο KDC) τότε στέλνει ένα μήνυμα (A,B) το οποίο υποδηλώνει ότι θέλει να επικοινωνήσει με τον Bob. Για λόγους ασφάλειας κρυπτογραφεί το μήνυμα αυτό με το ιδιωτικό της κλειδί K_{A-KDC} που γνωρίζουν μόνο η ίδια και το KDC. Το KDC από την πλευρά του αποκρυπτογραφεί το μήνυμα της Alice και παράγει έναν τυχαίο αριθμό R1. Αυτός ο αριθμός ονομάζεται κλειδί συνόδου μιας χρήσης και είναι το διαμοιρασμένο κλειδί που η Alice και ο Bob θα χρησιμοποιήσουν για συμμετρικό κλειδί της σύνδεσής τους.



Σχήμα 3.7 : Κλειδί συνόδου R1 από ένα KDC

Η Alice λαμβάνει από το KDC ένα κρυπτογραφημένο μήνυμα που περιέχει τον R1, αλλά και το μήνυμα $K_{B-KDC}(A,R1)$. Στέλνει δηλαδή το KDC κρυπτογραφημένο με το ιδιωτικό κλειδί του Bob και ένα μήνυμα που υποδηλώνει ότι η Alice θέλει να επικοινωνήσει μαζί του χρησιμοποιώντας το κλειδί συνόδου R1. Όμως η Alice δεν

μπορεί να αποκρυπτογραφήσει το μήνυμα αυτό και επομένως δεν ξέρει τι λέει. Στη συνέχεια η Alice κάνει προώθηση του μηνύματος στον Bob. Πλέον και οι δυο τους γνωρίζουν το κλειδί συνόδου μιας χρήσης. Φυσικά, προτού αρχίσουν την επικοινωνία τους, ο Bob φροντίζει να ταυτοποιήσει την Alice χρησιμοποιώντας το R1.

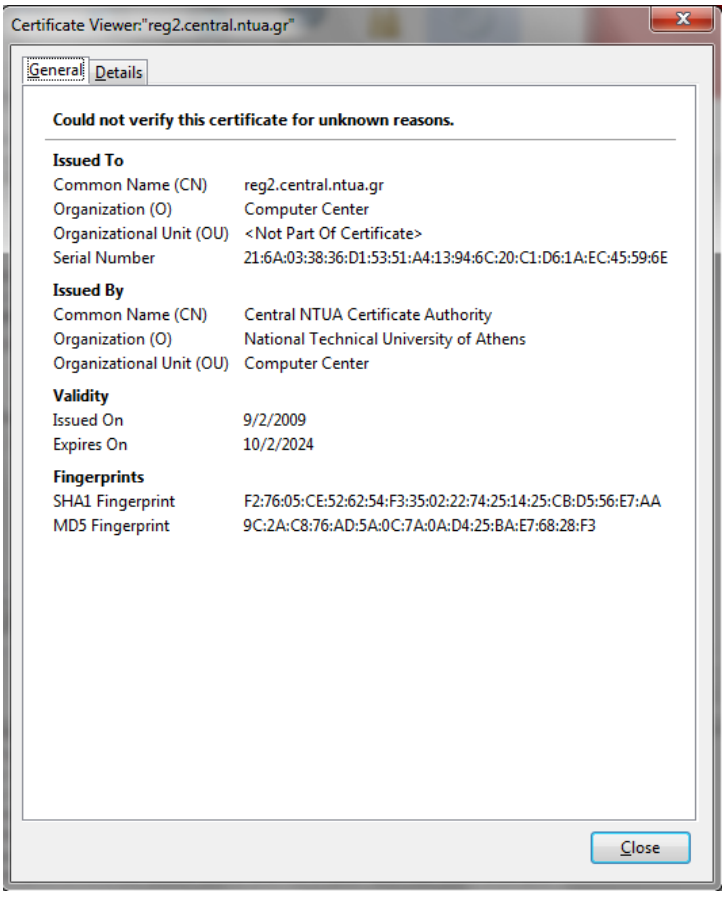
3.5.2 Πιστοποίηση δημόσιου κλειδιού

Η κρυπτογραφία δημόσιου κλειδιού εξαλείφει την ανάγκη για μια υποδομή KDC. Παρόλα αυτά όπως είδαμε και στο πρωτόκολλο ταυτοποίησης ap2.0, τα δύο μέρη που επικοινωνούν μεταξύ τους με υποδομή δημόσιου κλειδιού χρειάζεται να ανταλλάξουν δημόσια κλειδιά. Το δημόσιο κλειδί ενός χρήστη μπορεί να γίνει διαθέσιμο και γνωστό μέσω της προσωπικής του ιστοσελίδας ή με την αποστολή ενός e-mail. Το πρόβλημα όμως είναι ότι οι οντότητες, είτε είναι χρήστες, είτε δρομολογητές, είτε browsers, πρέπει να είναι σίγουρες ότι το δημόσιο κλειδί της οντότητας με την οποία επικοινωνούν είναι στην πραγματικότητα δικό της. Η δέσμευση ενός δημόσιου κλειδιού με μια συγκεκριμένη οντότητα γίνεται από μια αρχή πιστοποίησης (CA). Ας δούμε τώρα τα βήματα που ακολουθούνται για την παραγωγή ενός πιστοποιητικού και την επαλήθευση του :

- Όταν ο Bob θέλει να δεσμευτεί με ένα δημόσιο κλειδί κάνει αίτηση για ψηφιακό πιστοποιητικό σε μια CA. Ένα ψηφιακό πιστοποιητικό είναι ένα ηλεκτρονικό έγγραφο που χρησιμοποιείται για την αναγνώριση μιας οντότητας (φυσικό πρόσωπο, εξυπηρετητής, οργανισμός κοκ) και την ανάκτηση του δημοσίου κλειδιού αυτής.
- Η αρχή πιστοποίησης CA επιβεβαιώνει την ταυτότητα του Bob και εκδίδει το πιστοποιητικό το οποίο συνοπτικά περιλαμβάνει τα εξής στοιχεία :
    - Το ονοματεπώνυμο και διάφορες άλλες πληροφορίες σχετικά με τον κάτοχο του πιστοποιητικού.
    - Το δημόσιο κλειδί του κατόχου του πιστοποιητικού.
    - Την ημερομηνία λήξης του πιστοποιητικού.
    - Το όνομα και την ψηφιακή υπογραφή της Αρχής Πιστοποίησης που το εξέδωσε.

- Όταν η Alice επικοινωνεί με τον Bob λαμβάνει και το πιστοποιητικό του Bob. Χρησιμοποιεί το δημόσιο κλειδί της CA για να ελέγξει την εγκυρότητα του πιστοποιητικού του Bob από την ψηφιακή υπογραφή της CA. Πλέον μπορεί να είναι σίγουρη ότι επικοινωνεί με τον Bob και όχι με κάποιον εισβολέα.

Για να δούμε και οπτικά το ζήτημα αυτό, παρουσιάζουμε ένα ψηφιακό πιστοποιητικό που εκδόθηκε από την CA του NTUA και βρίσκεται αποθηκευμένο στον browser μας.



Σχήμα 3.8 : Ψηφιακό πιστοποιητικό από το NTUA

3.6 Σύστημα Kerberos

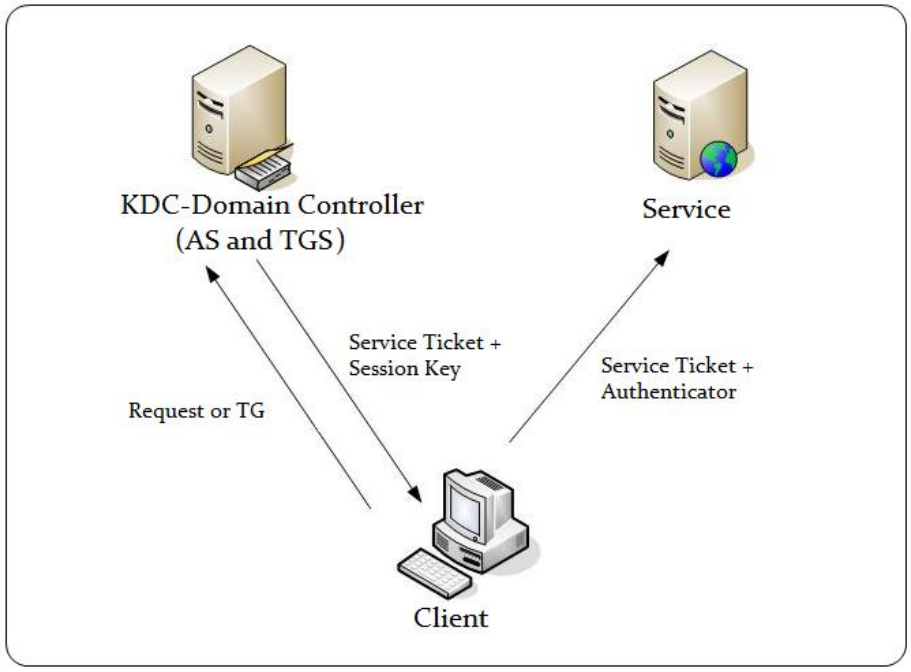
Το σύστημα Kerberos αναπτύχθηκε από το *Massachusetts Institute of Technology (MIT)* για να προστατέψει τις δικτυακές υπηρεσίες που παρέχονταν από το Project Athena και βασίζεται στο μοντέλο διανομής κλειδιών *(key distribution model)* των Needham και Schoeder. Οι εκδόσεις 1 έως 3 χρησιμοποιήθηκαν εσωτερικά από το MIT. Παρ' όλο που σχεδιάσθηκε αρχικά για χρήση με το Project Athena, η 4^{η} έκδοση

υιοθετήθηκε παγκοσμίως. Εξαιτίας του γεγονότος, όμως, ότι πολλά περιβάλλοντα είχαν απαιτήσεις που δεν μπορούσε να καλύψει η 4η έκδοση, νέα χαρακτηριστικά εισηγήθηκαν με την ανάπτυξη της έκδοσης 5.0 του Kerberos, που απευθυνόταν σε περισσότερες περιπτώσεις. Η τρέχουσα έκδοση είναι η 5.0.

Το Kerberos είναι μια υπηρεσία πιστοποίησης ταυτότητας η οποία αναπτύχθηκε με σκοπό να αντικαταστήσει το σύστημα πιστοποίησης βάσει ισχυρισμού *(authentication by assertion)*. Η πιστοποίηση βάσει ισχυρισμού στηρίζεται στην εξής αρχή : όταν ο χρήστης τρέχει ένα πρόγραμμα που απαιτεί πρόσβαση σε μία δικτυακή υπηρεσία, το πρόγραμμα ανακοινώνει στον εξυπηρετητή ότι λειτουργεί εκ μέρους του συγκεκριμένου χρήστη. Ο εξυπηρετητής πιστεύει τα στοιχεία που του παρέχει ο πελάτης (δηλαδή το πρόγραμμα) και εξυπηρετεί τον χρήστη χωρίς να ζητά άλλες αποδείξεις. Όπως γίνεται αντιληπτό η ασφάλεια που παρέχεται είναι από ελάχιστη έως ανύπαρκτη, κάτι που καθιστά το σύστημα ευάλωτο.

Ένα άλλο σύστημα που χρησιμοποιείται ευρέως είναι η συνοδεία του ονόματος του χρήστη από έναν μυστικό κωδικό. Στο εναλλακτικό αυτό σχήμα πιστοποίησης ταυτότητας υπάρχουν πολλά σημαντικά μειονεκτήματα. Το κυριότερο είναι πως το σύστημα είναι ευάλωτο σε επιθέσεις παθητικού τύπου (passive attacks), δεδομένου ότι ο κωδικός μεταδίδεται στο δίκτυο χωρίς κάποιου είδους κρυπτογράφηση. Το σύστημα Kerberos καλύπτει ένα σημαντικό κενό των συστημάτων πιστοποίησης ταυτότητας. Όπως γίνεται αντιληπτό ένα τέτοιο σύστημα είναι εντελώς διαφορετικό από τα συστήματα p2p. Πρόκειται για ένα συγκεντρωμένο δίκτυο, όπου ένας χρήστης προκειμένου να συνδεθεί με μια υπηρεσία, πρέπει να πιστοποιηθεί από την κεντρική αρχή του Kerberos. Στη συνέχεια αναλύουμε τα χαρακτηριστικά γνωρίσματα του Kerberos.

Το σύστημα ταυτοποίησης Kerberos χρησιμοποιεί τεχνικές κρυπτογράφησης δημόσιου κλειδιού και ένα κέντρο διανομής κλειδιού KDC. Ο εξυπηρετητής ταυτοποίησης του Kerberos (authentication server, AS) παίζει τον ρόλου του KDC. Ο AS δεν είναι μονάχα ο χώρος αποθήκευσης των μυστικών κλειδιών όλων των χρηστών (έτσι ώστε κάθε χρήστης να μπορεί να επικοινωνεί με ασφάλεια με τον AS), αλλά και των πληροφοριών για το ποιοι χρήστες έχουν δικαιώματα προσπέλασης σε ποιες υπηρεσίες, σε ποιους κόμβους του δικτύου κοκ. Στο ακόλουθο σχήμα περιγράφεται η λειτουργία του Kerberos.



Σχήμα 3.9 : Το σύστημα Kerberos

Έστω ότι ένας χρήστης (Client) θέλει να προσπελάσει μια υπηρεσία (Service) του δικτύου. Τότε ακολουθούνται τα ακόλουθα βήματα :

1. Ο Client έρχεται σε επαφή με τον Kerberos AS αιτώντας την σύνδεση με την υπηρεσία Service. Όλες οι επικοινωνίες ανάμεσα στον Client και τον AS κρυπτογραφούνται με ένα μυστικό κλειδί που διαμοιράζεται στον Client και τον AS. Η αίτηση που στέλνει ο Client στον AS καλείται authentication request και περιέχει τα στοιχεία της ταυτότητας του Client, το όνομα της υπηρεσίας Service, την ζητούμενη διάρκεια ζωής του ticket και ένα τυχαίο αριθμό που θα χρησιμοποιηθεί για το ταίριασμα της authentication request με την authentication response.
2. Ο AS ταυτοποιεί τον Client και αποκρίνεται στέλνοντας τα διαπιστευτήρια. Τα διαπιστευτήρια αποτελούνται από (α) ένα κλειδί συνόδου (session key) που χρησιμοποιείται σαν κλειδί κρυπτογράφησης και (β) ενός ticket για την υπηρεσία Service. Το session key και το ticket διαφέρουν για κάθε υπηρεσία με την οποία επικοινωνεί ο χρήστης. Σε αναλογία με το σχήμα 3.7 ο εξυπηρετητής ταυτοποίησης (που στην ορολογία του Kerberos τώρα ονομάζεται Ticket Granting Server, TGS) στέλνει στον Client το κλειδί συνόδου R1 και επίσης ένα εισιτήριο για την υπηρεσία Service. Το εισιτήριο περιέχει το όνομα του Client, το κλειδί συνόδου (session key/ R1) και ένα χρόνο λήξης. Το εισιτήριο είναι

κρυπτογραφημένο με το ιδιωτικό κλειδί της υπηρεσίας Service (που είναι γνωστό μεταξύ της υπηρεσίας και του AS). Όταν ο Client παραλάβει την authentication response, καταρχήν ελέγχει κατά πόσο ο τυχαίος αριθμός που είχε συμπεριλάβει στην αίτηση ταιριάζει με αυτόν που περιέχεται στο παραληφθέν μήνυμα. Γι' αυτό το σκοπό χρησιμοποιεί το κλειδί του χρήστη (user key) για να ανακτήσει το session key και το ticket. Αφού επιβεβαιώσει ότι η απάντηση ανταποκρίνεται στην αυθεντική αίτηση, αποκλείοντας έτσι την πιθανότητα επίθεσης *replay attack*, συνεχίζει με την επεξεργασία του υπόλοιπου μηνύματος. Το γεγονός ότι τα περιεχόμενα της authentication response ήταν κρυπτογραφημένα με το κλειδί του χρήστη, αποδεικνύει ότι η απάντηση προέρχεται από τον αληθινό AS, ενώ το γεγονός ότι ο Client μπορεί να αποκρυπτογραφήσει τα περιεχόμενα της απάντησης σημαίνει ότι αντιπροσωπεύει τον έγκυρο χρήστη.

3. Τελευταίο βήμα είναι η σύνδεση του Client με την υπηρεσία του δικτύου με την αποστολή μιας αίτησης εξυπηρέτησης. Η παροχή μόνο του ticket στην αίτηση εξυπηρέτησης δεν αποτελεί ικανοποιητικό στοιχείο για την απόδειξη της ταυτότητας του Client. Το ticket μπορεί να χρησιμοποιηθεί από εισβολέα που έχει καταγράψει την διακινούμενη πληροφορία. Η συνοδεία του ticket με επιπλέον πληροφορία (authenticator) που είναι δεμένη με την ταυτότητα του Client, εξασφαλίζει ολοκληρωμένη επαλήθευση. Στο authenticator περιλαμβάνεται ένα άθροισμα ελέγχου (checksum). Συνήθως είναι μια σύνοψη μηνύματος κρυπτογραφημένη με το session key ή άλλο κλειδί.

Αδυναμίες του Kerberos

Το Kerberos δεν έχει την δυνατότητα να προστατέψει ένα δίκτυο από κάθε είδους απειλή. Λειτουργεί βάσει συγκεκριμένων υποθέσεων όσον αναφορά την υποκείμενη δικτυακή δομή.

- Επιθέσεις του τύπου άρνησης εξυπηρέτησης (denial of service attack) δεν μπορούν να αντιμετωπιστούν με το Kerberos. Ένας εισβολέας μπορεί εκμεταλλευόμενος τις αδυναμίες του συστήματος να αποτρέψει έναν server από το να συμμετέχει στα κανονικά βήματα πιστοποίησης. Η ανίχνευση και η επιδιόρθωση τέτοιων καταστάσεων αφήνεται στα χέρια των διαχειριστών και των χρηστών.

- Οι χρήστες πρέπει να κρατούν τους κωδικούς τους μυστικούς. Το Kerberos δεν είναι σε θέση να προστατέψει το δίκτυο από ασυνείδητους χρήστες που μοιράζουν τους κωδικούς τους ή που δεν είναι αρκετά προσεκτικοί για να τον κρατήσουν κρυφό.
- Επιθέσεις που βασίζονται στην πρόβλεψη εύκολων κωδικών (password guessing attack) δεν αντιμετωπίζονται από τον Kerberos. Ένας εισβολέας με χρήση ενός λεξικού, μπορεί εύκολα να "σπάσει" μικρούς και εύκολους κωδικούς που αποτελούνται από λέξεις που μπορούν να βρεθούν σε λεξικό.
- Κάθε μηχανή του δικτύου πρέπει να έχει ένα καλά ρυθμισμένο ρολόι. Μηχανές με ρυθμίσεις ώρας που διαφέρουν σημαντικά (πάνω από 5 λεπτά) μπορεί να δημιουργήσουν πρόβλημα στην πιστοποίηση των timestamps που εμπεριέχονται στα μηνύματα. Έτσι, ένας εισβολέας εκμεταλλευόμενος αυτή την αδυναμία μπορεί να πραγματοποιήσει επίθεση επανάληψης (replay attack). Ή ακόμα βρίσκοντας τον απαραίτητο χρόνο, να σπάσει αδύναμους κωδικούς χρηστών.

4 Προσομοίωση αλγορίθμων εμπιστοσύνης

4.1 Γενικά

Η διαχείριση φήμης (reputation management, RM) υιοθετείται στα κατανεμημένα peer-to-peer δίκτυα για να βοηθήσει τους χρήστες να υπολογίσουν ένα μέτρο της εμπιστοσύνης προς τους υπόλοιπους χρήστες. Αυτές οι τιμές εμπιστοσύνης επηρεάζουν το πώς ή με ποιον ένας χρήστης θα αλληλεπιδράσει (συναλλαχθεί). Η υπάρχουσα βιβλιογραφία RM εστιάζει πρώτιστα στην ανάπτυξη των αλγορίθμων και όχι στη συγκριτική τους ανάλυση. Στο κεφάλαιο αυτό, σκοπός μας είναι να προσομοιώσουμε πειραματικά τέτοιους αλγορίθμους και εν συνεχεία να συγκρίνουμε τις επιδόσεις τους. Θα παρουσιάσουμε, λοιπόν, ένα πλαίσιο αξιολόγησης βασισμένο στο παράδειγμα trace-simulator (προσομοιωτής ίχνους).

Με την παραγωγή αρχείων trace εξομοιώνονται ποικίλες μορφές δικτύων και δίνεται ιδιαίτερη προσοχή στη διαμόρφωση της κακόβουλης συμπεριφοράς των χρηστών. Η προσομοίωση είναι βασισμένη στα αρχεία trace, ενώ επιπρόσθετες μέθοδοι υπολογισμού της εμπιστοσύνης αναπτύσσονται, ώστε να επιτρέψουν την προσομοίωση σε δίκτυα πραγματικού μεγέθους. Το πλαίσιο και ο προσομοιωτής που περιγράφονται είναι διαθέσιμα ως open source, έτσι ώστε οι ερευνητές να μπορούν να αξιολογήσουν την αποτελεσματικότητα άλλων τεχνικών διαχείρισης φήμης ή/και να επεκτείνουν τη λειτουργικότητά τους.

Όπως αναφέραμε, η έρευνα σχετικά με τη διαχείριση εμπιστοσύνης έχει στραφεί στην ανάπτυξη αλγορίθμων και λίγη προσοχή δίνεται στην ποσοτική συγκριτική ανάλυση μεταξύ των υπαρχόντων αλγορίθμων φήμης (οι ποιοτικές αναλύσεις εμφανίζονται συχνά, αλλά είναι ανεπαρκείς). Δοκιμές σε μερικά συστήματα χρησιμοποιούν ιδιόκτητους προσομοιωτές και closed-source κώδικα, όπως στο EigenTrust (Schlosser, Condie, & Kamvar, 2003) ενώ σε άλλα υπάρχει απλά μια θεωρητική περιγραφή του συστήματος, με ελάχιστα ή και καθόλου πειραματικά δεδομένα, όπως στο ROCQ. Προκειμένου να συγκριθούν τα συστήματα αυτά και να ελεγχθούν τα λεγόμενα των συγγραφέων απαιτείται ένας αντικειμενικός προσομοιωτής.

Παρότι υπάρχουν προσομοιωτές p2p δικτύων, το επιπλέον βάρος της προσομοίωσης των κατανεμημένων πινάκων κατακερματισμού DHT, κάνει τη χρήση τους υπολογιστικά ακατάλληλη. Επιπρόσθετα, τέτοιοι προσομοιωτές στηρίζονται σε αφηρημένες ιδέες, προκαλώντας δυσκολίες στην υλοποίησή τους. Επομένως, φαντάζει αναγκαία η ύπαρξη ενός προσομοιωτή που δεν θα έχει όλα τα παραπάνω μειονεκτήματα, αλλά θα είναι εύχρηστος, αποτελεσματικός και συμβατός με όλους τους αλγόριθμους φήμης.

4.2 P2P Simulator

Υπάρχουν πολλές προκλήσεις στην οικοδόμηση ενός πλαισίου αξιολόγησης γενικού σκοπού για τα συστήματα RM. Πρώτον, η RM χρησιμοποιείται σε ποικίλες δικτυακές αρχιτεκτονικές όπως τα p2p, τα service-oriented (SOA) και τα social δίκτυα. Δεύτερον, υπάρχει χάσμα μεταξύ προσομοίωσης της ρεαλιστικής συμπεριφοράς και της υπερβολικής παραμετροποίησης. Γι' αυτό χρειάζεται ένας βατός συμβιβασμός για την παραγωγή τιμών εμπιστοσύνης με ακρίβεια. Τρίτον, είναι μια πρόκληση να καθοριστούν τα μοντέλα συμπεριφοράς χρηστών, ειδικά εκείνα των κακόβουλων. Ενώ οι καλοί χρήστες συμπεριφέρονται κατά τρόπο προβλέψιμο, οι κακόβουλοι χρήστες, ειδικά εκείνοι που ενεργούν κατά τρόπο συλλογικό, μπορούν να συμπεριφερθούν ακανόνιστα και δυναμικά.

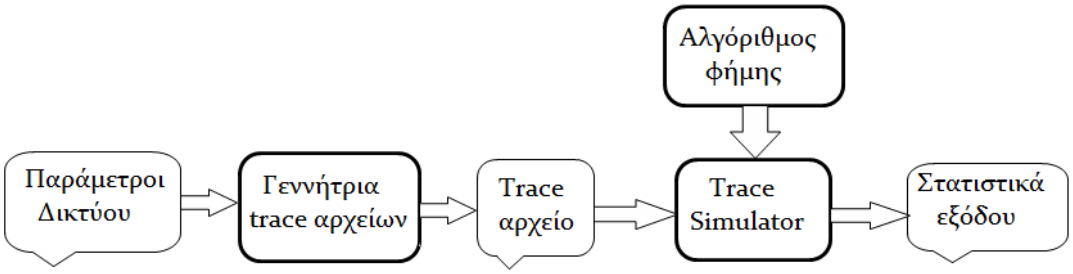
Στην ενότητα αυτή θα εξετάσουμε ακριβώς αυτές τις προκλήσεις, παρουσιάζοντας τον P2P Simulator. Ο P2P Simulator αναπτύχθηκε και υλοποιήθηκε στις γλώσσες προγραμματισμού Java και C, από τον Andrew West του πανεπιστημίου της Πενσυλβανία με σκοπό την αξιολόγηση των αλγορίθμων διαχείρισης εμπιστοσύνης. Αποτελείται από δύο κύρια στοιχεία : (1) Μια γεννήτρια trace που βγάζει στην έξοδο αρχεία trace που περιγράφουν την αρχικοποίηση του δικτύου και τις αλληλεπιδράσεις, και (2) Έναν προσομοιωτή που παίρνει ως είσοδο ένα αρχείο trace και τον αλγόριθμο εμπιστοσύνης, εκτελεί την προσομοίωση του δικτύου και βγάζει στην έξοδο στατιστικά αποτελέσματα σχετικά με τη συμπεριφορά του διαχειριστή εμπιστοσύνης. Ο P2P Simulator θεωρείται ανάμεσα στους ανταγωνιστές του ο καλύτερος και αποτελεσματικότερος προσομοιωτής reputation-based p2p δικτύων. Στο υπόλοιπο της ενότητας θα προσπαθήσουμε να παρουσιάσουμε με λεπτομέρεια τα στοιχεία του P2P Simulator καθώς και τον τρόπο με τον οποίο λειτουργεί.

4.2.1 Αρχιτεκτονική προσομοιωτή

Θεωρούμε ότι το σύστημα που προσομοιώνουμε αποτελείται από *χρήστες* ή *ομότιμους κόμβους*. Αυτοί οι χρήστες είναι μέρος ενός δικτύου. Ορισμένα ζευγάρια των χρηστών έχουν ένα κανάλι επικοινωνίας μεταξύ τους. Στην θεωρία των γράφων, οι χρήστες είναι οι κόμβοι και το κανάλι η γραμμή που τους συνδέει. Οποιαδήποτε στιγμή, ένας χρήστης μπορεί να λειτουργεί ως *προμηθευτής* (server), ως *αιτών* (client) ή και τα δύο. Τα αντικείμενα που αιτούνται και μεταφορτώνονται καλούνται *αρχεία*. Τα αρχεία αυτά είναι είτε *έγκυρα* είτε *ψεύτικα* και υποθέτουμε ότι οι χρήστες είναι σε θέση να προσδιορίσουν την εγκυρότητα ενός αρχείου. Η ισχύς ενός αρχείου είναι μόνιμη, ενώ πολλαπλά αντίγραφα του ίδιου αρχείου μπορούν να υπάρξουν σε ένα δίκτυο. Οι χρήστες αξιολογούνται σύμφωνα με την *ποιότητα των αρχείων* που παρέχουν σε άλλους χρήστες. Τα αρχεία καταχωρούνται στη *βιβλιοθήκη* ενός χρήστη και ένας χρήστης εισχωρεί στο δίκτυο με μια αρχική βιβλιοθήκη. Κάθε χρήστης που θέλει να ψάξει ένα συγκεκριμένο αρχείο εκτελεί broadcast στο δίκτυο, που αποκαλείται *query*, για να καθορίσει το σύνολο των ιδιοκτητών που κατέχουν το αρχείο αυτό. Μετά από κάθε *συναλλαγή,* ο χρήστης που ζήτησε και μεταφόρτωσε το αρχείο στέλνει *feedback* σχετικά με τον προμηθευτή και την ποιότητα των υπηρεσιών του. Τέλος, ένας χρήστης είναι σε θέση να αφαιρέσει αρχεία από τη βιβλιοθήκη του, μια ενέργεια που καλείται *καθαρισμός*.

Τώρα θα περιγράψουμε την υποδομή στην οποία η εφαρμογή του P2P Simulator είναι βασισμένη. Όπως γίνεται κατανοητό και από την ορολογία του δικτύου, ένα ελλοχεύον p2p δίκτυο (π.χ. ένα δίκτυο συναλλαγής αρχείων συγγενές του Gnutella) είναι το καταλληλότερο. Παραδείγματος χάριν, υποθέστε ότι κάποιος θέλει να εξετάσει ένα σύστημα με αποκλειστικό σύνολο χρηστών και προμηθευτών, όπου οι χρήστες χρησιμοποιούν RM για να καθορίσουν την αξιοπιστία προμηθευτών. Με μόνο παρά ελάχιστες τροποποιήσεις ένα p2p πλαίσιο μπορεί να εξομοιώσει ένα τέτοιο μοντέλο. Η συγκρισιμότητα των αλγορίθμων φήμης πραγματοποιείται μέσω της προσέγγισης του trace-simulator. Τα αρχεία trace είναι στατικής φύσεως, προ-παράγονται και δεν τροποποιούνται κατά τη διάρκεια της φάσης προσομοίωσης. Η παραγωγή των αρχείων trace κωδικοποιεί τις παραμέτρους του δικτύου και είναι ανεξάρτητη από τους αλγόριθμους φήμης που δοκιμάζονται. Τα αρχεία trace στη συνέχεια δίνονται στον προσομοιωτή, ο οποίος υλοποιεί τον αλγόριθμο φήμης και άλλες δυναμικές εκτιμήσεις. Κατά συνέπεια, πολλαπλές προσομοιώσεις που

υλοποιούν διάφορους αλγόριθμους φήμης μπορούν να εκτελεστούν χρησιμοποιώντας το ίδιο ακριβώς αρχείο trace. Το σχήμα 4.1 απεικονίζει αυτήν την γενική αρχιτεκτονική του P2P Simulator.



Σχήμα 4.1 : Αρχιτεκτονική του P2P Simulator

4.2.2 Γεννήτρια αρχείων trace

Η γεννήτρια ιχνών είναι ένα πρόγραμμα που παίρνει τις δικτυακές παραμέτρους και δίνει στην έξοδο ένα στατικό αρχείο εντολών που χρησιμοποιείται για την προσομοίωση του δικτύου. Στις προσομοιώσεις μας, μοντελοποιούμε ένα δίκτυο χρηστών. Υπάρχει μια βιβλιοθήκη με μοντέλα χρηστών, όπου μερικά αντιστοιχούν σε καλούς χρήστες και άλλα σε κακόβουλους. Οι παράμετροι εισόδου μας δείχνουν πόσοι χρήστες αντιστοιχίζονται ανά μοντέλο. Ένα αξίωμα του προσομοιωτή αυτού είναι πως υπάρχει μια σύνδεση μεταξύ κάθε ζευγαριού χρηστών. Σε κάθε χρήστη δίνεται επίσης μια αρχική τροφοδότηση αρχείων, μερικά έγκυρα και άλλα άκυρα. Όπως τονίσαμε και προηγουμένως μπορούν να υπάρξουν πολλά αντίγραφα οποιουδήποτε αρχείου.

Γενικά, ένα ίχνος είναι μια ακολουθία από queries. Κάθε query προσδιορίζει ένα αρχείο και τον χρήστη που ζητάει το αρχείο αυτό (client). Η επιλογή του προμηθευτή από τον οποίο θα μεταφορτώσει κάποιος ένα αρχείο γίνεται μέσα στον χρόνο εκτέλεσης της προσομοίωσης, λαμβάνοντας υπόψη την είσοδο από τον αλγόριθμο φήμης. Η συμπεριφορά των χρηστών εκτείνεται σε δύο διαστάσεις. Πρώτον, για κάθε αρχείο που μεταφορτώνουν μπορούν να επιλέξουν είτε να κρατήσουν το αρχείο, είτε να το *καθαρίσουν*, δηλαδή να ξεφορτωθούν το αρχείο. Οι καλοί χρήστες θα τείνουν να καθαρίσουν την βιβλιοθήκη τους από άκυρα αρχεία, μειώνοντας έτσι την

πιθανότητα πολλαπλασιασμού των ψεύτικων αρχείων. Από την άλλη πλευρά, οι κακόβουλοι χρήστες κάνουν ακριβώς το αντίθετο. Υποθέτουμε ότι κανένας καθαρισμός αρχείων δεν συμβαίνει στην αρχικοποίηση του δικτύου και ότι οι χρήστες έχουν μόνο μια ευκαιρία για τον καθαρισμό ενός αρχείου, αμέσως αφότου το μεταφορτώσουν. Δεύτερον, μετά από κάθε συναλλαγή ο χρήστης παρέχει feedback σχετικά με τον προμηθευτή, όσον αφορά την ποιότητα του αρχείου που παρείχε. Οι καλοί χρήστες θα τείνουν να παρέχουν τίμιο feedback, υποβάλλοντας θετική τιμή (+1) εάν έλαβαν ένα έγκυρο αρχείο. Κάποιοι κακόβουλοι χρήστες τείνουν να κάνουν το αντίθετο.

Πριν προχωρήσουμε σε περισσότερο βάθος να επισημάνουμε πως κάνουμε την απλουστευμένη υπόθεση ότι η κακόβουλη συμπεριφορά είναι καθαρά πιθανολογική και ανεξάρτητη από το χρήστη ή το αρχείο που ζητείται σε μια συγκεκριμένη συναλλαγή.

Αρχεία ίχνους

Στόχος του P2P Simulator είναι να κάνει χρήση όσο γίνεται λιγότερης παραμετροποίησης για να προσομοιώσει τα συστήματα RM. Τα κυριότερα ορίσματα που δίνονται στην γραμμή εντολών σχετικά με το δίκτυο είναι :

1. Αριθμός χρηστών στο δίκτυο.
2. Μοντέλο συμπεριφοράς για κάθε χρήστη.
3. Αριθμός διαφορετικών αρχείων.
4. Πιθανότητα ιδιοκτησίας ενός αρχείου.
5. Αριθμός queries/συναλλαγών για προσομοίωση.
6. Μέγιστος αριθμός συνδέσεων για κάθε χρήστη.
7. Περίοδος εύρους ζώνης σε μονάδα χρόνου.

Η έξοδος της γεννήτριας ίχνους είναι ένα σύντομο αρχείο κειμένου με τους εξής τέσσερις ευδιάκριτους τύπους στοιχείων :

1. Επικεφαλίδα: Τα δεδομένα που δόθηκαν ως ορίσματα στην γραμμή εντολών παρουσιάζονται τυπωμένα. Ο προσομοιωτής τα χρειάζεται για να ταξινομήσει τις δομές δεδομένων.

2. Αρχικοποιήσεις χρηστών: Τριπλέτες της μορφής (u, c, h) όπου το u είναι ο χρήστης που αρχικοποιείται, c το ποσοστό επί τοις εκατό που ο χρήστης αφαιρεί ένα άκυρο αρχείο από τη βιβλιοθήκη του (καθαρισμός) και h είναι το ποσοστό επί τοις εκατό που ο χρήστης παρέχει τίμιο feedback.

3. Αρχικοποιήσεις βιβλιοθήκης: Τριπλέτες της μορφής (u, f, v) που δηλώνουν ότι ο χρήστης u έχει το αρχείο f στην αρχικοποιημένη βιβλιοθήκη του με εγκυρότητα v (boolean μεταβλητή).

4. Στατικά queries: Ζευγάρια της μορφής (u, f) που δηλώνουν την επιθυμία του χρήστη u να αποκτήσει το αρχείο f.

Ο πίνακας 4.1 περιγράφει τις αρχικοποιήσεις των διάφορων μοντέλων χρήστη που θα χρησιμοποιήσουμε για τις προσομοιώσεις μας. Η επιλογή που κάνουμε δεν είναι καταναγκαστική καθώς και άλλα μοντέλα χρήστη μπορούν να εφαρμοστούν εύκολα στο πλαίσιο του P2P Simulator.

Τύπος χρήστη	***Καθαρισμός άκυρων αρχείων%***	***Τιμιότητα στο feedback %***
Good	90 - 100 %	100 %
Purely Malicious	0 - 10 %	0 %
Malicious Provider	0 - 10 %	100 %
Feedback Malicious	90 - 100 %	0 %
Disguised Malicious	50 - 100 %	50 - 100 %
Sybil Attacker	0 - 10 %	Ανεξάρτητο

Πίνακας 4.1 : Παράμετροι αρχικοποίησης συμπεριφοράς χρήστη

Ένας καλός χρήστης θέλει απλά να λάβει ένα έγκυρο αντίγραφο του αρχείου που αιτεί. Οι κακοί χρήστες θέλουν να αναμεταδώσουν τα άκυρα αρχεία. Στη συνέχεια περιγράφουμε λεπτομερώς τα μοντέλα χρηστών που υποστηρίζει ο P2P Simulator.

- **Good:** Αναφερόμαστε στους καλούς χρήστες. Κατ' αρχάς, οι καλοί χρήστες παρέχουν τίμιο feedback σε κάθε συναλλαγή τους. Αφετέρου, είναι προσεκτικοί με τις βιβλιοθήκες αρχείων τους, αφαιρώντας τα άκυρα αρχεία που βρίσκονται εκεί. Ένα ποσοστό καθαρισμού μεταξύ 90-100% επιτρέπει κάποιο βαθμό απάθειας, δεδομένου ότι δεν μπορούμε να περιμένουμε από τους όλους τους καλούς χρήστες να είναι ιδανικοί.

- **Purely Malicious:** Ένας χρήστης που συμπεριφέρεται κακόβουλα και στις δύο διαστάσεις καλείται αμιγής κακόβουλος. Τέτοιοι χρήστες διατηρούν τα άκυρα αρχεία, ξεφορτώνονται έγκυρα και ψεύδονται συνεχώς για τη φύση των αρχείων που λαμβάνουν. Επειδή συμπεριφέρονται κακόβουλα με συνεχή συχνότητα, ένας αλγόριθμος εμπιστοσύνης μπορεί γρήγορα να εντοπίσει τέτοιους χρήστες και να λάβει τα κατάλληλα μέτρα.

- **Malicious Provider & Feedback Malicious:** Οι κακόβουλοι προμηθευτές και οι feedback κακόβουλοι είναι συμπληρωματικά μοντέλα χρηστών που συμπεριφέρονται κακόβουλα στη μία μόνο διάσταση. Οι κακόβουλοι προμηθευτές παρέχουν άκυρα αρχεία, αλλά δίνουν τίμιο feedback. Οι feedback κακόβουλοι κάνουν ακριβώς το αντίστροφο. Τέτοιες κακόβουλες στρατηγικές είναι δυσκολότερο να ανιχνευθούν σε σχέση με τις αμιγείς κακόβουλες.

- **Disguised Malicious:** Οι μεταμφιεσμένοι κακόβουλοι χρήστες λειτουργούν άλλοτε ως κακόβουλοι και στις δύο διαστάσεις και άλλοτε ως καλοί χρήστες με σκοπό να ξεγελάσουν το σύστημα. Ειδικά σε συστήματα, όπως το EigenTrust, που στηρίζονται στην κανονικοποίηση των τιμών, οι μεταμφιεσμένοι χρήστες καταφέρνουν να λογίζονται ως καλοί χρήστες κάποιες φορές. Κάποια συστήματα φήμης, όπως το TNA-SL, καταφέρνουν να καταπολεμήσουν τέτοιες κακόβουλες επιθέσεις, κάνοντας χρήση beta-PDF στρατηγικών[6].

- **Sybil Attacker:** Ένας σιβυλλικός χρήστης κατέχει μία άκυρη βιβλιοθήκη και περιμένει έως ότου είναι προμηθευτής σε κάποια συναλλαγή. Έπειτα διαγράφει

[6] Περισσότερες λεπτομέρειες στο **Trust Network Analysis with Subjective Logic,** Audun Jøsang et al

τον λογαριασμό του και επανεμφανίζεται στο δίκτυο με διαφορετικό λογαριασμό και όνομα χρήστη.

- **Malicious collectives:** Μια ομάδα συνεργαζόμενων κακόβουλων χρηστών που αποτελούν την σοβαρότερη απειλή σε ένα αποκεντρωμένο σύστημα. Οι απομονωμένοι κακόβουλοι χρήστες συμμετέχουν σε τυχαία συνεργασία, παραδείγματος χάριν, όταν παρέχουν θετικό feedback σε έναν άγνωστο χρήστη που τους έστειλε ένα άκυρο αρχείο. Εντούτοις, η οργανωμένη κακόβουλη συνεργασία, που χαρακτηρίζεται από ευφυείς και peer-aware στρατηγικές είναι ο πραγματικός κίνδυνος.

Αρχικοποίηση βιβλιοθήκης

Η σύνθεση της βιβλιοθήκης διαμορφώνεται ως μια κατανομή Zipf [7] (Zipf, 1949). Η παράμετρος Zipf a παρέχεται από τη γραμμή εντολών, έτσι ώστε το αρχείο i να έχει πιθανότητα $(1/i^a)$ να ανήκει σε έναν συγκεκριμένο χρήστη. Η εγκυρότητα ενός αρχικού αρχείου καθορίζεται από το ποσοστό καθαρισμού του χρήστη που του ανήκει. Παραδείγματος χάριν, ένας χρήστης με ποσοστό καθαρισμού c, έχει πιθανότητα c% να έχει στην κατοχή του ένα έγκυρο αρχικό αρχείο. Ο καθαρισμός δεν εκτελείται κατά τη διάρκεια της αρχικοποίησης της βιβλιοθήκης.

Κάτω από αυτές τις συνθήκες αναμένεται κάθε χρήστης να έχει μια αρχικοποιημένη βιβλιοθήκη του ίδιου μεγέθους. Αυτό είναι μια μη ρεαλιστική υπόθεση αλλά πιστεύουμε ότι δεν έχει σημαντικές επιπτώσεις στα εμπειρικά αποτελέσματα.

Η επιλογή μιας καλής τιμής για την παράμετρο a είναι σημαντική. Διάφορες μελέτες που έχουν γίνει αντικρούονται μεταξύ τους. Κάποιες μελέτες προτείνουν παράμετρο Zipf ίση με 0.8 ή μεγαλύτερη. Σε κάποιες περιπτώσεις μια τόσο υψηλή τιμή του a μπορεί να είναι προβληματική. Για παράδειγμα, αν κάποιος επιθυμεί να εκτελέσει έναν μεγάλο αριθμό από queries πρέπει να υπάρχουν ικανοποιητικοί πόροι, έτσι ώστε κάθε χρήστης να μην αποκτήσει κάθε αρχείο. Για να αποφύγουμε ένα τέτοιο πρόβλημα, στις προσομοιώσεις μας χρησιμοποιούμε προκαθορισμένη τιμή της παραμέτρου Zipf ίση με 0.4.

[7] http://en.wikipedia.org/wiki/Zipf's_law

Παραγωγή queries

Το τελικό ερώτημα είναι ποιος θα πρέπει να ζητάει αρχεία, ποια αρχεία και σε τι ποσότητα. Η γεννήτρια του P2P Simulator υποστηρίζει δύο τρόπους τους οποίους καλούμε *naive* (αφελής) και *intelligent* (ευφυής) παραγωγή queries. Στη naive έκδοση, ένας τυχαίος χρήστης ζητά ένα τυχαίο αρχείο και αυτό καταγράφεται ως query στο αρχείο trace. Στην intelligent έκδοση υπάρχουν επιπρόσθετα στοιχεία. Κατ' αρχάς, ένας χρήστης δεν μπορεί να ζητήσει ένα αρχείο που κατέχει ήδη ή ζήτησε στο παρελθόν. Αυτό εξαλείφει την δυνατότητα σε οποιοδήποτε χρήστη να κρατάει πολλαπλά αντίγραφα του ίδιου αρχείου. Επιπλέον, ένα ζητούμενο αρχείο πρέπει να υπάρχει στο δίκτυο. Ένα query που δεν επιστρέφει κανένα αποτέλεσμα είναι ανακόλουθο. Αυτό εντούτοις, δεν αποτελεί εγγύηση ότι ένα αρχείο θα είναι διαθέσιμο όταν ζητείται, δεδομένου ότι όλοι οι ιδιοκτήτες του αρχείου έχουν προκαθορισμένο εύρος ζώνης.

Κάθε χρήστης έχει ίση πιθανότητα με οποιονδήποτε άλλον χρήστη να ζητήσει κάποιο αρχείο. Το ποιο αρχείο ζητείται υπαγορεύεται από την ίδια κατανομή Zipf που χρησιμοποιείται για την αρχικοποίηση των βιβλιοθηκών. Στο Παράρτημα Α υπάρχει παράδειγμα εκτέλεσης παραγωγής αρχείου ίχνους, καθώς και άλλες λεπτομέρειες για την γεννήτρια trace.

4.2.3 Προσομοίωση αρχείων trace

Στην ενότητα αυτή περιγράφουμε τον *προσομοιωτή*, στον οποίο γίνεται το «τρέξιμο» του αρχείου ίχνους δυναμικά και παίρνουμε σαν έξοδο τα σχετικά στατιστικά αποτελέσματα. Αφότου γίνει η αρχικοποίηση του συστήματος, ο προσομοιωτής τρέχει τον απλουστευμένο βρόχο που δίνεται από τον ψευδοκώδικα του σχήματος 4.2.

Κατανομή φόρτου και εύρος ζώνης

Με τη χρήση ενός αλγόριθμου εμπιστοσύνης είναι δυνατή η σύγκλιση της παγκόσμιας τιμής εμπιστοσύνης. Έτσι, ένα μικρό σύνολο χρηστών μπορεί να αναγνωριστεί ως το περισσότερο «αξιόπιστο». Αυτοί οι κόμβοι το πιθανότερο είναι να κατακλυστούν με queries.

While υπάρχουν περισσότερα queries **do**:

Διάβασε το query από το αρχείο trace;

Κάνε broadcast το query για αναζήτηση πιθανών προμηθευτών;

Υπολόγισε τις τιμές εμπιστοσύνης των σχετικών χρηστών;

Επέλεξε προμηθευτή με διαθέσιμο εύρος ζώνης;

Αντέγραψε το αρχείο στην βιβλιοθήκη του αιτούντος;

Ο αιτών υποβάλλει feedback όσον αφορά τον προμηθευτή;

End

Σχήμα 4.2 : Ψευδοκώδικας P2P Simulator

Όμως, οι περιορισμοί στο εύρος ζώνης θα αποτρέψουν μερικά από αυτά τα αιτήματα από το να πραγματοποιηθούν ή θα εξυπηρετήσουν τα αρχεία αυτά σε πολύ αργό ρυθμό (φτωχό QoS) (Papaioannou & Stamoulis, 2004). Η κατανομή του φόρτου εργασίας είναι μια πολύ σημαντική παράμετρος για τα δίκτυα. Από το Κεφάλαιο 2, είδαμε πως από την μία πλευρά συστήματα όπως το EigenTrust χειρίζονται την κατανομή φόρτου και εύρους ζώνης, ενώ άλλα όπως το ROCQ δεν δίνουν στο θέμα αυτό καμία προσοχή.

Η εξισορρόπηση του φόρτου μειώνει εγγενώς την απόδοση του αλγορίθμου φήμης. Η μείωση του φορτίου των έμπιστων χρηστών σημαίνει ότι το φορτίο θα αυξηθεί για τους λιγότερο έμπιστους χρήστες και πιθανολογικά μιλώντας, περισσότερα άκυρα αρχεία θα συναλλαχθούν στο δίκτυο. Ο P2P Simulator κάνει χρήση μιας απλής στρατηγικής. Ο αλγόριθμος εμπιστοσύνης εξάγει την σχετική λίστα με τους προμηθευτές βασισμένη στις τιμές εμπιστοσύνης τους. Έπειτα, ο διαχειριστής εύρους ζώνης του προσομοιωτή επιτρέπει με αντικειμενικό τρόπο μόνο στους χρήστες με διαθέσιμο εύρος ζώνης να συμμετέχουν στις συναλλαγές.

Οι περιορισμοί εύρους ζώνης τίθενται με δύο παραμέτρους της γραμμής εντολών που δίνονται στο Παράρτημα Α. Συνοπτικά, ένας χρήστης μπορεί να έχει μέγιστο αριθμό συνδέσεων X οποιαδήποτε στιγμή και η συναλλαγή ενός αρχείου να απαιτεί Y μονάδες χρόνου για να ολοκληρωθεί. Παραδείγματος χάριν, εάν X = 2 και Y = 100, ένας χρήστης που αρχίζει μία μεταφόρτωση την χρονική στιγμή 12 και άλλη μία την χρονική στιγμή 30, θα πρέπει να περιμένει μέχρι την χρονική στιγμή 112, ώστε να

κάνει και άλλη σύνδεση. Εμείς για τις προσομοιώσεις μας θα κάνουμε χρήση των προεπιλεγμένων τιμών του P2P Simulator (X=2, Y=1). Δηλαδή, κάθε χρήστης μπορεί να κάνει δύο ταυτόχρονες συνδέσεις, κάθε query απαιτεί μία μονάδα χρόνου και μόνο ένα query επιτρέπεται σε κάθε κύκλο ρολογιού.

Κακόβουλες στρατηγικές

Ένα ακόμα θέμα που διαπραγματεύεται ο P2P Simulator είναι ο τρόπος με τον οποίο οι κακόβουλοι χρήστες βλάπτουν το σύστημα. Δηλαδή, τι στρατηγική χρησιμοποιούν προκειμένου είτε να γεμίζουν με άκυρα αρχεία το σύστημα, είτε να υποβάλλουν ανέντιμο feedback. Στον P2P Simulator η στρατηγική των κακόβουλων χρηστών δίνεται από την γραμμή εντολών (Παράρτημα Α) και μπορεί να είναι μία από τις ακόλουθες :

- **Naive:** Στη στρατηγική αυτή, οι κακόβουλοι χρήστες χαρακτηρίζονται ως αφελείς. Έστω ότι ο χρήστης u λαμβάνει ένα έγκυρο αρχείο από έναν καλό χρήστη v και υποβάλλει ανέντιμα αρνητικό feedback. Τώρα ο χρήστης u την επόμενη φορά που θα κάνει κάποιο query και θα χρειαστεί να υπολογίσει τις τιμές εμπιστοσύνης, θα πάει στη βάση δεδομένων feedback και θα συναθροίσει όλα τα feedbacks συμπεριλαμβανομένου και αυτού που έκανε ο ίδιος προηγουμένως. Ειδικότερα, η εμπιστοσύνη trust(u→v) μπορεί να υπολογιστεί χαμηλή με αποτέλεσμα ο κακόβουλος χρήστης u να χρησιμοποιήσει τον v ως πηγή. Εντούτοις, ο v είναι πραγματικά καλός χρήστης, ο u παίρνει ένα έγκυρο αρχείο και έχει πλέον λιγότερα άκυρα αρχεία, άρα το δίκτυο είναι ελαφρώς καλύτερο. Επομένως, *στη naive στρατηγική χρησιμοποιούνται αποκλειστικά τα καθολικά δεδομένα αλληλεπιδράσεων*.

- **Isolated:** Στην *απομονωμένη* στρατηγική οι κακόβουλοι χρήστες είναι ένα βήμα πιο ευφυείς. Υποθέτουμε εδώ, ότι ένας χρήστης που θέλει να κάνει αναζήτηση των τιμών εμπιστοσύνης θα εξαγάγει ένα αντίγραφο του συγκεντρωμένου feedback της βάσης δεδομένων και θα υπολογίσει τις τιμές εμπιστοσύνης τοπικά στο μηχάνημα του. Περαιτέρω, εκτός από την υποβολή -πιθανώς ανέντιμων- feedback οι χρήστες θα αποθηκεύουν τοπικά ένα διάνυσμα του ιστορικού των απόλυτα τίμιων αλληλεπιδράσεων. Επανερχόμενοι στον κακόβουλο χρήστη u, τώρα όταν θα

συνυπολογίσει την τιμή εμπιστοσύνης του χρήστη v, δε θα λάβει υπόψη το feedback που υπέβαλλε για εκείνον προηγουμένως. Επομένως, *στην isolated στρατηγική το τοπικό ιστορικό τίμιων αλληλεπιδράσεων επικαλύπτει το καθολικό.*

- **Collective:** Στη συλλογική κακόβουλη στρατηγική οι κακόβουλοι χρήστες είναι πιο ευφυείς και αποτελεσματικοί. Συνεργάζονται μεταξύ τους μοιράζοντας ο ένας κακόβουλος χρήστης στον άλλον τα τοπικά διανύσματα του ιστορικού τίμιων αλληλεπιδράσεων.

Επιλογή προμηθευτή

Μετά τον υπολογισμό της εμπιστοσύνης, ο αιτών (requester) έχει την ευθύνη να επιλέξει από την σχετική λίστα προμηθευτών τον κατάλληλο. Στον πίνακα 4.2 παρουσιάζεται ο επιθυμητός προμηθευτής ανάλογα με το μοντέλο requester.

Μοντέλο Requester	***Προμηθευτής***
Good	ΚΑΛΥΤΕΡΟΣ
Purely Malicious	ΧΕΙΡΟΤΕΡΟΣ
Malicious Provider	ΧΕΙΡΟΤΕΡΟΣ
Feedback Malicious	ΤΥΧΑΙΟΣ
Disguised Malicious	ΤΥΧΑΙΟΣ
Sybil Attacker	ΧΕΙΡΟΤΕΡΟΣ

Πίνακας 4.2 : Επιλογή προμηθευτή ανάλογα με το μοντέλο requester

Όσοι χρήστες έχουν ως στόχο να αυξήσουν το πεδίο της μη έγκυρης βιβλιοθήκης τους προσπαθούν σε κάθε συναλλαγή να κάνουν τη χειρότερη δυνατή επιλογή. Δηλαδή, επιλέγουν τον προμηθευτή με την μικρότερη τιμή εμπιστοσύνης (ΧΕΙΡΟΤΕΡΟΣ). Κακόβουλοι χρήστες που νοιάζονται μόνο να δώσουν ανέντιμο feedback ακολουθούν μια τυχαία προσέγγιση στην επιλογή προμηθευτών (ΤΥΧΑΙΟΣ). Αντιθέτως, οι καλοί χρήστες επιθυμούν τον πιο έμπιστο προμηθευτή (ΚΑΛΥΤΕΡΟΣ). Υπενθυμίζουμε σε αυτό το σημείο πως μόνο χρήστες με διαθέσιμο εύρος ζώνης μπορούν να χρησιμοποιηθούν ως προμηθευτές. Τέλος, η επιλογή ανάμεσα σε δύο χρήστες με ίδιες τιμές εμπιστοσύνης γίνεται με τυχαίο τρόπο.

4.2.4 Μέτρα αξιολόγησης

Κύριος στόχος του P2P Simulator του A.West είναι να μπορεί να συγκρίνει την αποτελεσματικότητα των αλγόριθμων εμπιστοσύνης. Ακολούθως, καθορίζουμε ένα σαφή και συνοπτικό μέτρο για τον σκοπό αυτό. Εντούτοις, το πλαίσιο του προσομοιωτή επιφέρει και άλλες ευκαιρίες αξιολόγησης των αλγορίθμων. Ειδικότερα, μπορεί να εξεταστεί η αποδοτικότητα του αλγορίθμου και να δοκιμαστούν επιταχυντικές στρατηγικές.

$$Metric = \frac{\#\,\text{έγκυρων αρχείων που μεταφόρτωσαν καλοί χρήστες}}{\#\,\text{συναλλαγών που συμμετείχαν καλοί χρήστες}}$$

4.3 Υλοποίηση αλγορίθμων φήμης στον P2P Simulator

Η υλοποίηση ενός αλγορίθμου φήμης στον P2P Simulator συνίσταται από δύο μέρη. Τα δύο μέρη αυτά είναι ουσιαστικά δύο συναρτήσεις ή ρουτίνες. Το πρώτο μέρος είναι η υλοποίηση μιας συνάρτησης ενημέρωσης των τιμών εμπιστοσύνης. Έπειτα από κάθε αλληλεπίδραση ο client στέλνει feedback σχετικά με τον προμηθευτή με αποτέλεσμα να μεταβάλλεται η τιμή εμπιστοσύνης του τελευταίου. Γι' αυτό μετά από κάθε συναλλαγή ο P2P Simulator καλεί την ρουτίνα ενημέρωσης των τιμών feedback **trans_alg_update( )**. Το δεύτερο μέρος είναι η υλοποίηση μιας συνάρτησης υπολογισμού των τιμών εμπιστοσύνης. Προτού ξεκινήσει μια συναλλαγή, ο client έχει να διαλέξει κάποιον ανάμεσα στους πιθανούς προμηθευτές του αρχείου. Για τους προμηθευτές αυτούς, είναι αναγκαίος ο υπολογισμός των τιμών εμπιστοσύνης τους, ώστε ο αιτών να κάνει την επιλογή του. Η ρουτίνα υπολογισμού των τιμών εμπιστοσύνης στον P2P Simulator είναι η **trans_alg_compute( )**.

Στη συνέχεια της ενότητας προχωρούμε στην προσομοίωση τριών από τους αλγόριθμους εμπιστοσύνης που παρουσιάσαμε στο Κεφάλαιο 2. Τον EigenTrust, τον ROCQ και έναν Bayesian αλγόριθμο. Στόχος μας είναι να παρουσιάσουμε την συγκριτική αξιολόγηση αυτών των αλγορίθμων φήμης με την βοήθεια του P2P Simulator. Αυτό θα γίνει στην ενότητα 4.4.

4.3.1 Προσομοίωση EigenTrust

Η ιδέα υλοποίησης του EigenTrust είναι αυτή που δόθηκε στον αλγόριθμο Basic EigenTrust της σελ.37. Η υλοποίηση του EigenTrust είναι βασισμένη στον πολλαπλασιασμό κανονικοποιημένων πινάκων για τον συνυπολογισμό της εμπιστοσύνης, έτσι ώστε να υπάρχει στο σύστημα μια καθολική εικόνα σύγκλισης. Τα feedback που στέλνουν οι χρήστες του συστήματος φυλάσσονται στον πίνακα διανυσμάτων A. Εκεί συλλέγονται οι αρνητικές και οι θετικές τιμές feedback. Το στοιχείο a_{ij} αντιστοιχεί στη τιμή feedback που έχει υποβάλλει ο χρήστης i για τον προμηθευτή j.

$$A = \begin{bmatrix} \begin{pmatrix} pos:0 \\ neg:0 \end{pmatrix} & \begin{pmatrix} pos:9 \\ neg:3 \end{pmatrix} & \begin{pmatrix} pos:2 \\ neg:4 \end{pmatrix} \\ \begin{pmatrix} pos:3 \\ neg:1 \end{pmatrix} & \begin{pmatrix} pos:0 \\ neg:0 \end{pmatrix} & \begin{pmatrix} pos:5 \\ neg:4 \end{pmatrix} \\ \begin{pmatrix} pos:3 \\ neg:2 \end{pmatrix} & \begin{pmatrix} pos:8 \\ neg:1 \end{pmatrix} & \begin{pmatrix} pos:0 \\ neg:0 \end{pmatrix} \end{bmatrix}$$

Κάθε στοιχείο του A είναι μια ακέραια μεταβλητή feedback (fback_int) που υπολογίζεται από τον εξής απλό αλγόριθμο :

```
fback_int := pos-neg;
if (fback_int < 0) fback_int := 0
```

$$A = \begin{bmatrix} \begin{pmatrix} pos:0 \\ neg:0 \end{pmatrix} = 0 & \begin{pmatrix} pos:9 \\ neg:3 \end{pmatrix} = 6 & \begin{pmatrix} pos:2 \\ neg:4 \end{pmatrix} = 0 \\ \begin{pmatrix} pos:3 \\ neg:1 \end{pmatrix} = 2 & \begin{pmatrix} pos:0 \\ neg:0 \end{pmatrix} = 0 & \begin{pmatrix} pos:5 \\ neg:4 \end{pmatrix} = 1 \\ \begin{pmatrix} pos:3 \\ neg:2 \end{pmatrix} = 1 & \begin{pmatrix} pos:8 \\ neg:1 \end{pmatrix} = 7 & \begin{pmatrix} pos:0 \\ neg:0 \end{pmatrix} = 0 \end{bmatrix}$$

Κατόπιν, υπολογίζουμε τον κανονικοποιημένο πίνακα διανυσμάτων. Η κανονικοποίηση γίνεται για κάθε χρήστη (γραμμή) παίρνοντας το feedback που έχει δώσει για κάποιον προμηθευτή (στήλη) και διαιρώντας το, με το συνολικό feedback που έχει υποβάλλει ο χρήστης αυτός.

$$A' = \begin{bmatrix} {}^{0}/_{6} & {}^{6}/_{6} & {}^{0}/_{6} \\ {}^{2}/_{3} & {}^{0}/_{3} & {}^{1}/_{3} \\ {}^{1}/_{8} & {}^{7}/_{8} & {}^{0}/_{8} \end{bmatrix}$$

Στη συνέχεια αρχικοποιούμε το διάνυσμα p των pre-trusted κόμβων του συστήματος. Οι pre-trusted χρήστες είναι ένα υποσύνολο των καλών χρηστών και αποτελούν μια a priory έννοια της εμπιστοσύνης. Έστω, ότι στο σύστημα υπάρχουν z pre-trusted χρήστες, τότε τα στοιχεία p_i του πίνακα διανυσμάτων p θα είναι ίσα με $p_i = 1/z$, εάν ο χρήστης i είναι pre-trusted, αλλιώς $p_i = 0$. Στην περίπτωση που δεν υπάρχουν pre-trusted χρήστες, κάθε στοιχείο του πίνακα θα είναι ίσο με $1/n$ (όπου n ο αριθμός των κόμβων του δικτύου).

$$p = \begin{bmatrix} {}^{1}/_{3} \\ {}^{1}/_{3} \\ {}^{1}/_{3} \end{bmatrix}$$

Ο αλγόριθμος φήμης EigenTrust κάνει χρήση δύο σημαντικών παραμέτρων. Την παράμετρο a που είναι μια σταθερά μικρότερη του 1 και δείχνει την επιρροή των pre-trusted χρηστών στον υπολογισμό της εμπιστοσύνης και την παράμετρο ε που ισοδυναμεί με το επιτρεπτό απόλυτο σφάλμα μεταξύ δύο διαδοχικών τιμών εμπιστοσύνης ενός χρήστη. Για τις προσομοιώσεις μας έχουμε θέσει προκαθορισμένες τιμές : a=0.5 και ε=0.001.

Πριν από κάθε αλληλεπίδραση ο αλγόριθμος EigenTrust θα πάει να υπολογίσει τα στοιχεία t_i του μονοδιάστατου πίνακα t , όπου φυλάσσονται οι τιμές εμπιστοσύνης των χρηστών του συστήματος.
Ο υπολογισμός αυτός γίνεται με βάση την εξίσωση :

$$\vec{t}^{(k+1)} = (1-a) \times A'^{T} \times t^{(k)} + ap$$

Στην αρχικοποίηση του συστήματος θέτουμε :

$$t_0 = p$$

Υπολογίζουμε την τιμή t^k για αρκετά υψηλό k (συνήθως k=7-10) τέτοιο ώστε το διάνυσμα t να συγκλίνει. Η τελική τιμή του t αποτελεί το καθολικό διάνυσμα εμπιστοσύνης, μια αναφορική διάταξη εμπιστοσύνης μεταξύ των χρηστών.

$$t_\infty = \begin{bmatrix} 0.35 \\ 0.49 \\ 0.16 \end{bmatrix}$$

Πλεονεκτήματα EigenTrust

- Μαθηματικά κομψός αλγόριθμος
- Εύκολα προσαρμόσιμοι υπολογισμοί και δυνατή η επέκταση
- Εμπιστοσύνη δεν αποδυναμώνεται μέσω της μεταβατικότητας

Μειονεκτήματα EigenTrust

- Η κανονικοποίηση οδηγεί σε σχεσιακή ερμηνεία της εμπιστοσύνης
- Δεν υπάρχουν μέσα υπολογισμού της αρνητικής εμπιστοσύνης
- Το καθολικό αποδεκτό διάνυσμα εμπιστοσύνης δεν είναι η καλύτερη λύση για εχθρικά και κακόβουλα δίκτυα

4.3.2 Προσομοίωση ROCQ

Η υλοποίηση του ROCQ είναι βασισμένη στο τετράπτυχο Φήμη, Γνώμη, Αξιοπιστία και Ποιότητα. Έπειτα από κάθε συναλλαγή του κόμβου i (αιτών) με τον κόμβο j (προμηθευτής), ο i στέλνει feedback σχετικό με τον προμηθευτή j. Ανάλογα με το μοντέλο του χρήστη i και την εγκυρότητα του αρχείου που έλαβε, στέλνεται και αποθηκεύεται η τιμή 1 ή 0. Ο αλγόριθμος που υλοποιήσαμε αποτελείται από πέντε (5) δισδιάστατες δομές δεδομένων. Ο σχεδιασμός του στηρίχθηκε, πρώτον, στον υπολογισμό των τιμών εμπιστοσύνης πριν από κάθε συναλλαγή και δεύτερον στην ενημέρωση των δομών δεδομένων του ROCQ έπειτα από κάθε συναλλαγή.

Έπειτα από μία αλληλεπίδραση του χρήστη i με κάποιον χρήστη j είναι αναγκαία η ενημέρωση των δομών δεδομένων. Αυτές είναι με σειρά προτεραιότητας οι : (1)opinion, (2) standard deviation, (3) quality, (4) credibility και (5) reputation. Πρώτος ενημερώνεται ο πίνακας γνωμών opinion. Το στοιχείο O_{ij}^{avg} είναι ο μέσος αριθμός γνωμών που έχει ο χρήστης i για τον χρήστη j. Ουσιαστικά είναι ίσο με τον αριθμό των θετικών feedback που έχει στείλει ο i στον j διαιρούμενο με τον συνολικό αριθμό των αλληλεπιδράσεων του i με προμηθευτή τον j, N_{ij} [8].

$$O_{ij}^{avg} = \begin{bmatrix} 0 & O_{12}^{avg} & O_{13}^{avg} \\ O_{21}^{avg} & 0 & O_{23}^{avg} \\ O_{31}^{avg} & O_{32}^{avg} & 0 \end{bmatrix}$$

Στη συνέχεια πρέπει να ενημερώσουμε τον πίνακα της τυπικής απόκλισης s των γνωμών σύμφωνα με τον τύπο :

$$s_{ij} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_{ij} - O_{ij}^{avg})^2}$$

[8] Δίνουμε παραδείγματα πινάκων για ένα σύστημα 3x3, δηλαδή δίκτυο με 3 χρήστες.

Το στοιχείο s_{ij} αντιστοιχεί στην τυπική απόκλιση των γνωμών που έχει υποβάλλει ο χρήστης i για τον χρήστη j.

$$s_{ij} = \begin{bmatrix} 0 & s_{12} & s_{13} \\ s_{21} & 0 & s_{23} \\ s_{31} & s_{32} & 0 \end{bmatrix}$$

Τρίτος κατά σειρά ενημέρωσης είναι ο δείκτης ποιότητας Q_{ij}. Ο δείκτης ποιότητας δείχνει την βαρύτητα που δίνεται στην γνώμη του i για τον j και υπολογίζεται από την σχέση 2.8 της σελίδας 47.

$$Q_{ij} = \begin{bmatrix} 0 & Q_{12} & Q_{13} \\ Q_{21} & 0 & Q_{23} \\ Q_{31} & Q_{32} & 0 \end{bmatrix}$$

Όπως είδαμε στην ενότητα 2.2.2, όπου αναλύσαμε το μοντέλο ROCQ, για τον υπολογισμό του δείκτη ποιότητας κάνουμε χρήση της *Ατελούς Συνάρτησης Β*. Να τονίσουμε σε αυτό το σημείο πως στις παραπάνω δομές δεδομένων στην συναλλαγή N_{ij} ενημερώνουμε μόνο τα αντίστοιχα στοιχεία των πινάκων.

Ακολουθούν οι ενημερώσεις των πινάκων αξιοπιστίας και φήμης. Ο δείκτης αξιοπιστίας δείχνει την εμπιστοσύνη που έχουν όλοι οι χρήστες για το feedback του χρήστη i που υποβάλλει για τον χρήστη j.

$$C_{mi} = \begin{bmatrix} 0 & C_{12} & C_{13} \\ C_{21} & 0 & C_{23} \\ C_{31} & C_{32} & 0 \end{bmatrix}$$

Από την άλλη πλευρά ο δείκτης φήμης R_{mj} δείχνει την εμπιστοσύνη που έχει ο χρήστης m για τον χρήστη j.

$$R_{mj} = \begin{bmatrix} 0 & R_{12} & R_{13} \\ R_{21} & 0 & R_{23} \\ R_{31} & R_{32} & 0 \end{bmatrix}$$

Έπειτα από κάθε συναλλαγή, στον πίνακα αξιοπιστίας ενημερώνεται ολόκληρη η στήλη του χρήστη i. Δηλαδή, κάθε χρήστης m μεταβάλλει την αξιοπιστία που έχει στο πρόσωπο του i. Αντίστοιχα, στον πίνακα φήμης ενημερώνεται ολόκληρη η στήλη του χρήστη j. Δηλαδή, κάθε χρήστης μεταβάλλει την εμπιστοσύνη που έχει στο πρόσωπο του j. Ο τρόπος με τον οποίον μεταβάλλονται οι πίνακες αυτοί παρουσιάστηκε στην ενότητα 2.2.2.

Πριν από κάθε συναλλαγή ο χρήστης m που ζητάει ένα συγκεκριμένο αρχείο, χρειάζεται να γνωρίζει ένα μέτρο εμπιστοσύνης για τους υπόλοιπους χρήστες, ώστε να κάνει την καλύτερη για εκείνον επιλογή. Αυτό επιτυγχάνεται με τον τύπο :

$$vectorA_j = \frac{\sum_y R_{yj} \cdot C_{my} \cdot Q_{yj}}{\sum_y C_{my} \cdot Q_{yj}}$$

Ο χρήστης m εξάγει και αποθηκεύει στον πίνακα γραμμή vectorA τις τιμές εμπιστοσύνης για όλους τους υπόλοιπους χρήστες. Τέλος, να επισημάνουμε πως κατά την αρχικοποίηση των δομών δεδομένων, θέτουμε τα στοιχεία του πίνακα ποιότητας Q_{ij} $(i \neq j)$ ίσα με 1, ενώ τα στοιχεία του πίνακα αξιοπιστίας C_{mi} ίσα με 0.5. Ο κώδικας που υλοποιήσαμε σε γλώσσα προγραμματισμού C για τον αλγόριθμο ROCQ δίνεται στο Παράρτημα Α..

4.3.3 Προσομοίωση Bayesian

Η ιδέα υλοποίησης του Bayesian αλγορίθμου φήμης είναι αυτή που δόθηκε στην Ενότητα 2.2.3. Όταν κάποιος χρήστης i ζητάει ένα μέτρο αξιολόγησης της εμπιστοσύνης των διάφορων προμηθευτών, παίρνει συστάσεις από όλους τους χρήστες και τις συνυπολογίζει σύμφωνα με τη σχέση :

$$r_{ij} = w_t \times \frac{\sum_{l=1}^{k} tr_{il} \cdot t_{lj}}{\sum_{l=1}^{k} tr_{il}} + w_s \times \frac{\sum_{z=1}^{g} t_{zj}}{g}$$

όπου $w_t + w_s = 1$, με w_t και w_s τα βάρη που υποδηλώνουν το πόσο πολύ ο χρήστης λαμβάνει υπόψη τις συστάσεις από τις αξιόπιστες αναφορές και από τις άγνωστες αναφορές αντίστοιχα. Το μέτρο r_{ij} είναι η συνολική τιμή σύστασης για τον προμηθευτή αρχείων j που παίρνει ο χρήστης i, τα k και g είναι οι αριθμοί των αξιόπιστων και άγνωστων αναφορών αντίστοιχα, tr_{il} είναι η εμπιστοσύνη που έχει ο χρήστης i στην αξιόπιστη αναφορά l, t_{lj} είναι η εμπιστοσύνη που έχει η αξιόπιστη αναφορά l στον προμηθευτή αρχείων j, t_{zj} είναι η εμπιστοσύνη που έχει η άγνωστη αναφορά z στον προμηθευτή αρχείων j. Δεδομένης μιας τιμής κατωφλίου θ, εάν η συνολική τιμή σύστασης r_{ij} είναι μεγαλύτερη από το θ, ο πράκτορας θα αλληλεπιδράσει με τον προμηθευτή αρχείων, διαφορετικά, όχι. Στο σύστημα που υλοποιήσαμε θέσαμε $\theta = 0.5$.

Εάν ο χρήστης αλληλεπιδράσει με τον προμηθευτή αρχείων, όχι μόνο θα ενημερώσει την τιμή εμπιστοσύνης του για τον προμηθευτή αρχείων, δηλαδή το αντίστοιχο Bayesian δίκτυό του, αλλά επιπλέον θα ενημερώσει τις τιμές εμπιστοσύνης του για τους πράκτορες που του έδωσαν τις συστάσεις, σύμφωνα με τον ακόλουθο τύπο :

$$tr_{ij}^{n} = a \cdot tr_{ij}^{o} + (1-a) \cdot e_a$$

όπου η tr_{ij}^{n} είναι η νέα τιμή εμπιστοσύνης που ο πράκτορας i έχει στην αναφορά j μετά από την ενημέρωση, η tr_{ij}^{o} είναι η προηγούμενη τιμή εμπιστοσύνης, ο a είναι ο ρυθμός εκμάθησης και είναι ένας πραγματικός αριθμός στο διάστημα [0,1] (εδώ παίρνουμε $a = 0.3$) και η e_a είναι η νέα τιμή στοιχείων (evidence) και μπορεί να είναι -1 ή 1. Εάν η τιμή της σύστασης είναι μεγαλύτερη από το θ και η κατόπιν αλληλεπίδραση με τον προμηθευτή αρχείων είναι, τότε τίθεται e_a ίσο με 1. Σε αντίθετη περίπτωση, τίθεται e_a ίσο με -1.

Ο κώδικας που υλοποιήσαμε για τον αλγόριθμο φήμης Bayesian βρίσκεται σε γλώσσα προγραμματισμού C, στο Παράρτημα Α.

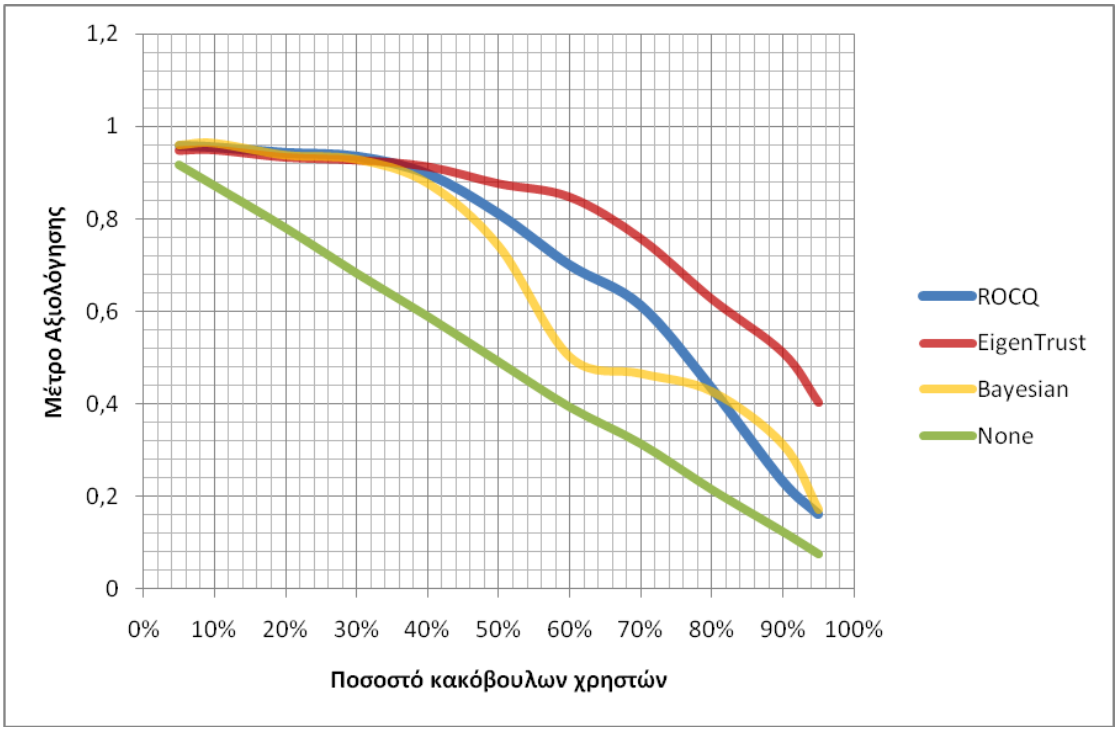
4.4 Αξιολόγηση και σύγκριση πειραματικών προσομοιώσεων

Στην ενότητα αυτή θα εκτιμήσουμε τους αλγόριθμους φήμης EigenTrust, ROCQ και Bayesian σε ένα p2p δίκτυο, θα τους συγκρίνουμε μεταξύ τους, αλλά και ταυτόχρονα θα δούμε τις διαφορές που υφίστανται εάν δεν υπάρχει κάποιος αλγόριθμος φήμης. Για την ορθότερη αξιολόγηση των αλγορίθμων φήμης θα ακολουθήσουμε διαφορετικά σενάρια προσομοιώσεων κάθε φορά[9].

Σενάριο Α

Στο Σενάριο Α έχουμε ένα δίκτυο 200 κόμβων με 5 pre-trusted χρήστες, όπου λαμβάνουν χώρο 10.000 συναλλαγές μεταξύ των χρηστών (~ 50 trans ανά χρήστη). Κάναμε προσομοιώσεις για κακόβουλους χρήστες που αποτελούν το 5 - 95% στο σύνολο του δικτύου, με βήμα 10% και 10 επαναλήψεις ανά προσομοίωση. Σε κάθε αλγόριθμο ξεχωριστά, χρησιμοποιήσαμε το ίδιο αρχείο ίχνους, ώστε να επικρατούν ίδιες συνθήκες στο σύστημα. Τα αποτελέσματα απεικονίζονται στην γραφική παράσταση 4.1.

[9] Όλες οι προσομοιώσεις που πραγματοποιήθηκαν υπάρχουν στην διεύθυνση : http://pithos.grnet.gr/pithos/rest/el04120@ntua.gr/files/%CE%9A%CE%BF%CE%B9%CE%BD%CF%8C%CF%87%CF%81%CE%B7%CF%83%CF%84%CE%BF%CF%82/Pr.rar



Γραφική Παράσταση 4.1: Σενάριο Α

Παρατηρούμε πως όταν δεν υπάρχει κάποιος αλγόριθμος φήμης η πιθανότητα κάποιος καλός χρήστης να συμμετέχει σε έγκυρη συναλλαγή μειώνεται γραμμικά (πράσινη καμπύλη). Για μικρό ποσοστό κακόβουλων χρηστών (έως 10%) τα αποτελέσματα είναι ικανοποιητικά, αλλά στη συνέχεια ακολουθεί φθίνουσα πορεία με σταθερό ρυθμό.

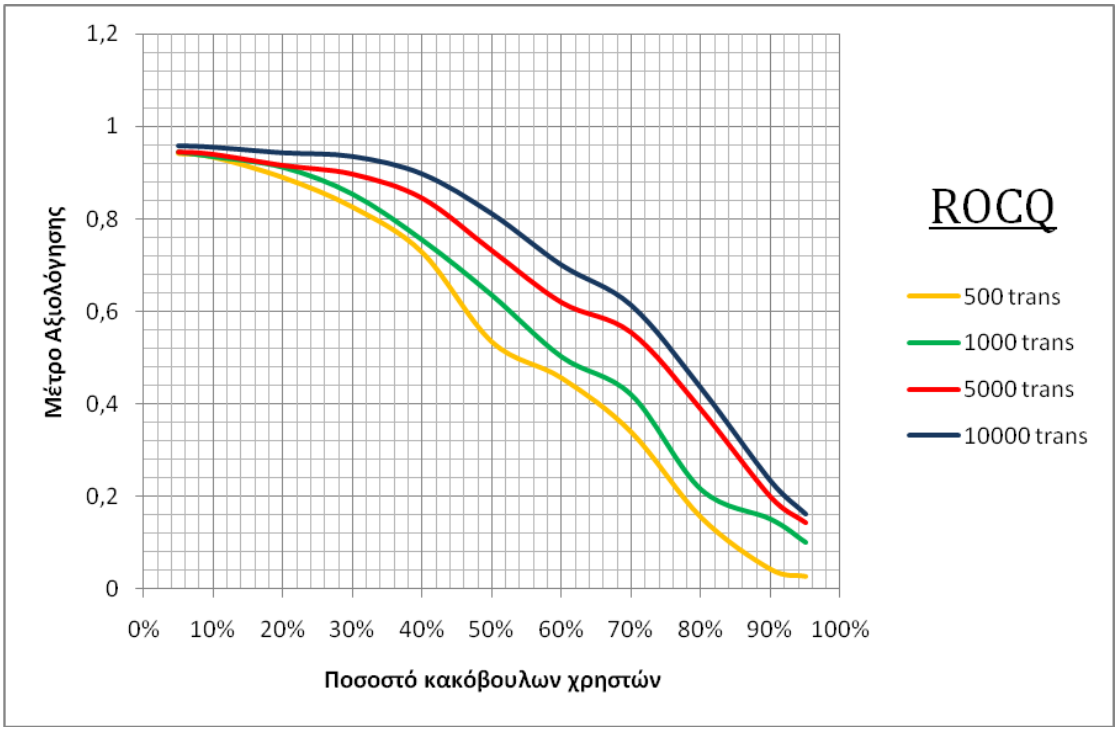
Για τον αλγόριθμο EigenTrust (κόκκινη καμπύλη) συμπεραίνουμε πως έχει καλή απόδοση με πιθανότητα έγκυρης συναλλαγής μεγαλύτερη από 0.8, μέχρι 60% κακόβουλων χρηστών (κόκκινη καμπύλη). Έπειτα για περισσότερους κακόβουλους χρήστες το μέτρο αξιολόγησης μειώνεται έως και λίγο πιο πάνω από το 0.4.

Για τον αλγόριθμο ROCQ (μπλε καμπύλη) παρατηρούμε μέχρι 50% κακόβουλων χρηστών πολύ καλή απόδοση, παρόμοια με του EigenTrust. Στη συνέχεια, όπως περιμέναμε, η απόδοση του αλγορίθμου μειώνεται με εκθετικό ρυθμό και πέφτει λίγο πιο κάτω από 0.2.

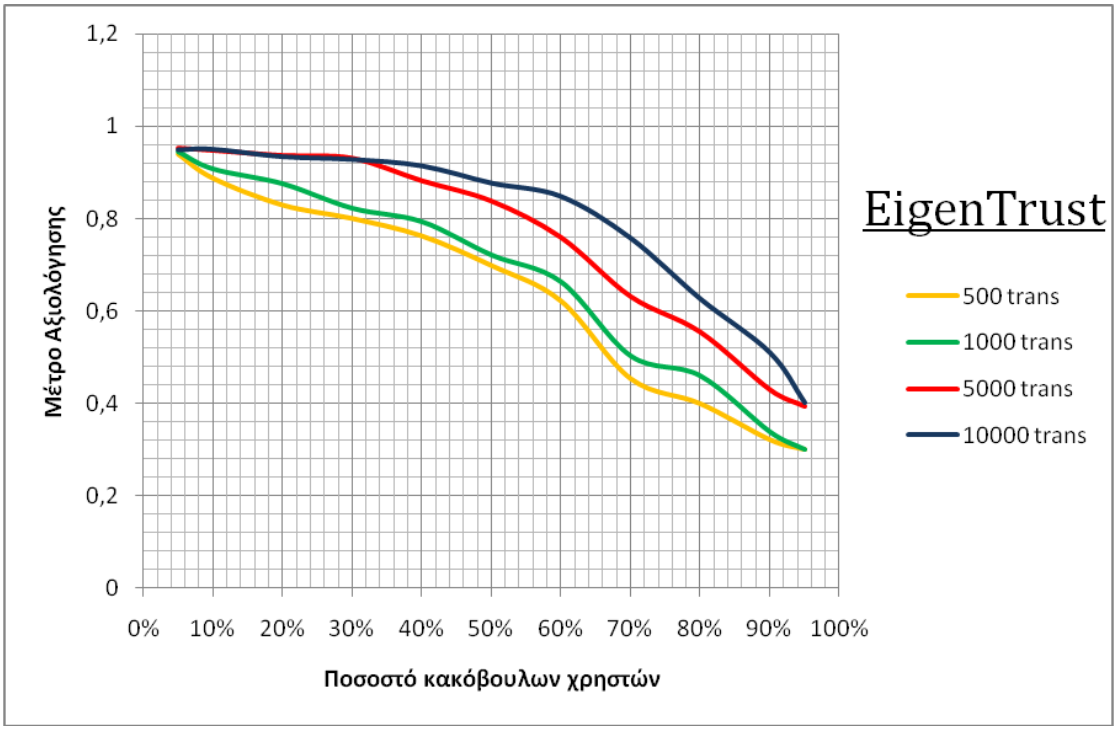
Τέλος, ο Bayesian αλγόριθμος έχει καλύτερη απόδοση σε σχέση με τους άλλους δύο για μικρά ποσοστά κακόβουλων χρηστών και έπειτα μειώνεται απότομα όμοια με τον ROCQ. Η απότομη μείωση αυτή παρατηρείται μεταξύ 40% και 60% κακόβουλων χρηστών και οφείλεται στο γεγονός πως όταν οι καλοί χρήστες αποτελούν μειοψηφία σε ένα δίκτυο, ο αλγόριθμος σταματάει να είναι αποδοτικός.

Σενάριο Β

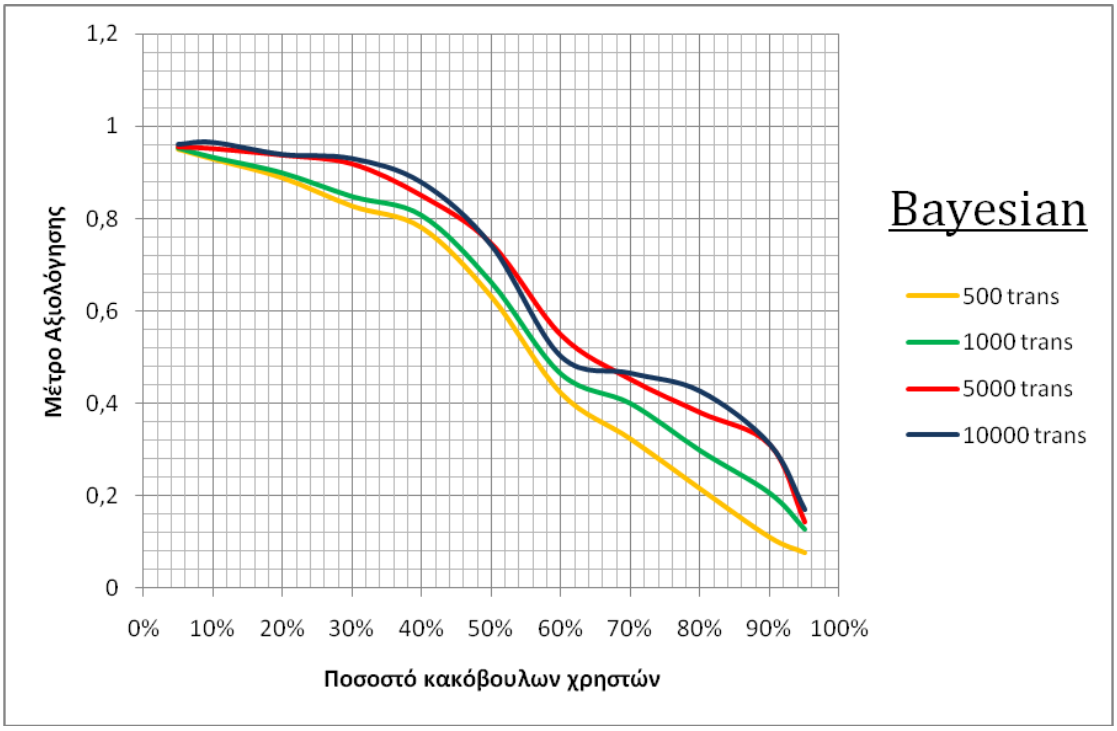
Στο Σενάριο Β έχουμε ένα δίκτυο 200 κόμβων με 5 pre-trusted χρήστες, όπου εκτελούμε τους αλγορίθμους EigenTrust, ROCQ και Bayesian για 500, 1.000, 5.000 και 10.000 συναλλαγές. Σκοπός είναι να δούμε την σύγκλιση των αλγορίθμων φήμης σε αναλογία με τον αριθμό των συναλλαγών. Κάναμε προσομοιώσεις για κακόβουλους χρήστες που αποτελούν το 5 - 95% στο σύνολο του δικτύου, με βήμα 10% και 10 επαναλήψεις ανά προσομοίωση. Τα αποτελέσματα από το τρέξιμο των προσομοιώσεων του σεναρίου Β παρουσιάζονται στις ακόλουθες γραφικές παραστάσεις.



Γραφική Παράσταση 4.2: Σενάριο Β ROCQ



Γραφική Παράσταση 4.3: Σενάριο Β EigenTrust

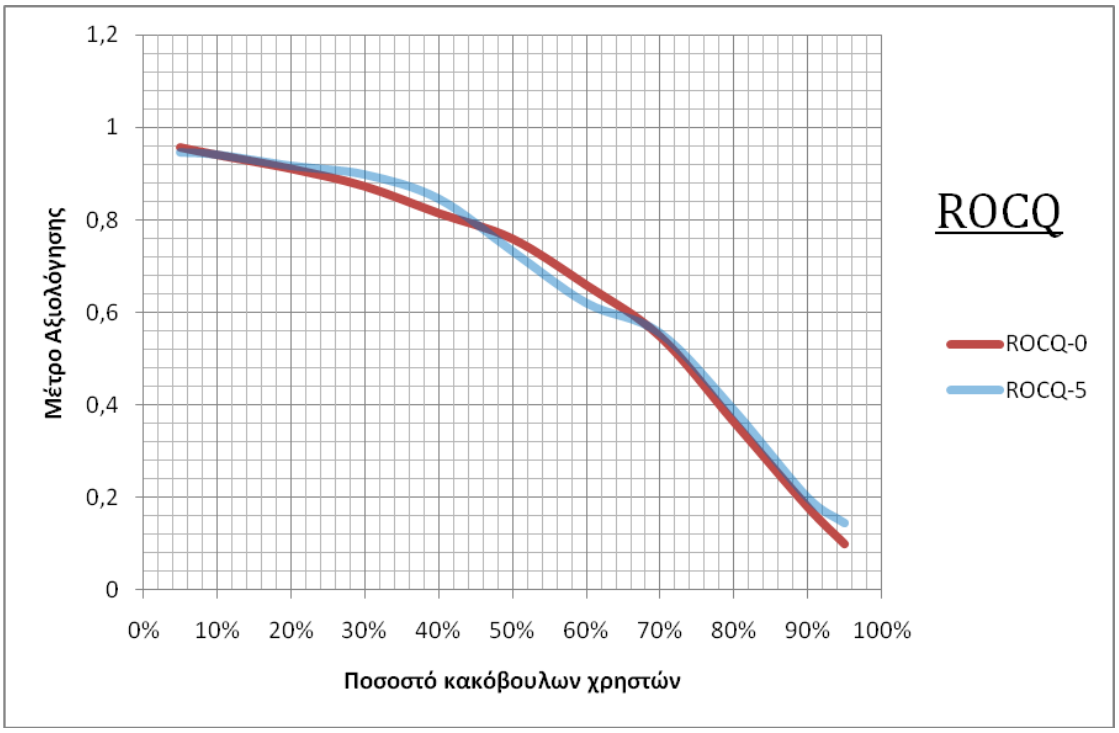


Γραφική Παράσταση 4.4: Σενάριο Β Bayesian

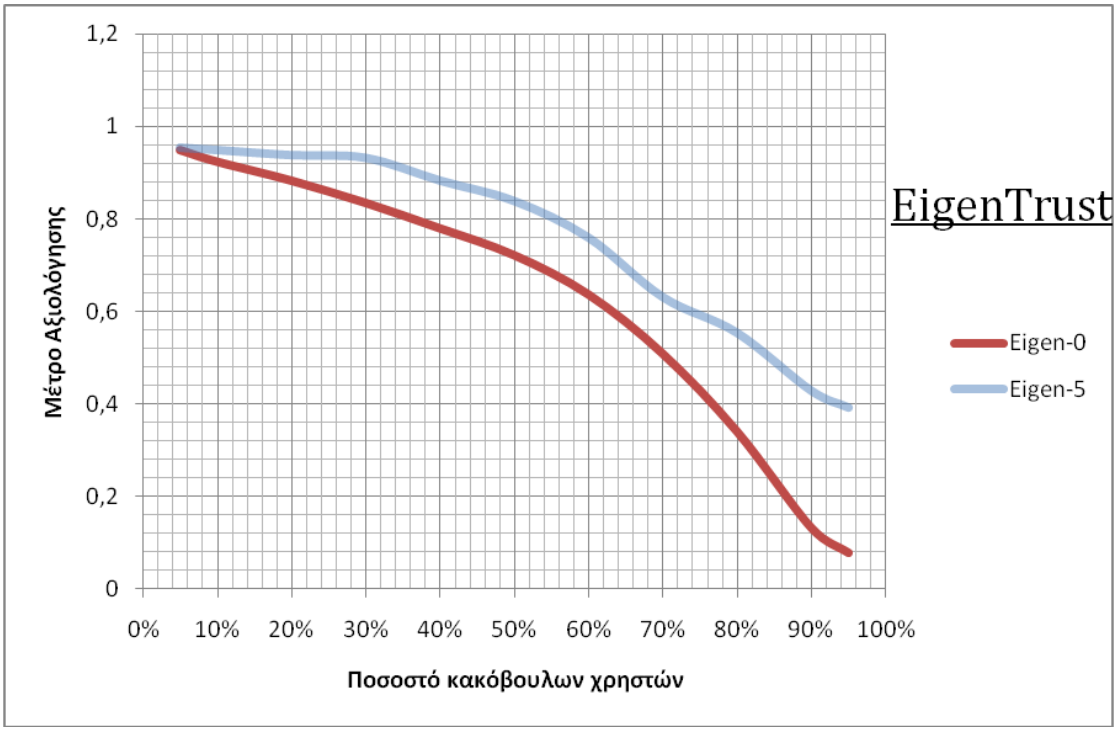
Παρατηρούμε πως οι τιμές εμπιστοσύνης και στους τρεις αλγόριθμους συγκλίνουν πολύ γρήγορα, γεγονός που οφείλεται στο ότι οι χρήστες στον P2P Simulator συμπεριφέρονται με συνέπεια και αλληλεπιδρούν συνεχώς. Χρησιμοποιούμε αυτό το δεδομένο, για να δημιουργήσουμε γρήγορες στρατηγικές που ξεπερνούν προβλήματα πολυπλοκότητας. Για αριθμό συναλλαγών 5.000 και μεγαλύτερο παρατηρούμε ότι τα αποτελέσματα συγκλίνουν και το περιθώριο λάθους είναι < 1% σε όλα τα σημεία δεδομένων.

Σενάριο Γ

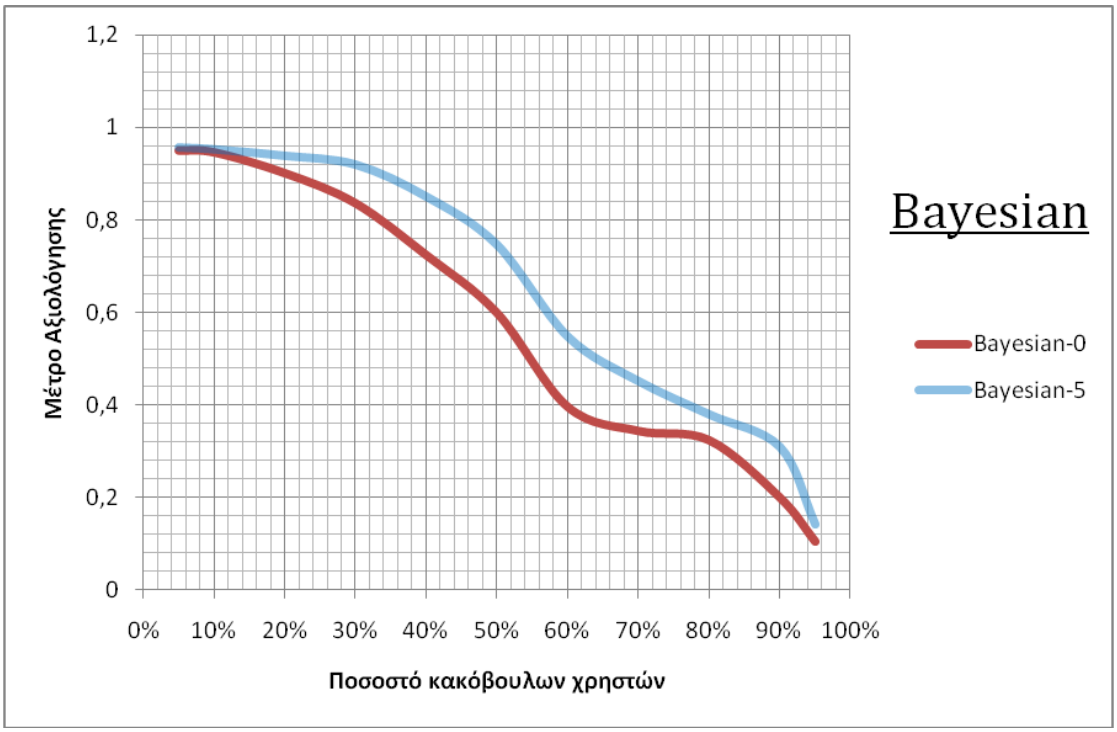
Στο Σενάριο Γ έχουμε ένα δίκτυο 200 κόμβων, όπου λαμβάνουν χώρο 5.000 συναλλαγές μεταξύ των χρηστών (~ 25 trans ανά χρήστη). Κάναμε προσομοιώσεις για κακόβουλους χρήστες που αποτελούν το 5 - 95% στο σύνολο του δικτύου, με βήμα 10% και 10 επαναλήψεις ανά προσομοίωση. Σκοπός του σεναρίου Γ είναι να συγκρίνουμε τους τρεις αλγόριθμους σε δύο καταστάσεις του συστήματος. Στην πρώτη δεν υπάρχουν καθόλου pre-trusted χρήστες, ενώ στην δεύτερη υπάρχουν 5 pre-trusted χρήστες.



Γραφική Παράσταση 4.5: Σενάριο Γ ROCQ



Γραφική Παράσταση 4.6: Σενάριο Γ EigenTrust



Γραφική Παράσταση 4.7: Σενάριο Γ Bayesian

Οι pre-trusted χρήστες αποτελούν ένα υποσύνολο των καλών χρηστών. Η συνεισφορά τους σε ένα δίκτυο είναι πολύτιμη, ιδιαίτερα για τους νέους χρήστες που δεν έχουν εμπειρία με άλλους χρήστες. Στην περίπτωση αυτή μπορούν για τις συναλλαγές τους να εμπιστευτούν τους pre-trusted χρήστες.

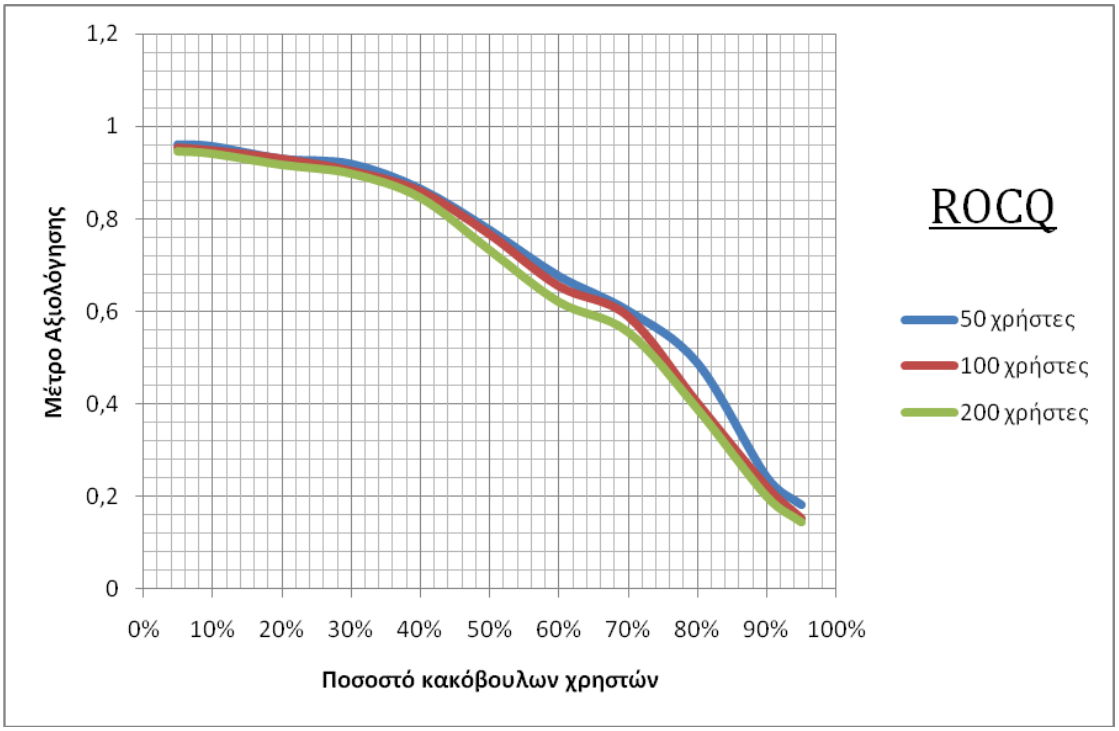
Τα αποτελέσματα των προσομοιώσεων διαφέρουν για τους τρεις αλγόριθμους φήμης. Στον EigenTrust η προσθήκη pre-trusted κόμβων παράγει μια απίστευτη βελτίωση. Η ύπαρξη της έννοιας pre-trust κάνει τον καθολικό συνυπολογισμό των τιμών εμπιστοσύνης λιγότερο αφελή και παρέχει επιπρόσθετες βελτιώσεις που αναλύσαμε στην ενότητα 2.2.1. Για παράδειγμα, καινούριοι χρήστες που δεν έχουν εικόνα του δικτύου τοποθετούν όλη τους την εμπιστοσύνη στους pre-trusted χρήστες.

Από την άλλη πλευρά, στον αλγόριθμο φήμης ROCQ, η ύπαρξη pre-trusted κόμβων στο δίκτυο δεν προσφέρει κάποιο πλεονέκτημα. Η απόδοση του αλγορίθμου είναι το ίδιο καλή και στις δύο περιπτώσεις και συνεπώς ανεξάρτητη των pre-trusted χρηστών. Το γεγονός αυτό ήταν αναμενόμενο και από την θεωρητική προσέγγιση του αλγορίθμου που παρουσιάσαμε στην ενότητα 2.2.2, καθώς ο ROCQ δεν στηρίζεται στην ύπαρξη της έννοιας των pre-trusted χρηστών στο δίκτυο.

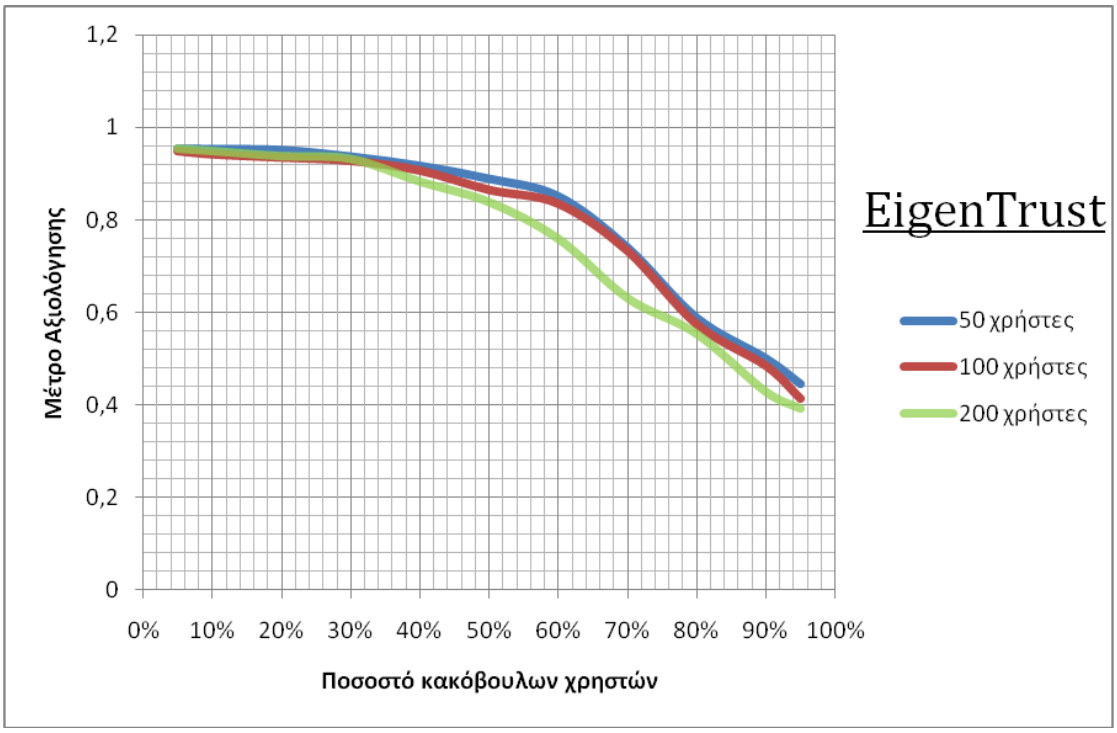
Για τον Bayesian παρατηρούμε πως η προσθήκη pre-trusted χρηστών βελτιώνει την απόδοση του αλγορίθμου, αλλά σε μικρότερα ποσοστά σε σχέση με τον EigenTrust.

Σενάριο Δ

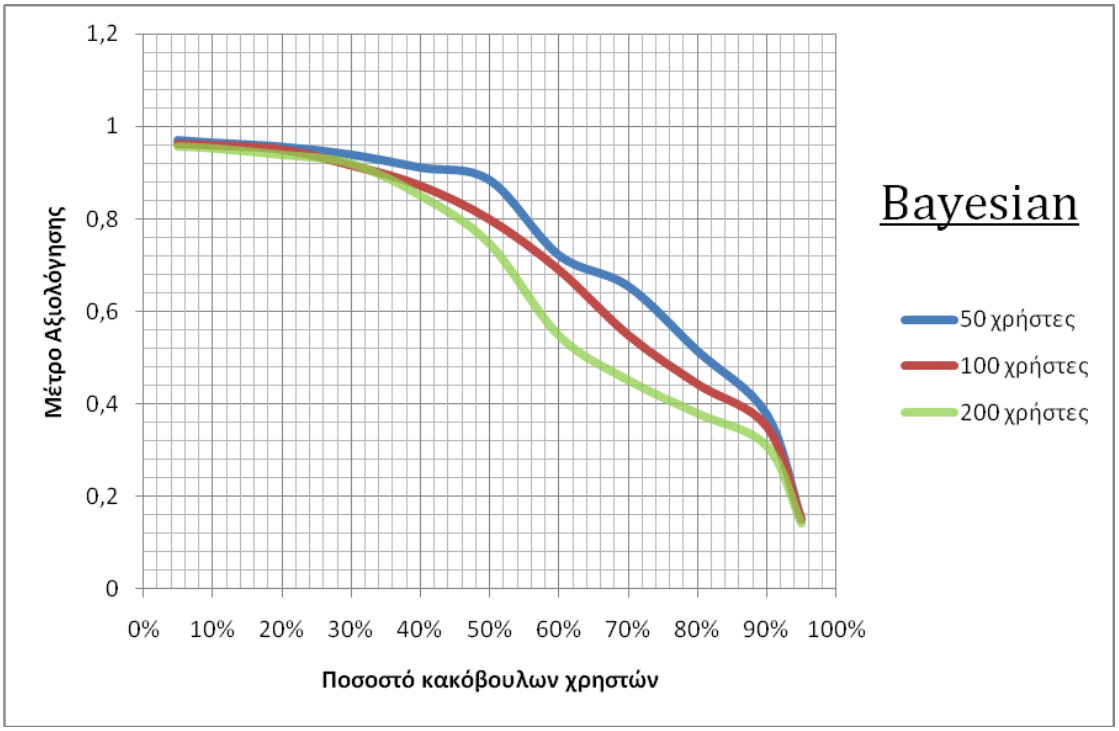
Στο Σενάριο Δ έχουμε ένα δίκτυο όπου λαμβάνουν χώρο 5.000 συναλλαγές μεταξύ των χρηστών για αριθμό χρηστών ίσο με 50, 100 και 200. Κάναμε προσομοιώσεις για κακόβουλους χρήστες που αποτελούν το 5 - 95% στο σύνολο του δικτύου, με βήμα 10% και 10 επαναλήψεις ανά προσομοίωση. Σκοπός του σεναρίου Δ είναι να συγκρίνουμε τους τρεις αλγόριθμους, ανάλογα με τον αριθμό των χρηστών του συστήματος και να δούμε πόσο επηρεάζει την σύγκλιση των αλγορίθμων η πυκνότητα σε χρήστες του δικτύου.



Γραφική Παράσταση 4.8: Σενάριο Δ ROCQ



Γραφική Παράσταση 4.9: Σενάριο Δ EigenTrust



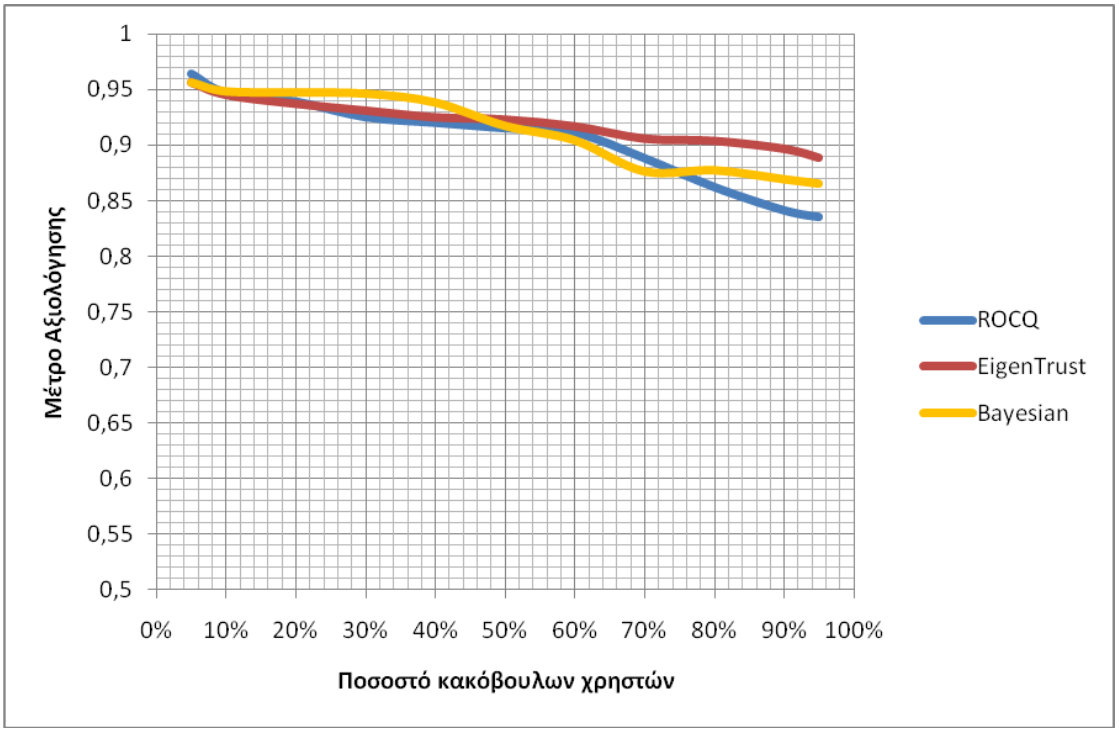
Γραφική Παράσταση 4.10: Σενάριο Δ Bayesian

Εύκολα γίνεται αντιληπτό, παρατηρώντας τις παραπάνω γραφικές παραστάσεις, πως για τους αλγόριθμους EigenTrust και ROCQ, η πυκνότητα του δικτύου δεν επηρεάζει σε μεγάλο βαθμό την απόδοση τους. Ειδικότερα στον ROCQ τα αποτελέσματα συγκλίνουν με μεγάλη ταχύτητα. Το συμπέρασμα αυτό είναι αρκετά ενθαρρυντικό για την σκέψη επέκτασης των αλγορίθμων αυτών σε δίκτυα πραγματικών διαστάσεων.
Από την άλλη πλευρά ο Bayesian αλγόριθμος, όπως ακριβώς περιμέναμε, είναι άριστα εφαρμόσιμος σε μικρά δίκτυα, όπου η πιθανότητα αγοραστή και πωλητή να έχουν συνεχόμενες αλληλεπιδράσεις είναι μεγάλη (φαινόμενο μικρού κόσμου). Αντίθετα, όσο μεγαλώνει το δίκτυο, όπου η πιθανότητα ένας πωλητής να συναντήσει τον ίδιο αγοραστή είναι μικρή, το Bayesian μοντέλο δεν δίνει το ίδιο ικανοποιητικά αποτελέσματα.

Σενάριο Ε

Στο Σενάριο Ε έχουμε ένα δίκτυο 100 χρηστών, όπου λαμβάνουν χώρο 5.000 συναλλαγές μεταξύ των χρηστών. Κάναμε προσομοιώσεις για μεταμφιεσμένους κακόβουλους χρήστες (disguised malicious) που αποτελούν το 5 - 95% στο σύνολο του δικτύου, με βήμα 10% και 10 επαναλήψεις ανά προσομοίωση. Σκοπός του

σεναρίου Ε είναι να συγκρίνουμε τους αλγόριθμους στην αντιμετώπιση μεταμφιεσμένων κακόβουλων χρηστών, δηλαδή χρηστών που λειτουργούν άλλοτε ως κακόβουλοι και άλλοτε ως καλοί χρήστες με σκοπό να ξεγελάσουν το σύστημα.



Γραφική Παράσταση 4.11: Σενάριο Ε

Τα αποτελέσματα που φαίνονται στην γραφική παράσταση 4.8 είναι άκρως εντυπωσιακά. Και οι τρεις αλγόριθμοι αντιδρούν εξαιρετικά απέναντι σε μεταμφιεσμένους κακόβουλους χρήστες. Για τον EigenTrust παρατηρούμε πως σπάνια πέφτει σε απόδοση το 0.9. Ο ROCQ αντιδράει το ίδιο καλά μέχρι 60% κακόβουλων χρηστών. Έπειτα, πέφτει κάτω από το 0.9 έως και απόδοση 0.84. Παρόλα αυτά υπάρχει τρομακτική διαφορά σε σύγκριση με τους αμιγής κακόβουλους χρήστες, όπου το μέτρο αξιολόγησης μειώνεται σε 0.1-0.2 για ποσοστό κακόβουλων χρηστών ίσο με 95%.

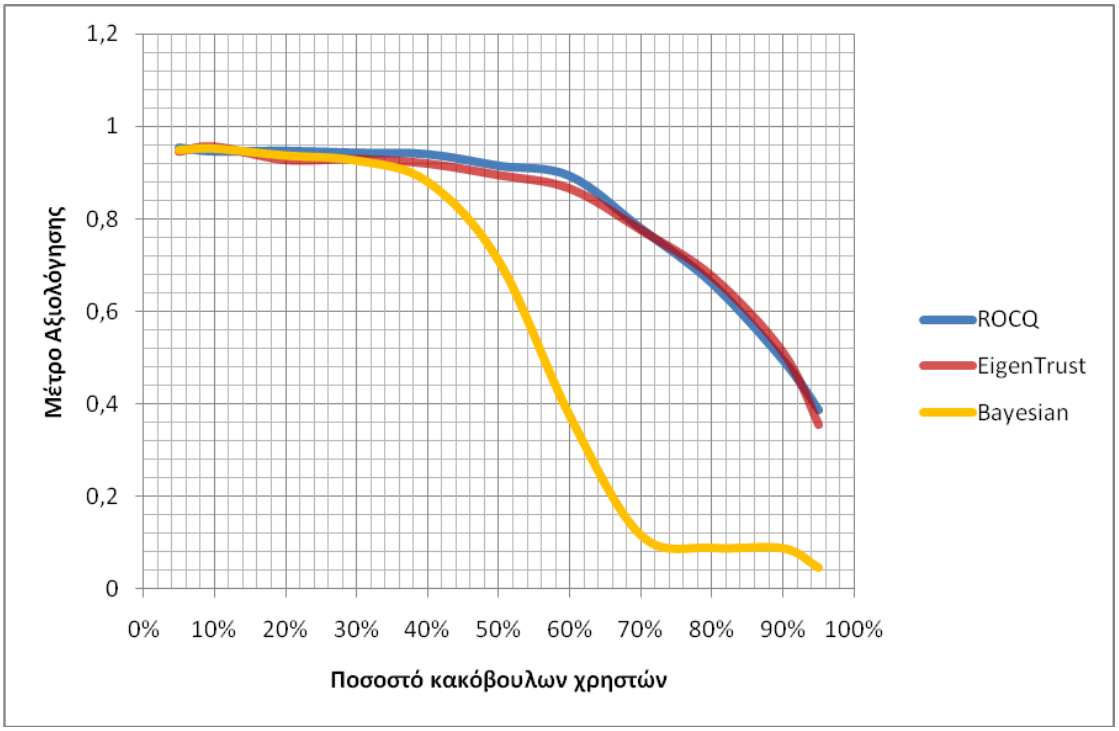
Η καμπύλη του Bayesian έχει καλύτερη απόδοση από τους άλλους αλγορίθμους μέχρι 40% μεταμφιεσμένων κακόβουλων. Και πάλι, όταν οι κακόβουλοι αποτελούν την πλειοψηφία, η απόδοση του Bayesian μειώνεται απότομα.

Οι τρεις αλγόριθμοι ακόμα και όταν υπάρχει πάρα πολύ μεγάλη πυκνότητα κακόβουλων χρηστών, καταφέρνουν να εντοπίζουν σε υψηλό βαθμό τους μεταμφιεσμένους κακόβουλους. Το γεγονός αυτό οφείλεται στο ότι η στρατηγική που

ακολουθούν οι κακόβουλοι χρήστες είναι επιζήμια για τους ίδιους. Όταν οι κακόβουλοι αυτοί χρήστες, εν μέρει παρέχουν γνήσια αρχεία, τους δίνονται περισσότερες θετικές τιμές τοπικής εμπιστοσύνης, με αποτέλεσμα να επιλέγονται ως προμηθευτές περισσότερες φορές. Παρόλα αυτά, με το να παρέχουν, συχνά, γνήσια αρχεία και να κρατούν στην βιβλιοθήκη τους τέτοια αρχεία, κοστίζει στις κακόβουλες βλέψεις τους.

Σενάριο ΣΤ

Στο Σενάριο ΣΤ έχουμε ένα δίκτυο 100 χρηστών, όπου λαμβάνουν χώρο 5.000 συναλλαγές μεταξύ των χρηστών. Κάναμε προσομοιώσεις για σιβυλλικούς κακόβουλους χρήστες (Sybil malicious) που αποτελούν το 5 - 95% στο σύνολο του δικτύου, με βήμα 10% και 10 επαναλήψεις ανά προσομοίωση. Σκοπός του σεναρίου ΣΤ είναι να μελετήσουμε την συμπεριφορά των αλγορίθμων φήμης σε σιβυλλικές επιθέσεις.



Γραφική Παράσταση 4.12: Σενάριο ΣΤ

Τα μοντέλα EigenTrust και ROCQ αντιδρούν το ίδιο καλά σε σιβυλλικούς κακόβουλους χρήστες. Οι καμπύλες (μπλε και κόκκινη) όπως παρατηρούμε είναι

παρόμοιες. Μάλιστα, συγκρίνοντας τις γραφικές παραστάσεις 4.1 και 4.12, παρατηρούμε πως η απόδοση του EigenTrust είναι ακριβώς ίδια απέναντι σε αμιγής και σιβυλλικούς κακόβουλους.

Ο αλγόριθμος φήμης Bayesian παρότι μέχρι 40% σιβυλλικών χρηστών αντιδράει το ίδιο καλά με τους άλλους αλγορίθμους, στη συνέχεια όταν οι καλοί χρήστες αποτελούν την μειοψηφία, η απόδοση του μειώνεται ακαριία. Το γεγονός αυτός μας δείχνει πως ο Bayesian δεν έχει κάποιον μηχανισμό, ώστε να αμύνεται επιτυχώς απέναντι στις σιβυλλικές επιθέσεις.

ΠΑΡΑΡΤΗΜΑ Α

Πηγαίος Κώδικας

Στο παράρτημα Α παρουσιάζουμε τους πηγαίους κώδικες σε γλώσσα προγραμματισμού C των αλγορίθμων εμπιστοσύνης που υλοποιήθηκαν και προσομοιώθηκαν στον P2P Simulator. Αρχικά δίνουμε ένα παράδειγμα παραγωγής του αρχείου trace και τρέχουμε το αρχείο αυτό μαζί με τον αλγόριθμο εμπιστοσύνης στον προσομοιωτή από την γραμμή εντολών.

Πίνακας Α.1 : Εντολές trace generation

> -users	(int) # of users/nodes/peers in network
> -files	(int) # of distinct files in network
> -trans	(int) # of transactions to simulate
> -zipf:	(float) Zipf constant controlling file popularity
> -output	(string) Filename for trace output (*.trace)
>	
> -usr:pre_trusted	(int) # of 'pre-trusted' users, a subset of 'good'
> -usr:purely	(int) # of 'purely malicious' users
> -usr:feedback	(int) # of 'feedback skewing' users
> -usr:provider	(int) # of 'malicious provider' users
> -usr:disguise	(int) # of 'disguised malicious' users
> -usr:sybil	(int) # of 'Sybil attack' users
>	
> -band:max_conn	(int) # of max upload/download connections per user
> -band:period	(int) # of time units each upload/download requires
>	
> -mode:smartgen	(bool) Use intelligent transaction generation?
> -mode:warmup	(int) # of warm-up transactions before statistic tabulation

Πίνακας Α.2 : Εντολές trust simulation

Το πρόγραμμα δέχεται τρία ορίσματα. Το αρχείο trace εισόδου, τον αλγόριθμο διαχείρισης εμπιστοσύνης και την κακόβουλη στρατηγική. Και τα τρία ορίσματα είναι υποχρεωτικά.

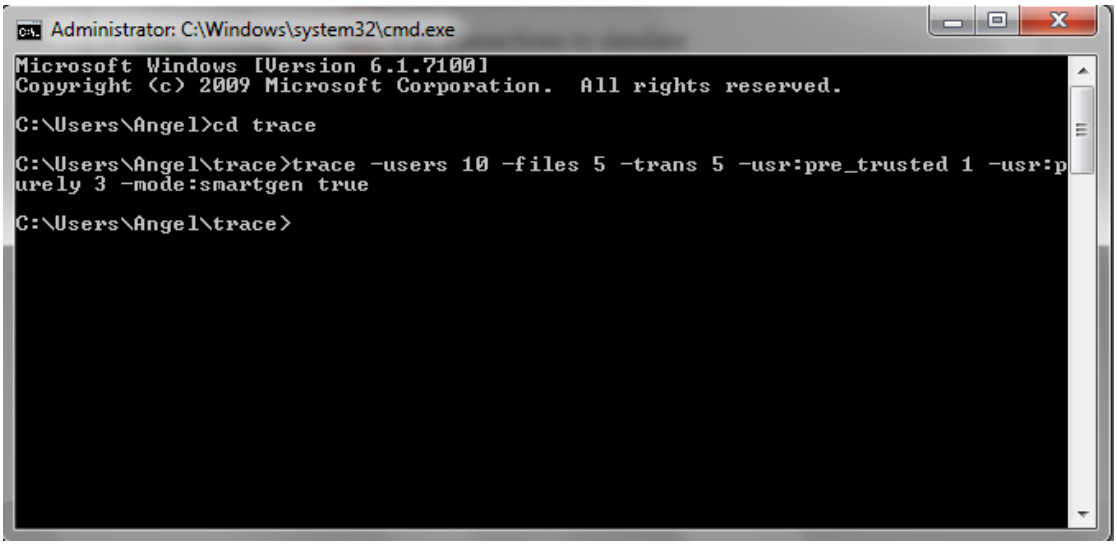
> -input	(string) Input trace file per 'gen_trace' creation (*.trace)
> -tm	(string) Descriptor of trust management algorithm to simulate
> -strategy	(string) Descriptor of malicious strategy to simulate

Το όρισμα -tm είναι κάποιος αλγόριθμος που ήδη έχει υλοποιηθεί στο σύστημα.

> eigen	The EigenTrust algorithm of Hector Garcia-Molina, et al.
> rocq	The ROCQ algorithm of R.Battiti, et al.
> bayesian	A bayesian algorithm of Yao Wang & Julita Vassileva
> none	Absence of trust management. Essentially random source selection.

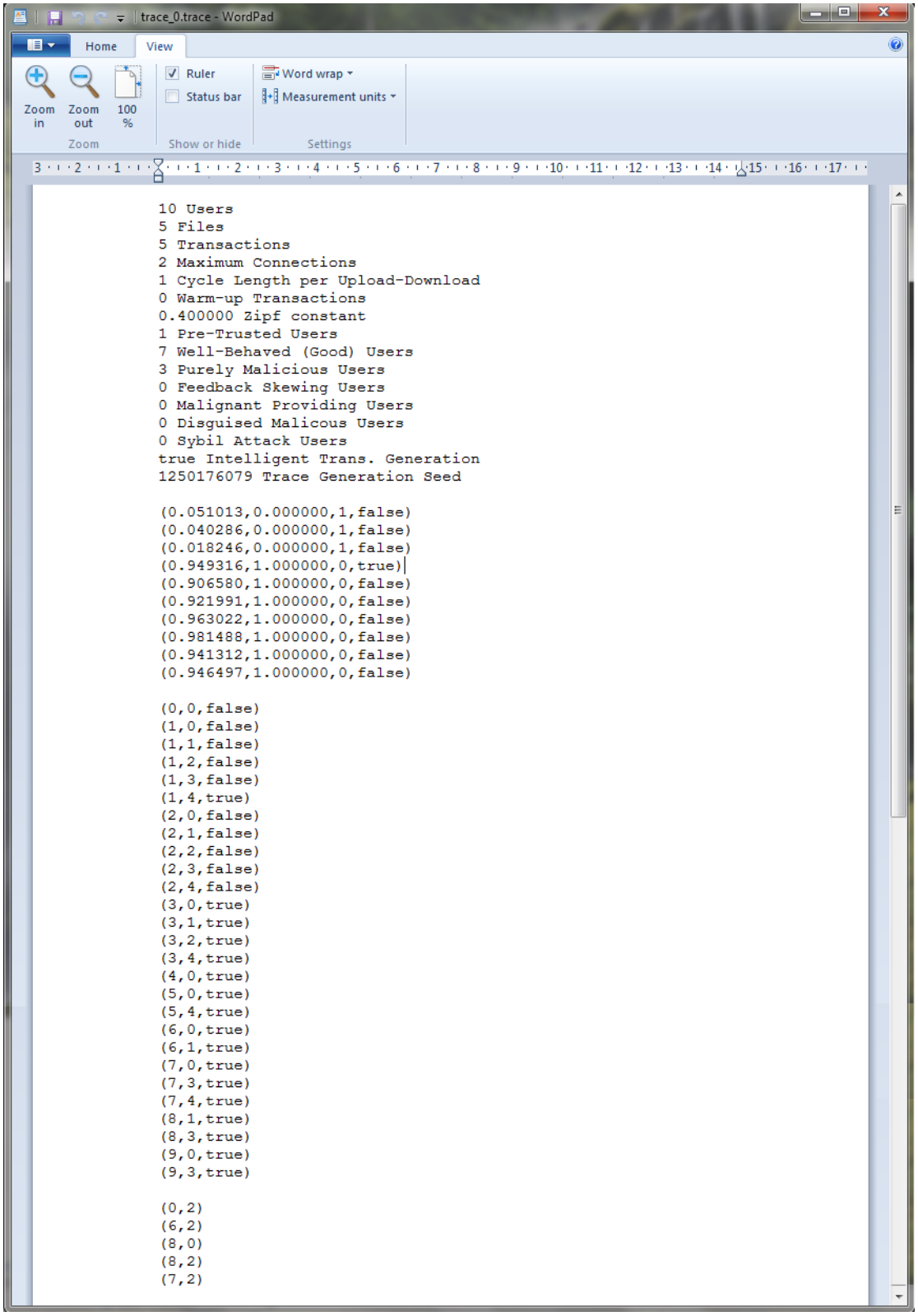
Τα πιθανά ορίσματα του -strategy είναι προγραμματισμένα εκ των προτέρων

> naive	Global interaction data is used exclusively
> isolated	Local honest interaction history overwrites global one
> collective	All malicious peers share honest information with each other



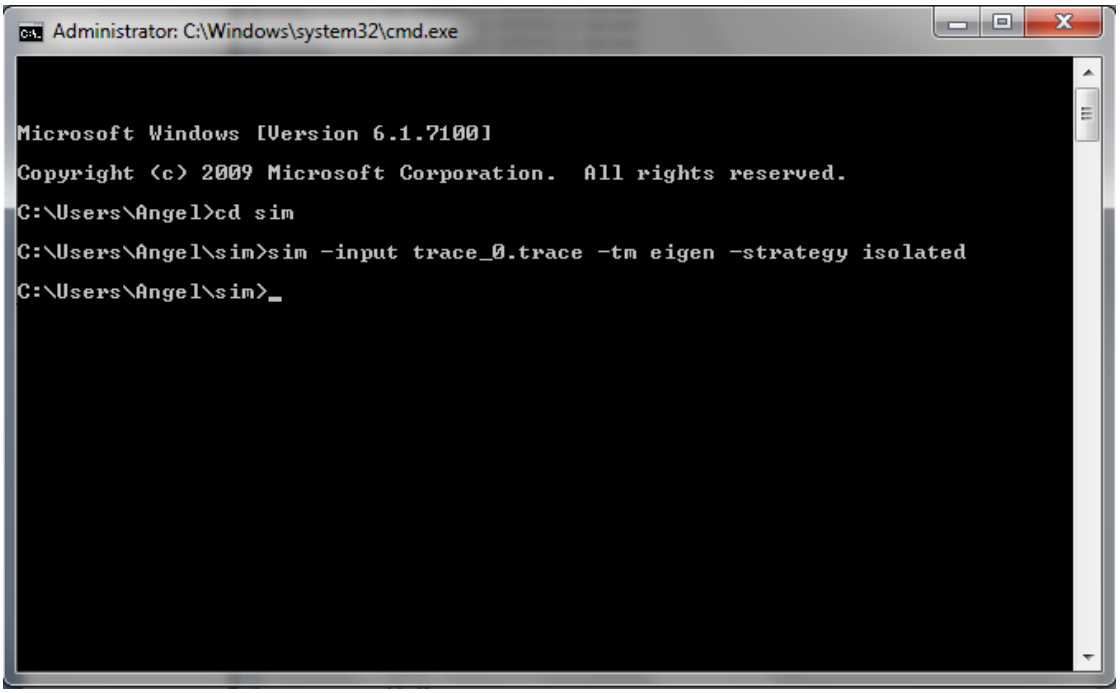
Σχήμα Α.1 : Παραγωγή trace στη γραμμή εντολών

Δίνουμε στην γραμμή εντολών τα στοιχεία του δικτύου σύμφωνα με τα ορίσματα του Πίνακα Α.1 και τρέχουμε το εκτελέσιμο αρχείο trace. Το αποτέλεσμα είναι η δημιουργία του αρχείου ίχνους του σχήματος Α.2.



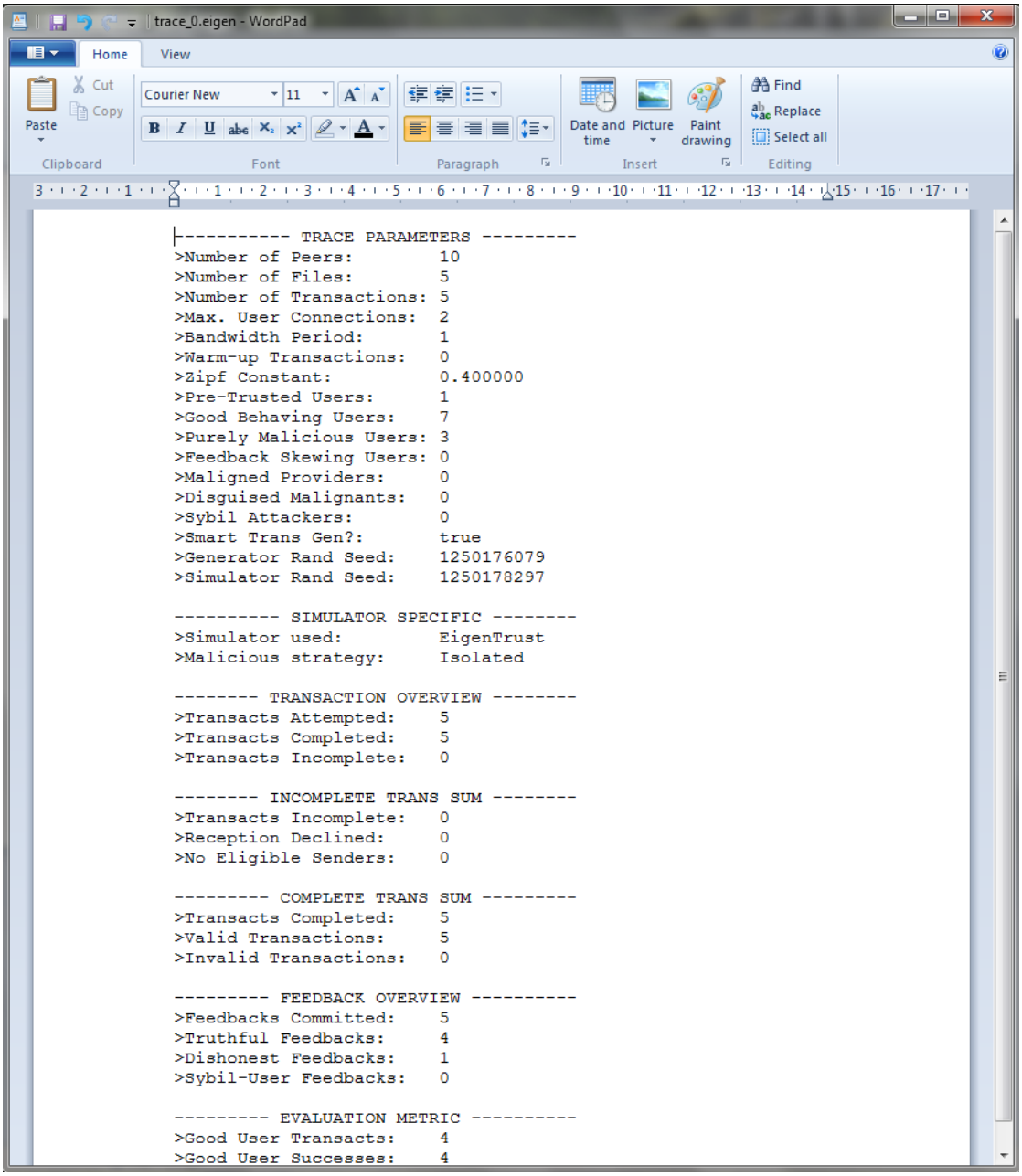
Σχήμα Α.2 : Αρχείο ίχνους trace_0.trace

Έπειτα προχωρούμε στην προσομοίωση του αρχείου ίχνους. Για να το κάνουμε αυτό χρειαζόμαστε κάποιον αλγόριθμο φήμης που θα επιτελέσει το ρόλο του διαχειριστή εμπιστοσύνης ανάμεσα στις συναλλαγές των χρηστών. Εμείς, όπως αναφέραμε, έχουμε υλοποιήσει στο σύστημά μας τον EigenTrust, τον ROCQ και τον Bayesian. Βεβαίως, μπορούμε να συνεχίσουμε και χωρίς την μεσολάβηση κάποιου αλγόριθμου. Κατόπιν, επιλέγουμε την κακόβουλη στρατηγική που θα υιοθετήσουν οι κακόβουλοι χρήστες του συστήματος. Τα ορίσματα αυτά τα δίνουμε στην γραμμή εντολών όπως διακρίνουμε και στην εικόνα του σχήματος Α.3.



Σχήμα Α.3 : Προσομοίωση αλγορίθμου στη γραμμή εντολών

Η έξοδος της προσομοίωσης που τρέξαμε είναι ένα αρχείο trace_0.eigen, αφού επιλέξαμε τον αλγόριθμο φήμης EigenTrust. Το αρχείο αυτό περιέχει στατιστικά αποτελέσματα και δίνει την πλήρη εικόνα της προσομοίωσης. Στο σχήμα Α.4 που ακολουθεί απεικονίζεται το αρχείο στατιστικών της προσομοίωσης που τρέξαμε.



Σχήμα Α.4 : Στατιστικά αποτελέσματα προσομοίωσης

Αφού επεξηγήσαμε τον τρόπο με τον οποίο εκτελούμε τις προσομοιώσεις, παρουσιάζουμε στη συνέχεια τους πηγαίους κώδικες του αλγορίθμου εμπιστοσύνης EigenTrust, που έχει υλοποιηθεί από τον Andrew West του πανεπιστημίου της Πενσυλβανία και των αλγορίθμων εμπιστοσύνης ROCQ και Bayesian που υλοποιήθηκαν για τον σκοπό της διπλωματικής. Οι κώδικες είναι σε γλώσσα προγραμματισμού C, η οποία μας παρέχει αρκετά γρήγορη εκτέλεση προσομοιώσεων[10]. Να τονίσουμε πως για ευνόητους λόγους δεν παραθέτουμε τον κώδικα του P2P Simulator [11].

EigenTrust

```
// Andrew West - tsys_eigen.h - Header file for tsys_eigen.c

#ifndef TSYS_EIGEN_H
#define TSYS_EIGEN_H

// ------------------------- LOCAL INCLUDES ------------------------------


// ----------------------- FUNCTION PROTOTYPES ---------------------------

void eigen_compute_trust(Network* nw, int user, int cycle);
void eigen_update(Network* nw, int new_vec, int new_row);

void eigen_initialize(User_Library* ulib);
void eigen_normalize_vector(User_Library* ulib, int new_vec);
float* eigen_trust_multiply(User_Library* ulib, int user, int max_iters);
void eigen_single_multiply(float* source_vec, float* dest_vec);

int eigen_converged(float* vec1, float* vec2);
int eigen_fback_int(Relation* rel);

void lalg_vector_matrix_mult(float* vector, float** matrix, float* dest_vec);
void lalg_constant_vector_mult(float constant, float* vector, float* dest_vec);
void lalg_vector_add(float* vector1, float* vector2, float* dest_vec);

#endif
```

[10] Το project που υλοποιήσαμε μπορεί να βρεθεί στην διεύθυνση : http://pithos.grnet.gr/pithos/rest/el04120@ntua.gr/files/%CE%9A%CE%BF%CE%B9%CE%BD%CF%8C%CF%87%CF%81%CE%B7%CF%83%CF%84%CE%BF%CF%82/Simulation.rar

[11] Ο πηγαίος κώδικας του P2P Simulator υπάρχει στις γλώσσες προγραμματισμού Java και C στην ιστοσελίδα : http://rtg.cis.upenn.edu/qtm/p2psim.php3

```
// Andrew West - tsys_eigen.c - Abstraction of EigenTrust system

// --------------------------- LOCAL INCLUDES -----------------------------

#include "sim_globals.h"

// ---------------------------- GLOBAL VARIABLES -----------------------------

const float ALPHA = 0.5;
const float EPSILON = 0.001;

float* pretrust;
float* vectorA;
float* vectorB;
float** normals;

float* scratch_vec;

// ------------------------- HIGH-LEVEL TRUST CALLS -------------------------

void eigen_compute_trust(Network* nw, int user, int cycle){
	if(cycle == 0)
		eigen_initialize(nw->ulib);
	else // Setup infrastructure on first call
		eigen_trust_multiply(nw->ulib, user, 8); // Compute trust thereafter
	return;
}

void eigen_update(Network* nw, int new_vec, int new_row){
	eigen_normalize_vector(nw->ulib, new_vec);
	return;
}

// ----------------------------- TRUST HELPERS ------------------------------

void eigen_initialize(User_Library* ulib){

	pretrust = (float*) calloc(NUM_USERS, sizeof(float));
	vectorA = (float*) calloc(NUM_USERS, sizeof(float));
	vectorB = (float*) calloc(NUM_USERS, sizeof(float));
	normals = (float**) calloc(NUM_USERS, sizeof(float*));
	if(pretrust == NULL || vectorA == NULL || vectorB == NULL || normals == NULL){
		printf("\nMemory allocation failed. Aborting.\n\n");
		exit(1);
	} // Allocate 2-D pointer setup for normalization/multiplication matrices

	scratch_vec = (float*) calloc(NUM_USERS, sizeof(float));

if(scratch_vec == NULL){
		printf("\nMemory allocation failed. Aborting.\n\n");
		exit(1);
	} // Setup some buffer space to avoid constant re-allocation
```

```
	int i, j;
	for(i=0; i < NUM_USERS; i++){
		normals[i] = (float*) calloc(NUM_USERS, sizeof(float));
		if(normals[i] == NULL){
			printf("\nMemory allocation failed. Aborting.\n\n");
			exit(1);
		} // Allocate row dimension of the matrix
	} // Create simple matrix to store normalized vals

	for(i=0; i < NUM_USERS; i++){
		if((PRE_TRUSTED > 0) && (ulib->users[i].pre_trusted))
			pretrust[i] = (1.0 / PRE_TRUSTED);
		else if(PRE_TRUSTED > 0) // and not pre-trusted
			pretrust[i] = 0.0;
		else // (there are no pre-trusted users)
			pretrust[i] = (1.0 / NUM_USERS);

		for(j=0; j < NUM_USERS; j++)
			normals[i][j] = pretrust[i];
	} // Setup the persistent normalized and pre-trusted structs
	return;
}

void eigen_normalize_vector(User_Library* ulib, int new_vec){
	int i, fback_int;
	int normalizer = 0;
	for(i=0; i < NUM_USERS; i++){
		fback_int = eigen_fback_int(&ulib->users[new_vec].vector[i]);
		normalizer += fback_int;
		normals[i][new_vec] = fback_int;
	} // Calculate normalizing sum in first pass

	if(normalizer == 0){
		for(i=0; i < NUM_USERS; i++)
			normals[i][new_vec] = pretrust[i];
	} else{ // If a user trusts no one, default to the pre_trust vector
		for(i=0; i < NUM_USERS; i++)
			normals[i][new_vec] /= (normalizer*1.0);
	} // Else, do the normalizing division in a second pass

	return;
}

float* eigen_trust_multiply(User_Library* ulib, int user, int max_iters){
	eigen_single_multiply(pretrust, vectorA);
	max_iters--;
	do{ // Multiply until convergence or maximum iters reached
		eigen_single_multiply(vectorA, vectorB);
		eigen_single_multiply(vectorB, vectorA);
		max_iters -= 2;
	} while((max_iters > 0) && !eigen_converged(vectorA, vectorB));
```

```
int i;
        for(i=0; i < NUM_USERS; i++){
                ulib->users[user].vector[i].trust_val = vectorA[i];
        } // Import trust values back into Object form
        return vectorA;
}

void eigen_single_multiply(float* source_vec, float* dest_vec){
        lalg_vector_matrix_mult(source_vec, normals, scratch_vec);
        lalg_constant_vector_mult((1-ALPHA), scratch_vec, scratch_vec);
        lalg_constant_vector_mult(ALPHA, pretrust, dest_vec);
        lalg_vector_add(scratch_vec, dest_vec, dest_vec);
        return;
}

int eigen_converged(float* vec1, float* vec2){
        int i;
        for(i=0; i < NUM_USERS; i++){
                if(abs(vec1[i]-vec2[i]) > EPSILON)
                        return 0;
        } // Compare vector elements, examining delta change
        return 1;
}

int eigen_fback_int(Relation* rel){
        int fback_int = *rel->pos - *rel->neg;
        if(fback_int < 0)
                fback_int = 0;
        return fback_int;
}

// ------------------------- LINEAR ALGEBRA STUFF -------------------------

void lalg_vector_matrix_mult(float* vector, float** matrix, float* dest_vec){
        int i,j;
        for(i=0; i < NUM_USERS; i++){
                dest_vec[i] = 0.0;
                for(j=0; j < NUM_USERS; j++){
                        dest_vec[i] += (matrix[i][j] * vector[j]);
                } // Inner loop of matrix-vector multiplication
        } // Outer loop of matrix-vector multiplication
        return;
}

void lalg_constant_vector_mult(float constant, float* vector, float* dest_vec){
        int i;
        for(i=0; i < NUM_USERS; i++){
                dest_vec[i] = vector[i] * constant;
        } // Just multiply every vector element by the constant
        return;
}
```

```
void lalg_vector_add(float* vector1, float* vector2, float* dest_vec){
        int i;
        for(i=0; i < NUM_USERS; i++){
                dest_vec[i] = vector1[i] + vector2[i];
        } // Just add the elements at correspondong positions

        return;
}
```

ROCQ

Ο κώδικας που δημιουργήθηκε για την υλοποίηση του αλγορίθμου ROCQ αποτελείται από τις δύο κύριες κλήσεις συναρτήσεων **rocq-update( )** για την ενημέρωση των τιμών εμπιστοσύνης και την **rocq_trust_equation( )** για τον υπολογισμό των τιμών εμπιστοσύνης. Να επισημάνουμε πως για την ενημέρωση του δείκτη ποιότητας Quality έγινε χρήση της Incomplete Beta Function[12].

```
// Angel Kapoukakis - tsys_rocq.h - Header file for tsys_rocq.c

#ifndef TSYS_ROCQ_H
#define TSYS_ROCQ_H


// ------------------------ FUNCTION PROTOTYPES -----------------------------

void rocq_compute_trust(Network* nw, int user, int cycle);
void rocq_update(Network* nw, int new_vec, int new_row);
void rocq_initialize(User_Library* ulib);
void rocq_trust_equation(User_Library* ulib, int user);
void rocq_opinion_update(User_Library* ulib, int new_vec, int new_row);
void rocq_deviation_update(User_Library* ulib, int new_vec, int new_row);
void rocq_quality_update(User_Library* ulib, int new_vec, int new_row);
void rocq_credibility_update(int new_vec, int new_row);
void rocq_reputation_update(int new_vec, int new_row);
int rocq_transactionsNo (Relation* rel);
int rocq_fback_int(Relation* rel);

#endif
```

[12] http://mymathlib.webtrellis.net/functions/gamma_beta.html

```
// Angel Kapoukakis - tsys_rocq.c - Abstraction of ROCQ system

#include <math.h>
#include <stdio.h>
#include <stdlib.h>

#include "sim_globals.h"


// ----------------------Externally Defined Routines----------------------//


extern double Incomplete_Beta_Function(double x, double a,double b);


// ------------------------- GLOBAL VARIABLES ----------------------------//

const double r = 10.0;
double** opinion;
double** reputation;
double** credibility;
double** quality;
double** st_deviation;
double* pretrust;
double* vectorA;


// ------------------------- HIGH-LEVEL TRUST CALLS -------------------------//


void rocq_compute_trust(Network* nw, int user, int cycle){
        if(cycle == 0) rocq_initialize(nw->ulib);
        else rocq_trust_equation(nw->ulib, user);
        return;
}


void rocq_update(Network* nw, int new_vec,int new_row){

   rocq_opinion_update(nw->ulib, new_vec, new_row);
   rocq_deviation_update(nw->ulib, new_vec, new_row);
   rocq_quality_update(nw->ulib, new_vec, new_row);
   rocq_credibility_update(new_vec, new_row);
   rocq_reputation_update(new_vec, new_row);

        return;
}
```

```
// ---------------------- ΑΡΧΙΚΟΠΟΙΗΣΗ ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ ----------------------//

void rocq_initialize(User_Library* ulib){

        opinion = (double**) calloc(NUM_USERS, sizeof(double));
        reputation = (double**) calloc(NUM_USERS, sizeof(double));
        credibility = (double**) calloc(NUM_USERS, sizeof(double));
        quality = (double**) calloc(NUM_USERS, sizeof(double*));
        st_deviation = (double**) calloc(NUM_USERS, sizeof(double*));

        if (opinion == NULL || reputation == NULL || credibility == NULL || quality == NULL
        || st_deviation == NULL){
                printf("\nMemory allocation failed. Aborting.\n\n");
                exit(1);
        }

        pretrust = (double*) calloc(NUM_USERS, sizeof(double));
        vectorA = (double*) calloc(NUM_USERS, sizeof(double));

        if(pretrust == NULL || vectorA == NULL){
                printf("\nMemory allocation failed. Aborting.\n\n");
                exit(1);
        }

        int i, j;

        for(i=0; i < NUM_USERS; i++){

                opinion[i] = (double*) calloc(NUM_USERS, sizeof(double));
                reputation[i] = (double*) calloc(NUM_USERS, sizeof(double));
                credibility[i] = (double*) calloc(NUM_USERS, sizeof(double));
                quality[i] = (double*) calloc(NUM_USERS, sizeof(double));
                st_deviation[i] = (double*) calloc(NUM_USERS, sizeof(double));

                if (opinion[i] == NULL || reputation[i] == NULL || credibility[i] == NULL ||
                quality[i] == NULL || st_deviation == NULL)  {
                        printf("\nMemory allocation failed. Aborting.\n\n");
                        exit(1);
                }
        }

        for(i=0; i < NUM_USERS; i++){
                if((PRE_TRUSTED > 0) && (ulib->users[i].pre_trusted))
                        pretrust[i] = (1.0 / PRE_TRUSTED);

                else if(PRE_TRUSTED > 0)
                        pretrust[i] = 0.0;
                else
                        pretrust[i] = (1.0 / NUM_USERS);
```

```
                for(j=0; j < NUM_USERS; j++) {
                        reputation[i][j] = pretrust[j];
                        credibility[i][j] = 0.5;
                        quality[i][j] = 1.0;
        }

    quality[i][i] = 0.0;
}
        return;
}


// -------------------------ΥΠΟΛΟΓΙΣΜΟΣ ΤΙΜΩΝ ΕΜΠΙΣΤΟΣΥΝΗΣ-----------------------------//


void rocq_trust_equation(User_Library* ulib, int user){
        int j, y;
        double sum1, sum2;

        for(j=0; j < NUM_USERS; j++){
            sum1=0, sum2=0;
            for (y=0; y < j; y++) {
                sum1+=(reputation[y][j]*credibility[user][y]*quality[y][j]);
                sum2+=(credibility[user][y]*quality[y][j]);
                                }

        for (y=j+1; y < NUM_USERS; y++) {
                sum1+=(reputation[y][j]*credibility[user][y]*quality[y][j]);
                sum2+=(credibility[user][y]*quality[y][j]); }
                vectorA[j] = sum1/sum2;

    }

         int i;
        for(i=0; i < NUM_USERS; i++){
                ulib->users[user].vector[i].trust_val = vectorA[i];
        }
        return;
}


// -------------------------ΕΝΗΜΕΡΩΣΗ ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ-----------------------------//


void rocq_opinion_update(User_Library* ulib, int new_vec, int new_row) {
        int N, fback_int;
        fback_int = rocq_fback_int(&ulib->users[new_vec].vector[new_row]);
        N = rocq_transactionsNo(&ulib->users[new_vec].vector[new_row]);
        if (N!=0) opinion[new_vec][new_row] = (fback_int*1.0)/(N*1.0);
        return;
}
```

```
int rocq_fback_int(Relation* rel){
        int fback_int = *rel->pos ;
        return fback_int;
}


// ------------------------------------------------------------------------------------------------//


void rocq_deviation_update(User_Library* ulib, int new_vec, int new_row){
        int N, pos, neg;
        double o, sum, s, sd;

        N = rocq_transactionsNo(&ulib->users[new_vec].vector[new_row]);
        pos = *ulib->users[new_vec].vector[new_row].pos;
        neg = *ulib->users[new_vec].vector[new_row].neg;
        o = opinion[new_vec][new_row];
        sum = (pos*(1-o)*(1-o)) + (neg*o*o);

        if (N!=0) {
                s = sum/(N*1.0);
                sd = sqrt(s);
                st_deviation[new_vec][new_row] = sd; }
                return;
}


// ------------------------------------------------------------------------------------------------//

void rocq_quality_update(User_Library* ulib, int new_vec, int new_row){
   int N;
   double t, z, alpha, B;

   N = rocq_transactionsNo(&ulib->users[new_vec].vector[new_row]);

   if ((N==1) || (st_deviation[new_vec][new_row]==0))
        quality[new_vec][new_row] = 1.0;

   else {

     t= (r*opinion[new_vec][new_row]*sqrt(N))/(100*st_deviation[new_vec][new_row]);
     z = ((N-1)*1.0)/(N-1+t*t);
     alpha = ((N-1)*1.0)/2.0;
     B = Incomplete_Beta_Function(z, alpha, 0.5);

   if (B>1.0) {

     if ((N>4) && (opinion[new_vec][new_row] <0.5)) quality[new_vec][new_row] = 0.1;
```

```
      else
      if ((N>4) && (opinion[new_vec][new_row] >=0.5))
        quality[new_vec][new_row] = 0.9;
      else quality[new_vec][new_row] = 0.75; }
 else quality[new_vec][new_row] = 1.0 - B; }
return;
}


int rocq_transactionsNo (Relation* rel) {
   int N = *rel->pos + *rel->neg;
   return N;
}


// ------------------------------------------------------------------------------------------------------//


void rocq_credibility_update(int new_vec, int new_row) {
   int m;
   double cred, cred_next1, cred_next2 ;

   for(m=0; m < new_vec; m++){
      cred = reputation[m][new_row] - opinion[new_vec][new_row];
      if ((fabs(cred) == 0) && (st_deviation[m][new_row]==0));

      else
      if (fabs(cred) < st_deviation[m][new_row]) {

      cred_next1 = ((1.0-credibility[m][new_vec])*(quality[new_vec][new_row]))/2.0;
      cred_next2 = 1.0- fabs(cred)/st_deviation[m][new_row];
      cred_next1 = credibility[m][new_vec] + cred_next1*cred_next2;
      credibility[m][new_vec] = cred_next1;
      }

      else {

        cred_next1 = (credibility[m][new_vec]*quality[new_vec][new_row])/2.0;
        cred_next2 = 1.0- ((st_deviation[m][new_row])/fabs(cred));
        cred_next1= credibility[m][new_vec] - cred_next1*cred_next2;
        credibility[m][new_vec] = cred_next1;
      }
   }

   for(m=new_vec+1; m < NUM_USERS; m++){
      cred = reputation[m][new_row] - opinion[new_vec][new_row];

      if ((fabs(cred) == 0) && (st_deviation[m][new_row]==0));
```

```
      else

      if (fabs(cred) < st_deviation[m][new_row]) {
         cred_next1 = (1.0-credibility[m][new_vec])*quality[new_vec][new_row]/2.0;
         cred_next2 = 1.0- fabs(cred)/st_deviation[m][new_row];
         cred_next1 = credibility[m][new_vec] + cred_next1*cred_next2;
         credibility[m][new_vec] = cred_next1;
      }

      else {
         cred_next1 = (credibility[m][new_vec]*quality[new_vec][new_row])/2.0;
         cred_next2 = 1.0- ((st_deviation[m][new_row])/fabs(cred));
         cred_next1= credibility[m][new_vec] - cred_next1*cred_next2;
         credibility[m][new_vec] = cred_next1;
       }
     }
   return;
}


// ------------------------------------------------------------------------------------------------------------//


void rocq_reputation_update(int new_vec, int new_row) {
    int m, i;
    double sum1, sum2;

    for(m=0; m < new_row; m++){
        sum1=0; sum2=0;

        for (i=0; i <NUM_USERS; i++) {

          sum1+= (opinion[i][new_row] * credibility[m][i] * quality[i][new_row]);
          sum2+= (credibility[m][i] * quality[i][new_row]); }
          reputation[m][new_row]= sum1/sum2;
        }


    for(m=new_row+1; m < NUM_USERS; m++){
       sum1=0; sum2=0;

       for (i=0; i <NUM_USERS; i++) {

         sum1+= (opinion[i][new_row] * credibility[m][i] * quality[i][new_row]);
         sum2+= (credibility[m][i] * quality[i][new_row]); }
         reputation[m][new_row]= sum1/sum2;
       }
    return;
}
```

Bayesian

```
// Angel Kapoukakis - tsys_bayesian.h - Header file for tsys_bayesian.c

#ifndef TSYS_BAYESIAN_H
#define TSYS_BAYESIAN_H



// ------------------------ FUNCTION PROTOTYPES -----------------------------//

void bayesian_compute_trust(Network* nw, int user, int cycle);
void bayesian_update(Network* nw, int new_vec, int new_row);
void bayesian_initialize(User_Library* ulib);
void bayesian_trust_equation(User_Library* ulib, int user);
int bayesian_transactionsNo (Relation* rel);

#endif
```

```
// Angel Kapoukakis - tsys_bayesian.c - Abstraction of a Bayesian system


// ---------------------------- LOCAL INCLUDES ------------------------------

#include "sim_globals.h"


// ------------------------- GLOBAL VARIABLES -------------------------------

const double a = 0.3;
const double wt = 0.75;
const double ws = 0.25;

double** tr;
double* rec;
double* pretrust;
int inv_trans=0;

// ------------------------ HIGH-LEVEL TRUST CALLS --------------------------

void bayesian_compute_trust(Network* nw, int user, int cycle){
        if(cycle == 0)  bayesian_initialize(nw->ulib);
        else bayesian_trust_equation(nw->ulib, user);
        return;
}
```

```
void bayesian_update(Network* nw, int new_vec,int new_row){
    int j,  x=0;
    double ea, trust;

    if (inv_trans!=NUM_INVAL_TRANS) x=1;
    inv_trans=NUM_INVAL_TRANS;

     for (j=0; j < new_row; j++) {
        trust=tr[new_vec][j];

                if ((x==1) && (tr[j][new_row]>=0.5) ) ea=-1.0;
        else
        if ((x==1) && (tr[j][new_row]<0.5) ) ea=1.0;
        else
        if ((x==0) && (tr[j][new_row]<0.5) ) ea=-1.0;
        else ea=-1.0;

        tr[new_vec][j]= a*trust + (1-a)*ea; }

     for (j=new_row+1; j < NUM_USERS; j++) {
        trust=tr[new_vec][j];

                if ((x==1) && (tr[j][new_row]>=0.5) ) ea=-1.0;
        else
        if ((x==1) && (tr[j][new_row]<0.5) ) ea=1.0;
        else
        if ((x==0) && (tr[j][new_row]<0.5) ) ea=-1.0;
        else ea=-1.0;

        tr[new_vec][j]= a*trust + (1-a)*ea; }

          return;
}

// ----------------- ΑΡΧΙΚΟΠΟΙΗΣΗ ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ ----------------------------

void bayesian_initialize(User_Library* ulib){
          //printf("mesa sto initialize");
          tr = (double**)calloc(NUM_USERS, sizeof(double));

          if(tr == NULL){
                    printf("\nMemory allocation failed. Aborting.\n\n");
                    exit(1);
          }
          //printf("orisame r kai pretrust");
          pretrust = (double*)calloc(NUM_USERS, sizeof(double));
          rec = (double*)calloc(NUM_USERS, sizeof(double));
          if(rec == NULL || pretrust == NULL){
                    printf("\nMemory allocation failed. Aborting.\n\n");
                    exit(1);
          }
```

```
	int i, j;
	for(i=0; i < NUM_USERS; i++){
		tr[i] = (double*)calloc(NUM_USERS, sizeof(double));

		if (tr[i] == NULL){
			printf("\nMemory allocation failed. Aborting.\n\n");
			exit(1);
		}
	}

	for(i=0; i < NUM_USERS; i++){
		if((PRE_TRUSTED > 0) && (ulib->users[i].pre_trusted))
			pretrust[i] = (1.0 / PRE_TRUSTED);
		else if(PRE_TRUSTED > 0)
			pretrust[i] = 0.0;
		else
			pretrust[i] = (1.0 / NUM_USERS);
	rec[i] = 0.5;
		for(j=0; j < NUM_USERS; j++) tr[i][j] = pretrust[j];
		tr[i][i] = 1.0;
}
	return;
}

// ------------------ΥΠΟΛΟΓΙΣΜΟΣ ΤΙΜΩΝ ΕΜΠΙΣΤΟΣΥΝΗΣ----------------------------//

void bayesian_trust_equation(User_Library* ulib, int user){
	int j, y, k, g, N;
	double sum1, sum2, sum3;

	for(j=0; j < NUM_USERS; j++){
    sum1=0; sum2=0; sum3=0; k=0;g=0;

    for (y=0; y < j; y++) {
		N = bayesian_transactionsNo(&ulib->users[user].vector[y]);
		if (N!=0) {
      sum1+=tr[user][y]*tr[y][j];
      sum2+=tr[user][y]; }
     else {
      g++;
      sum3+= tr[y][j];
      }
     }
    for (y=j+1; y < NUM_USERS; y++) {
		N = bayesian_transactionsNo(&ulib->users[user].vector[y]);
		if (N!=0) {
      sum1+=tr[user][y]*tr[y][j];
      sum2+=tr[user][y]; }
```

```
        else {
          g++;
          sum3+= tr[y][j]; }

        if ((sum2!=0) && (g!=0))
        rec[j] = wt*(sum1/sum2) + ws*(sum3/(g*1.0));
        }
      }

   int i;

   for(i=0; i < NUM_USERS; i++) ulib->users[user].vector[i].trust_val = rec[i];
   return;
}



int bayesian_transactionsNo (Relation* rel) {
   int number = *rel->pos + *rel->neg;
   return number;
}
```

ΠΑΡΑΡΤΗΜΑ Β

Βιβλιογραφία

[1] K. Aberer and Z. Despotovic: *Managing Trust in a Peer-2-Peer Information System*. In Proceedings of the 10th International Conference on Information and Knowledge Management (ACM CIKM), New York, USA, 2001.

[2] A. Abdul-Rahman and S. Hailes: *Supporting Trust in Virtual Communities* Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.

[3] K. Aberer and Z. Despotovic: *P2P reputation management: Probabilistic estimation*, August 2005.

[4] Ν.Κοκκίνης, Λ.Κώνστα, Π.Νικολαΐδου: *Extended EigenTrust Algorithm for Reputation Management in P2P Networks*, Φεβρουάριος 2007.

[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia- Molina: *The EigenTrust algorithm for reputation management in P2P networks*. In Proc. of the Twelfth International World Wide Web Conference

[6] L. Xiong and L. Liu: A *reputation-based trust model for peer-to-peer ecommerce communities*. In IEEE Conference on E-Commerce (CEC'03), 2003.

[7] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati: *Choosing reputable servents in a p2p network*. In Eleventh International World Wide Web Conference, Honolulu, Hawaii, May 2002.

[8] R.Battiti: *Self-management in Autonomic Communication: Trust and Reputation*, 2004

[9] A. Garg and R. Battiti: *The reputation, opinion, credibility and quality (ROCQ) scheme*. Technical Report DIT-04-104, University of Trento, July 2004.

[10] A. Garg, R. Battiti, and R. Cascella: *Reputation management: Experiments on the robustness of ROCQ*. In Proceedings of the 7th International Symposium on Autonomous Decentralized Systems (First International Workshop on Autonomic Communication for Evolvable Next Generation Networks), pages 725.730, Chengdu, China, Apr. 2005.

[11] Yao Wang, Julita Vassileva: *Bayesian Network-Based Trust Model*. In Proceedings of Second International Workshop Peers and Peer-to-Peer Computing, July 14, 2003. Melbourne, Australia.

[12] Baptiste Pretre: *Attacks on Peer-to-Peer Networks*. Dept. of Computer Science Swiss Federal Institute of Technology (ETH) Zurich Autumn 2005.

[13] Jeffrey O.Kephart and David M.Chess: *The Vision of Autonomic Computing* IBM Thomas J. Watson Research Center, 2003.

[14] Rinkesh Patel: *Real-Time Trust Management for Agent Based Online Auction System*, Fall 2006.

[15] Β.Μάγκλαρης: *Διαχείριση δικτύων Ζητήματα ασφαλείας Μέρος Β'*, 2009.

[16] J.Goldbeck: *Computing with social trust*, Springer 2009.

[17] Lee Hyun-rok: *Multiple Selective Mutual Authentication Protocol For Peer-to-Peer System*, 2001.

[18] James F. Kurose & Keith W. Ross: *Computer Networking*, 2003

[19] Kerberos: http://www.islab.demokritos.gr

[20] Kerberos: www.adopenstatic.com/cs/blogs/ken/archive/2006/10/20/512.aspx

[21] Andrew G. West, Oleg Sokolsky et al, *An Evaluation Framework for Reputation Management Systems*, University of Pennsylvania.

[22] Andrew G. West, *Reputation Management Algorithms & Testing*, 2008.

[23] Wikipedia: http://en.wikipedia.org/wiki/Kerberos_(protocol)

[24] EBay: http://www.ebay.com.

[25] Wikipedia: http://en.wikipedia.org/wiki/Peer-to-peer

[26] Wikipedia: http://el.wikipedia.org/wiki/Peer-to-peer

[27] Wikipedia: http://en.wikipedia.org/wiki/Digital_signature

[28] Wikipedia: http://en.wikipedia.org/wiki/Denial-of-service_attack

[29] http://mymathlib.webtrellis.net/functions/gamma_beta.html