ΑΣΦΑΛΕΙΑ ΕΠΙΠΕΔΟΥ ΕΦΑΡΜΟΓΗΣ ΣΤΑ ΟΜΟΤΙΜΑ ΔΙΚΤΥΑ

Η Διπλωματική Εργασία
παρουσιάστηκε ενώπιον
του Διδακτικού Προσωπικού του
Πανεπιστημίου Αιγαίου

Σε Μερική Εκπλήρωση
των Απαιτήσεων για το Δίπλωμα του
Μηχανικού Πληροφοριακών και Επικοινωνιακών Συστημάτων

του
ΕΥΑΓΓΕΛΟΥ ΡΕΚΛΕΙΤΗ
ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2005

Η ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΔΙΔΑΣΚΟΝΤΩΝ ΕΓΚΡΙΝΕΙ
ΤΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ ΕΥΑΓΓΕΛΟΥ ΡΕΚΛΕΙΤΗ:

ΔΡ. ΣΤΕΦΑΝΟΣ ΓΚΡΙΤΖΑΛΗΣ, Επιβλέπων 9/2005
Πρόεδρος, Αναπληρωτής καθηγητής
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

Δρ. ΚΩΝΣΤΑΝΤΙΝΟΣ ΛΑΜΠΡΙΝΟΥΔΑΚΗΣ, Μέλος
Επίκουρος Καθηγητής
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

Δρ. ΣΠΥΡΟΣ ΚΟΚΟΛΑΚΗΣ, Μέλος
Λέκτορας
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2005

ΠΕΡΙΛΗΨΗ

Η διπλωματική αυτή εργασία, που τιτλοφορείται «Ασφάλεια Επιπέδου Εφαρμογής στα Ομότιμα Δίκτυα», έχει ως στόχο να παράσχει μια μη εξαντλητική επιθεώρηση των απαιτήσεων ασφαλείας που μπορούν να ζητηθούν σε δημοφιλείς ομότιμες εφαρμογές, ήτοι εφαρμογές που παρέχονται πάνω από ομότιμα δίκτυα και τα χαρακτηριστικά ασφάλειας που παρέχονται από υπάρχουσες ομότιμες εφαρμογές. Τέτοιες εφαρμογές περιλαμβάνουν το διαμοιρασμό αρχείων, την υπολογιστική κοινών πόρων, την άμεση επικοινωνία και τη κατανεμημένη συνεργασία. Δίνεται μια γενική περιγραφή των ομότιμων δικτύων και γίνεται ταξινόμησή τους, με κριτήριο τη δομή τους και το είδος των εφαρμογών που εξυπηρετούν.

ΕΥΑΓΓΕΛΟΣ ΡΕΚΛΕΙΤΗΣ
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
2005

ABSTRACT

The objective of this dissertation, entitled “Application Level Security in Peer-to-Peer Networks”, is to provide a non-exhaustive review of the security requirements that can be asked for in popular peer-to-peer applications, that is to say applications provided through peer-to-peer networks, and the security features existing peer-to-peer applications provide. Such applications include file-sharing, public-resource computing, instant communication and distributed collaboration. A generic description of peer-to peer networks is supplied along with a classification, based on their structure and the type of application they serve.

EVANGELOS REKLEITIS
Department of Information and Communication Systems Engineering
UNIVERSITY OF THE AEGEAN
2005

ΕΥΧΑΡΙΣΤΙΕΣ - ΑΦΙΕΡΩΣΕΙΣ

Ευχαριστώ.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΕΦΑΛΑΙΟ 0 - Δομή

«Ξέρω ότι δε ξέρω τίποτα, αλλά οι άλλοι ξέρουν ακόμα λιγότερα»
Zarko Petan, Svet v enem stavku, 1999

Σε αυτή τη διπλωματική θα ασχοληθώ με το ζήτημα της παροχής ασφάλειας επιπέδου εφαρμογής (application level security) σε σύγχρονα περιβάλλοντα ομότιμων (P2P, p-to-p) δικτύων. Στο πρώτο κεφάλαιο θα παρουσιάσω μια πανοραμική εικόνα του περιβάλλοντος των ομότιμων δικτύων. Πιο συγκεκριμένα στα υποκεφάλαια 1.1. και 1.2. δίνω μια πρακτική και επιπόλαια αίσθηση του τι μπορεί να θεωρηθεί ως ομότιμο σύστημα. Ακολούθως στο 1.3. παρατίθεται ένας πιο ολοκληρωμένος και συνεκτικός ορισμός του τι είναι ένα ομότιμο δίκτυο και μια ταξινόμηση αυτών. Στο υποκεφάλαιο 1.4. φαίνεται η διείσδυση των P2P τεχνολογιών στον ψηφιακό κόσμο. Αμέσως μετά, υποκεφάλαιο 1.5., δικαιολογώ την ανάγκη παροχής ασφάλειας σε αυτά τα συστήματα, δίνω ορισμένους χρήσιμους ορισμούς όρων του τομέα της ασφάλειας των πληροφοριακών και επικοινωνιακών συστημάτων, και εν συνεχεία παρουσιάζω γνωστές αδυναμίες και επιθέσεις των ομότιμων δικτύων.

Στο δεύτερο κεφάλαιο δίνεται μια επιλεκτική παράθεση σύγχρονων και αξιόλογων ομότιμων δικτύων μαζί με μια γενική περιγραφή των.

Στο τρίτο κεφάλαιο παρουσιάζονται σενάρια χρήσεως ομότιμων δικτύων και οι των αυτών απαιτήσεις ασφάλειας.

ΚΕΦΑΛΑΙΟ 1 - ΠΕΡΙΒΑΛΛΟΝ

«Οπότε εγώ λέω:
Ένας που ξέρει τον εχθρό του και ξέρει τον εαυτό του, δε θα κινδυνεύσει σε εκατό μάχες.
Ένας που δε ξέρει τον εχθρό του αλλά ξέρει τον εαυτό του, κάποιες φορές θα νικάει, κάποιες φορές θα ηττάται.
Ένας που δε ξέρει τον εχθρό του και δε ξέρει τον εαυτό του, θα βρίσκεται σε κίνδυνο σε κάθε μάχη»
Η Τέχνη του Πολέμου, 孫子 Σουν Τζου, (400-320 πΧ)

1.1 Εισαγωγή

Ομότιμο δίκτυο, αγγλιστί peer-to-peer network (P2P, P-to-P,…), ονομάζουμε κάθε δίκτυο υπολογιστών (συσκευών με υπολογιστική ικανότητα) που δε στηρίζει την επικοινωνία του σε αφοσιωμένους εξυπηρέτες (dedicated servers), αλλά αντίθετα χρησιμοποιεί, κυρίως, απευθείας συνδέσεις μεταξύ των πελατών, που ονομάζονται ομότιμοι κόμβοι (peer nodes). Ένα ομότιμο δίκτυο στην πιο ακραιφνή του μορφή δεν εμπεριέχει των διαχωρισμό ανάμεσα σε πελάτες και εξυπηρέτες, υπάρχουν μόνο ομότιμοι κόμβοι που ταυτόχρονα παίζουν το ρόλο του εξυπηρετούμενου και του εξυπηρέτη ως προς τους υπόλοιπους κόμβους. Ωστόσο στα περισσότερα ομότιμα δίκτυα γίνεται χρήση κεντρικών οντοτήτων (central entities), προκειμένου να παρασχεθούν αποδοτικότερα ορισμένες υπηρεσίες.

Στο τυπικό μοντέλο εξυπηρετούμενου-εξυπηρέτη (client-server) η επικοινωνία μεταδίδεται δια των εξυπηρετών, και συχνά η απευθείας/άμεση επικοινωνία μεταξύ των εξυπηρετούμενων δεν είναι εφικτή. Σε αντιδιαστολή στα ομότιμα δίκτυα κάθε κόμβος δύναται να εκκινήσει ή να ολοκληρώσει κάθε υποστηριζόμενη συναλλαγή με κάποιον άλλο ομότιμό του. Ακόμα αντίθετα από τις αρχιτεκτονικές MIMD και SIMD οι ομότιμοι κόμβοι είναι αυτόνομοι και δεν απαιτούν κεντρική διαχείριση, δε χρειάζεται να ούτε καν να έχουν παρόμοιες τεχνικές προδιαγραφές. Οι ομότιμοι κόμβοι μπορούν και διαφέρουν στις δυνατότητες (επεξεργαστική ισχύ, αποθηκευτικός χώρος, ταχύτητα δικτύου) τους, χωρίς καμία βλάβη της απόδοσης του συστήματος και χωρίς να απαιτείται ιδιαίτερος κόπος προκειμένου να επιτευχθεί η διαλειτουργικότητα (interoperability) (technicality: Διαφορετικά λειτουργικά συστήματα χρειάζονται τις δικές τους εκδόσεις ομότιμων

εφαρμογών, εκτός και αν έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού Java, καθώς επίσης προκειμένου να αξιοποιηθεί στο έπακρο το υλικό (hardware) απαιτείται ξεχωριστή μεταγλώττιση (compilation) ανάλογα με τις τεχνικές προδιαγραφές).

Ο όρος peer-to-peer τυπικά μπορεί να προσδοθεί σε κάθε δικτυακή τεχνολογία ή\και εφαρμογή που χρησιμοποιεί το προαναφερθέν μοντέλο επικοινωνίας, συμπεριλαμβανομένου του ARPANET, του πρωτοκόλλου SMTP για την διεκπεραίωση του ηλεκτρονικού ταχυδρομείου και του πρωτοκόλλου NNTP που χρησιμοποιείται για τη μεταφορά νέων στο Usenet. Στη πράξη, ο όρος χρησιμοποιείται κυρίως ως χαρακτηρισμός σε δίκτυα διαμοιρασμού αρχείων (file sharing networks), τα οποία παρέχουν δωρεάν και σχετικά ανώνυμες υπηρεσίες μεταφοράς περιεχομένου (content) μεταξύ προσωπικών υπολογιστών πάνω από το Internet. Παρότι μακράν η προσφιλέστερη, η ανταλλαγή αρχείων είναι μια μόνον εκ των εφαρμογών που μπορούν να αξιοποιήσουν αυτό το μοντέλο απευθείας επικοινωνίας, ένας μη εξαντλητικός κατάλογος εφαρμογών θα μπορούσε να συμπεριλάβει την υπολογιστική κοινών πόρων (public-resource computing), την κατανεμημένη συνεργασία (distributed collaboration), την απευθείας επικοινωνία (instant communication), την κατανεμημένη αποθήκευση (distributed storage) ή ακόμα και την κατανεμημένη αναζήτηση (distributed searching).

Ένα ενδιαφέρον χαρακτηριστικό των ομότιμων συστημάτων είναι ότι οι υπολογιστικοί πόροι των ομότιμων κόμβων (επεξεργαστική ισχύ, χωρητικότητα, εύρος δικτυακής ζεύξης, κτλ.) συναθροίζονται, βελτιώνοντας την ποιότητα των προσφερόμενων υπηρεσιών. Σε αντίθεση στο μοντέλο εξυπηρετούμενου-εξυπηρέτη οι (πολλοί) χρήστες αναγκάζονται να διαμοιράζονται τους περιορισμένους πόρους των (λίγων) εξυπηρετών.

1.2 Ιστορική Αναδρομή

Η έννοια της επικοινωνίας μεταξύ ομότιμων οντοτήτων είναι τόσο παλιά όσο και το ίδιο το Internet [1]. Ο στόχος του αρχικού ARPANET (τέλη δεκαετίας 60) ήταν να διαμοιράσει υπολογιστικούς πόρους στις ΗΠΑ. Δίκτυα όπως το USENET (1979) και πρωτόκολλα σαν το DNS (Domain Name System) αναπτύχθηκαν με βάση τη φιλοσοφία της ομότιμης επικοινωνίας. Οι πρώτοι κόμβοι του Internet δεν συνδέονταν με σχέση αφέντη-υπηρέτη, αλλά μάλλον ως ομότιμοι πόροι και είχαν τη δυαδική ιδιότητα του εξυπηρέτη-εξυπηρετούμενου, αυτό ίσχυε μέχρι τα μέσα της δεκαετίας του 90 όταν η έλευση των αναχωμάτων ασφαλείας, των δυναμικών διευθύνσεων IP (Internet Protocol) και του NAT (Network Address Translation) ήρθε να κατακερματίσει το Internet και να περιορίσει τους κόμβους των χρηστών σε εξυπηρετούμενους καταναλωτές

περιεχομένου, κατά το πρότυπο της τηλεόρασης. Η αλματώδης ανάπτυξη του Internet, κατά το δεύτερο μισό της δεκαετίας του 90, βασίστηκε στο μοντέλο εξυπηρετούμενου-εξυπηρέτη, καθώς οι εκατομμύρια νέοι χρήστες του δικτύου στερούνταν κινήτρων και γνώσης, ικανής να προωθήσει εναλλακτικά μοντέλα, που θα τους έδιναν έναν πιο ενεργό ρόλο στη διαμόρφωση του περιεχομένου του δικτύου.

Ωστόσο η μεγάλη ανάπτυξη των δυνατοτήτων των προσωπικών υπολογιστών, οι εξελίξεις στην οπτική τεχνολογία, την τεχνολογία των ολοκληρωμένων κυκλωμάτων, η παροχή καλύτερων υπηρεσιών επικοινωνιών έχει σαν αποτέλεσμα να συσσωρευτεί στις παρυφές του Internet, στα χέρια απλών χρηστών, ένα τεράστιο πλήθος υπολογιστικών πόρων, που τις περισσότερες ώρες της ημέρας έμεναν ανεκμετάλλευτοι. Ήδη, από τις αρχές του 1997, το δίκτυο Distributed.Net [2], της μετέπειτα Distributed Computing Technologies Inc., άρχισε να χρησιμοποιεί αυτούς τους υπολογιστικούς πόρους. Στόχος του δικτύου είναι η εύρεση του 56-bit μυστικού κλειδιού της πρόκλησης RC5-32/12/7 των RSA Labs [3], το οποίο και πετυχαίνει ύστερα από 250 ημέρες εξαντλητικής αναζήτησης (brute force attack). Όμως η μεγάλη δημοτικότητα των ομότιμων δικτύων αρχίζει το 1999 με τη δημιουργία του Napster [4] [5], για το διαμοιρασμό αρχείων mp3, από τον τότε δεκαοχτάχρονο Shawn Fanning. Επίσης τον Μάιο του ίδιου έτους ξεκινάει και το δημοφιλέστερο ομότιμο σύστημα κατανεμημένης επεξεργασίας το SETI@home [6], [7] που είχε σαν σκοπό την ανάλυση ραδιοσημάτων, για την εύρεση εξωγήινης νοημοσύνης. Η ίδια η επιτυχία του Napster τελικά επιφέρει και τη πτώση του, αφού πολλοί καλλιτέχνες και δισκογραφικές εταιρίες κινήθηκαν δικαστικά εναντίον του με τελικό αποτέλεσμα τον τερματισμό της δωρεάν λειτουργίας του. Έκτοτε πολλές εφαρμογές προσπάθησαν να προσφέρουν ομότιμες υπηρεσίες και ως εκ τούτου πολλά νέα ομότιμα δίκτυα προτάθηκαν, το καθένα με διαφορετικές απαιτήσεις αλλά και λύσεις σε κοινά προβλήματα. Δημιουργήθηκαν αμιγώς ομότιμα δίκτυα διαμοιρασμού αρχείων, όπως το (αρχικό) Gnutella [7], [5], [8], [9], [10], το Overnet [11], [5] και το Kademlia [12], [13], [5], αλλά και, λιγότερα κεντρικοποιημένα, υβριδικά ομότιμα δίκτυα, όπως το FastTrack [14], [5], το eDonkey2000 [11], [15], [5] και το Bittorrent [16], [17], [5]. Επίσης προτάθηκαν πλήθος ομότιμων δικτύων που κάλυπταν πιο εξειδικευμένες απαιτήσεις, όπως η προστασία της ιδιωτικότητας και η ανωνυμία, για παράδειγμα τα Free Haven [18], [19], [20], [5] και Freenet [21], [7], [5], αλλά και ομότιμα δίκτυα για εφαρμογές, όπως το Skype [22], [23], [5] για την απευθείας επικοινωνία και το Groove [24], [7] για την κατανεμημένη συνεργασία. Ένδειξη της μαζικής αποδοχής των P2P τεχνολογιών είναι το γεγονός ότι ένα ασύλληπτο ποσοστό της κίνησης στο Internet προέρχεται από τέτοια δίκτυα. Πιο συγκεκριμένα το Μάρτιο του 2000 μια έρευνα του πανεπιστημίου του Wisconsin έδειξε ότι η εφαρμογή Napster κατανάλωνε περισσότερο εύρος σύνδεσης από ότι η κίνηση HTTP [25]. Δύο χρόνια μετά, έρευνα του πανεπιστημίου της Washington κατέληξε ότι οι

εφαρμογές ομότιμου διαμοιρασμού αρχείων κατανάλισκαν το 43% του συνολικού εύρους έναντι 14% της κίνησης WWW [26]. Είναι, τώρα πλέον, κοινό μυστικό ότι τα ομότιμα δίκτυα και δη τα δίκτυα διαμοιρασμού αρχείων διακινούν περισσότερα δεδομένα από κάθε άλλη πηγή κίνησης στο Internet [27].

1.3 Ορισμός Ομότιμων Δικτύων

Ένα από τα πρώτα προβλήματα που αντιμετωπίζει κανείς μελετώντας τα ομότιμα δίκτυα είναι η εξεύρεση ενός ακριβούς ορισμού. Κάποιοι ορίζουν τα ομότιμα δίκτυα ως μια συλλογή ετερογενών κατανεμημένων πόρων που συνδέονται σε ένα δίκτυο [28], [29]. Άλλοι προσδιορίζουν τα ομότιμα δίκτυα ως το αντίθετο της αρχιτεκτονικής εξυπηρετούμενου-εξυπηρέτη [30], [31]. Ο Kindberg ορίζει τα ομότιμα συστήματα σαν αυτά με ανεξάρτητη διάρκεια ζωής. Ο Clay Shirky της O'Reilly and Associate λέει ότι: «P2P είναι ένα είδος εφαρμογών που εκμεταλλεύονται πόρους διαθέσιμους στις παρυφές του Internet. Επειδή η πρόσβαση σε αυτούς τους αποκεντρωμένους πόρους προϋποθέτει την λειτουργία σε περιβάλλον ασταθούς συνδεσιμότητας και απρόβλεπτων IP διευθύνσεων, οι ομότιμοι κόμβοι πρέπει να λειτουργούν έξω από το DNS και να έχουν σημαντική ή ολική αυτονομία από κεντρικούς εξυπηρέτες» [32]. Προκειμένου να μην κουράσω τον αναγνώστη με μια ανούσια παράθεση διφορούμενων και συχνά αντικρουόμενων ορισμών θα καταλήξω αποδεχόμενος τους τέσσερις ορισμούς του Rüdiger Schollmeier [33].

Μια κατανεμημένη δικτυακή αρχιτεκτονική μπορεί να καλείται ομότιμο (Peer-to-Peer, P-to-P, P2P,…) δίκτυο, εάν οι συμμετέχοντες διαμοιράζονται μέρος των υλικών (hardware) πόρων τους (επεξεργαστική ισχύς, αποθηκευτική χωρητικότητα, εύρος δικτυακής σύνδεσης,…). Αυτοί οι διαμοιραζόμενοι πόροι είναι απαραίτητοι προκειμένου να παρασχεθεί η Υπηρεσία και το περιεχόμενο που προσφέρεται από το δίκτυο (π.χ. διαμοιρασμός αρχείων ή διαμοιραζόμενοι χώροι εργασίας για συνεργασία). Είναι προσβάσιμοι από άλλους ομότιμους χρήστες, χωρίς να περνάνε από ενδιάμεσες οντότητες. Οι συμμετέχοντες σε αυτό το δίκτυο είναι ως εκ τούτου πάροχοι πόρων (Υπηρεσία και περιεχόμενο) καθώς και καταναλωτές πόρων (Υπηρεσία και περιεχόμενο).

Πίνακας 1. Ο ορισμός του ομότιμου δικτύου

Μια κατανεμημένη δικτυακή αρχιτεκτονική πρέπει να κατηγοριοποιηθεί ως Αμιγώς (Pure) ομότιμο δίκτυο, εάν καταρχάς είναι ομότιμο δίκτυο σύμφωνα με τον ορισμό του ομότιμου δικτύου και κατά δεύτερον εάν οποιαδήποτε, τυχαία εκλεγμένη Τερματική Οντότητα μπορεί να αφαιρεθεί από το δίκτυο χωρίς το δίκτυο να υποφέρει την απώλεια ουδεμίας δικτυακής υπηρεσία.

Πίνακας 2. Ο ορισμός του Αμιγώς ομότιμου δικτύου

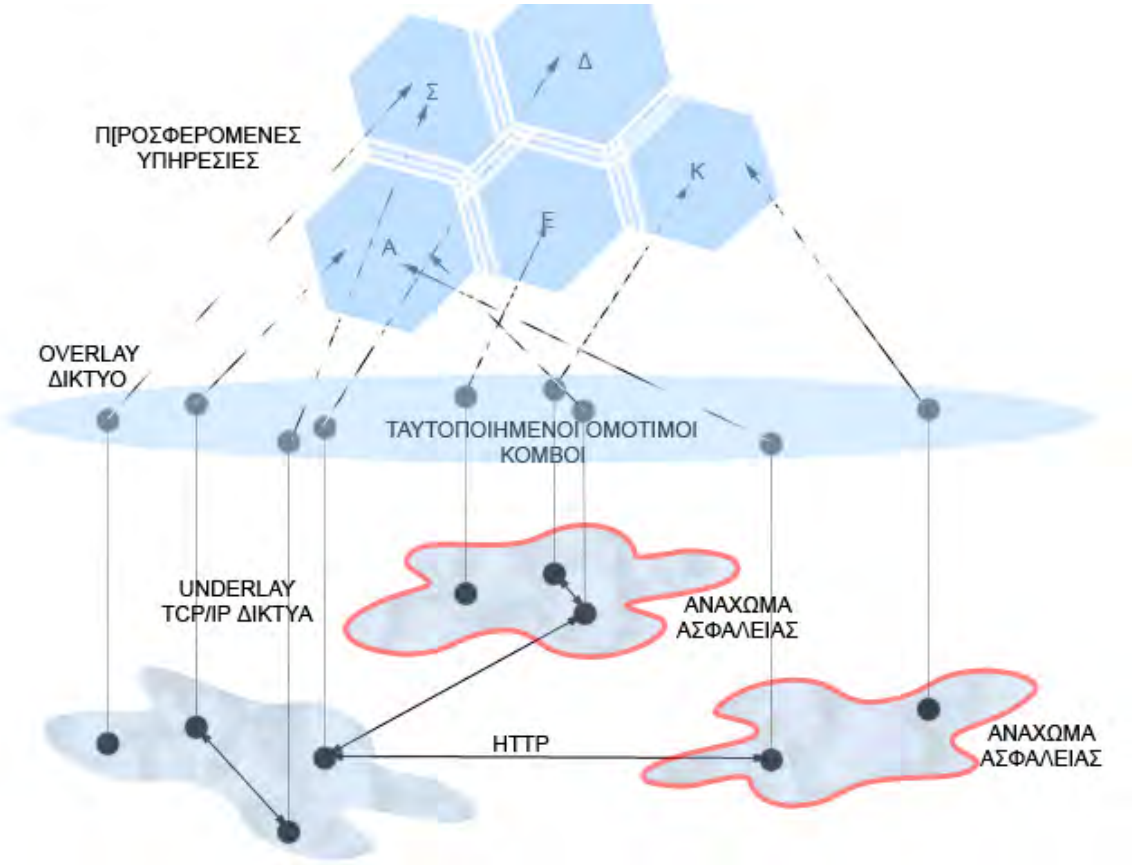
Μια κατανεμημένη δικτυακή αρχιτεκτονική πρέπει να κατηγοριοποιηθεί ως Υβριδικό ομότιμο δίκτυο, εάν καταρχάς είναι ομότιμο δίκτυο σύμφωνα με τον ορισμό του ομότιμου δικτύου και κατά δεύτερον μια κεντρική οντότητα είναι απαραίτητη προκειμένου να παράσχει μέρος των προσφερόμενων δικτυακών υπηρεσιών.

Πίνακας 3. Ο ορισμός του Υβριδικού ομότιμου δικτύου

Ένα δίκτυο εξυπηρετούμενου-εξυπηρέτη είναι ένα κατανεμημένο δίκτυο που αποτελείται από ένα σύστημα υψηλής επίδοσης, τον *εξυπηρέτη*, και αρκετά κυρίως χαμηλότερης επίδοσης συστήματα, τους *εξυπηρετούμενους*. Ο *εξυπηρέτη*ς είναι η κεντρική μονάδα καταχώρησης (central registering unit) καθώς και ο μόνος πάροχος περιεχομένου και υπηρεσίας. Ο *εξυπηρετούμενος* μόνο αιτείται περιεχόμενο ή την εκτέλεση υπηρεσιών, δίχως να μοιράζεται κανέναν από τους δικούς του πόρους.

Πίνακας 4. Ο ορισμός του μοντέλου εξυπηρετούμενου-εξυπηρέτη

Ένας άλλος τρόπος να θεωρήσουμε τα ομότιμα δίκτυα είναι ως ένα overlay επιπέδου εφαρμογής πάνω από το οποίο παρέχονται υπηρεσίες και περιεχόμενο. Το overlay αυτό χτίζεται πάνω από ένα υπάρχων δίκτυο, που συνήθως είναι το Internet, και αξιοποιεί τα πρωτόκολλα επιπέδου μεταφοράς TCP και UDP και το πρωτόκολλο επιπέδου Internet IP. Τα ομότιμα δίκτυα καθιστούν διαφανή, ως προς το χρήστη, την υπάρχουσα δικτυακή υποδομή. Οι χρήστες δε γνωρίζουν τίποτα ούτε για το υλικό (δρομολογητές, επίγεια/υποθαλάσσια καλωδίωση, δορυφορικές συνδέσεις, hopping frequencies, …), ούτε και για τα πρωτόκολλα των χαμηλότερων επιπέδων. Αντιλαμβάνονται μόνο ένα συνεχές και ενιαίο δίκτυο, ή τουλάχιστον το μέρος αυτού που τους επιτρέπει η ορατότητα, που απαρτίζεται από πόρους ομότιμων χρηστών, πάνω στο οποίο τρέχουν εφαρμογές (διαμοιρασμός αρχείων, κατανεμημένη επεξεργασία, …) και το οποίο προσφέρει υπηρεσίες απαραίτητες για τις εφαρμογές αυτές (αναζήτηση, σύνδεση, …).

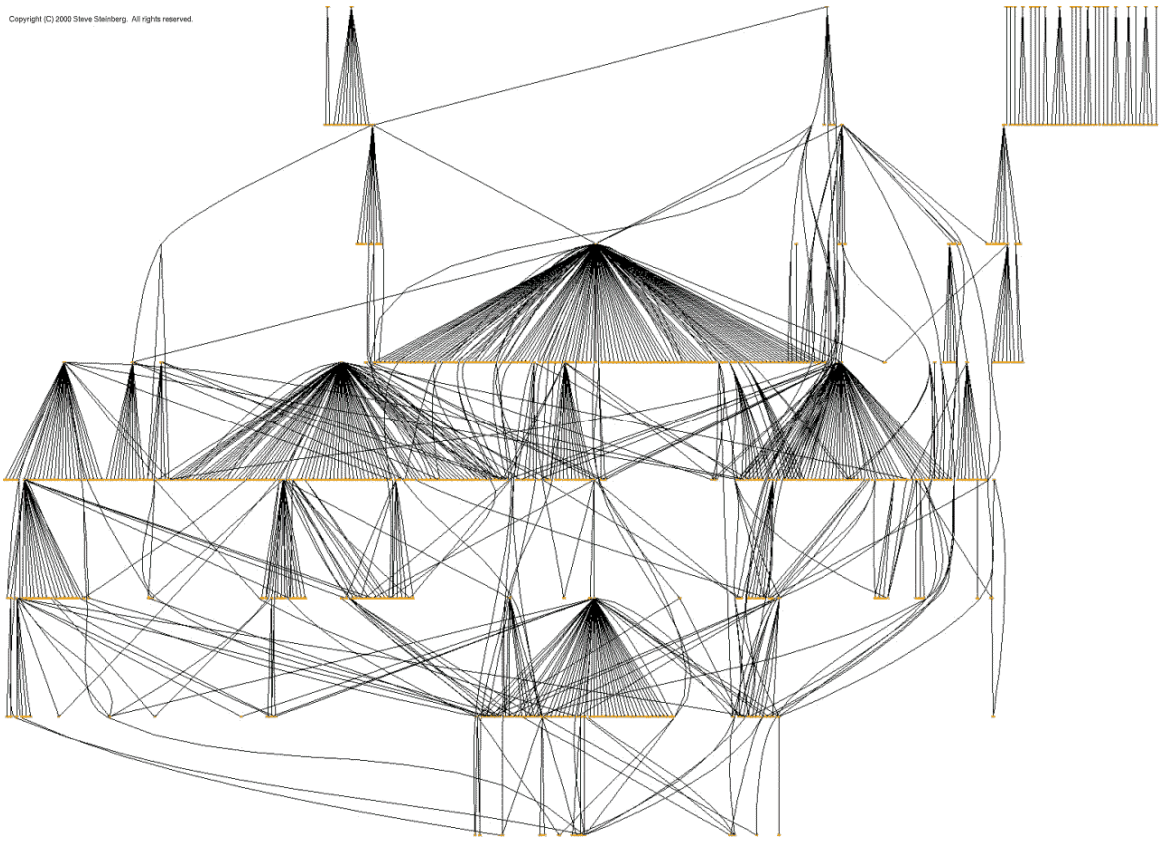


Εικόνα 1. Διαστρωμάτωση Ομότιμου Δικτύου

Αξίζει να σημειωθεί ότι πολλά ομότιμα δίκτυα δεν εκμεταλλεύονται πλήρως την υφιστάμενη υποδομή, παραδείγματος χάρη το αμιγές ομότιμο δίκτυο διαμοιρασμού αρχείων Gnutella συμπεριφέρεται στη κίνηση TCP ωσάν να ήταν αναξιόπιστη, την στιγμή που το TCP είναι ένα πρωτόκολλο προσανατολισμένο στη σύνδεση που παρέχει μηχανισμούς ανταλλαγής δεδομένων με αξιοπιστία (αυτό γίνεται επειδή το Gnutella επιτρέπει στους κόμβους του να πετάνε πακέτα εάν δε μπορούν να χειριστούν την κίνηση). Και φυσικά δεν γίνεται καν λόγος για υποστήριξη του IPsec και άλλων προηγμένων υπηρεσιών.

Επίσης μπορούμε να ταξινομήσουμε τα ομότιμα δίκτυα ανάλογα με το είδος της δρομολόγησης που υλοποιούν σε δομημένα (structured) και αδόμητα (unstructured) [34]. Οι αρχικές υλοποιήσεις του Gnutella για παράδειγμα ήταν αδόμητες και οι αναζητήσεις με λέξεις κλειδιά στέλνονταν σε όλους τους χρήστες (broadcast). Ενώ ως δομημένα συστήματα χαρακτηρίζονται οι κατεξοχήν ακαδημαϊκές υλοποιήσεις δικτύων, όπως τα CAN [35], Chord [36], Pastry [37] και Tapestry [38]. Οι δομημένες overlays επιτρέπουν στις εφαρμογές να βρίσκουν οποιοδήποτε αντικείμενο με ένα πιθανοτικά φραγμένο, μικρό πλήθος βημάτων (network hops), ενώ την ίδια στιγμή απαιτούν ανά κόμβο πίνακες

δρομολόγησης με μικρό αριθμό εισαγωγών. Επιπλέον τα συστήματα αυτά κλιμακώνονται, είναι ανθεκτικά στα σφάλματα και προσφέρουν αποτελεσματικό καταμερισμό φόρτου. Συνεπώς παρέχουν μια ισχυρή πλατφόρμα για την παροχή ποικίλων κατανεμημένων υπηρεσιών, όπως η κατανεμημένη αποθήκευση και ο διαμοιρασμός περιεχομένου. Το μειονέκτημα των δομημένων συστημάτων είναι η αναζήτησή τους, η οποία βασίζεται σε κατανεμημένους πίνακες συνόψεων (Distributed Hash Tables – DHT) [39], [40], [41], [42], [43]. Αν και η αναζήτηση μέσω DHT είναι κατάλληλη για συστήματα προσανατολισμένα στη διαθεσιμότητα, καθότι εγγυώνται την εύρεση του περιεχομένου εάν αυτό υπάρχει, με πεπερασμένο αριθμό hops, ωστόσο για να λειτουργήσει χρειάζεται τον έλεγχο της τοποθέτησης των δεδομένων στους κόμβους και της τοπολογίας του δικτύου και για να κάνει τα πράγματα ακόμα πιο απογοητευτικά υποστηρίζει μόνο την αναζήτηση με ταυτοποιητές. Αντίθετα άλλοι μηχανισμοί, όπως αυτοί του Gnutella (βλ. Εικόνα 2), έχουν σχεδιαστεί για εφαρμογές με πλουσιότερη αναζήτηση και παρέχουν μεγαλύτερη αυτονομία στους χρήστες.



Εικόνα 2. Απεικόνιση Αμιγούς Ομότιμου Δικτύου (Gnutella) [44]

Πρέπει να σημειώσω σε αυτό το σημείο ότι τα περισσότερα σύγχρονα ομότιμα δίκτυα, ακόμα και αυτό το Gnutella που ξεκίνησε ως αδόμητο, έχουν αρχίσει να

αναπτύσσουν κάποια δομή. Πολλά προϋποθέτουν την ύπαρξη χρηστών που λειτουργούν ως υπερκόμβοι ή κέντρα (hubs) και που παρέχουν υπηρεσίες στους υπόλοιπους χρήστες. Για παράδειγμα σε μια δενδροειδή τοπολογία με κέντρα στις ρίζες και απλούς κόμβους στα φύλλα, τα κέντρα θα μπορούσαν να κρατάνε πίνακες συνόψεων με τα διαμοιραζόμενα αρχεία των φύλλων τους, ώστε να επιταχύνονται οι αναζητήσεις στο δίκτυο. Ένας κόμβος ρωτάει τα γειτονικά του φύλλα για ένα αρχείο και εάν δε το βρει ζητάει από το κέντρο του να ψάξει το υπόλοιπα δίκτυο. Ως εκ τούτου η διάκριση ανάμεσα σε δομημένα και μη δομημένα θεωρείται πλέον αν όχι ανακριβής τουλάχιστον ομιχλώδης.

Είδαμε ότι τα ομότιμα δίκτυα διαχωρίζονται σε Αμιγή και Υβριδικά, για να μπορέσουν να μελετηθούν πιο εύκολα, από πλευράς απαιτήσεων ασφαλείας θα προχωρήσω σε μια άλλη διάκριση ανάλογα με το είδος της εφαρμογής που προσφέρεται πάνω από αυτά. Οι πιο διαδεδομένες εφαρμογές είναι ο διαμοιρασμός αρχείων (file-sharing), η κατανεμημένη επεξεργασία (distributed computation), η άμεση επικοινωνία (instant communication), η κατανεμημένη συνεργασία (distributed collaboration) και η κατανεμημένη αποθήκευση (distributed storage).

Ο διαμοιρασμός αρχείων (file-sharing) είναι και η πλέον δημοφιλής υπηρεσία με εκατομμύρια χρήστες κάθε στιγμή να ανταλλάσουν terabytes δεδομένων. Τα σημαντικότερα δίκτυα είναι με σειρά μεγέθους το Bittorrent, το eDonkey2000 και το Fasttrack. Ωστόσο υπάρχει ένα μεγάλο πλήθος διαφορετικών ομότιμων δικτύων και ακόμα περισσότερα λογισμικά πελάτη, τα οποία πολλές φορές μπορούν και συνδέονται σε παραπάνω από ένα δίκτυα. Στη στοιχειωδέστερη της μορφή, η εφαρμογή του διαμοιρασμού αρχείων είναι ένα σύνολο υπηρεσιών (σύνδεση, αποσύνδεση, ταυτοποίηση, αναζήτηση, ταυτόχρονο κατέβασμα αρχείων, μηχανισμοί υπόληψης, …) που επιτρέπουν σε ένα χρήστη να επιλέγει κάποια αρχεία, που βρίσκονται αποθηκευμένα στον υπολογιστή του, και να προσφέρει πρόσβαση ανάγνωσης (read access/authorization) σε αυτά σε ομότιμους χρήστες, οι οποίοι βρίσκονται στο ίδιο δίκτυο με αυτόν. Ανάλογα με το δίκτυο δίνονται διαφορετικές υπηρεσίες και καλύπτονται διαφορετικές ανάγκες (ταχύτητα κατεβάσματος, ανωνυμία, διαθεσιμότητα περιεχομένου). Αυτή καθαυτή η ανταλλαγή δεδομένων γίνεται με άμεσες συνδέσεις από χρήστη σε χρήστη, αν και υπάρχουν λογισμικά χρήστη που επιτρέπουν το ταυτόχρονο κατέβασμα και από web εξυπηρέτες και από χρήστες.

Η κατανεμημένη επεξεργασία (distributed computation) ή αλλιώς υπολογιστική κοινών πόρων (public-resource computing), επίσης γνωστή και σαν παγκόσμια υπολογιστική (global computing) και ομότιμη υπολογιστική (Peer-to-Peer computing), είναι μια ακόμα εφαρμογή που μπορεί να εξυπηρετηθεί από ένα ομότιμο σύστημα. Το

«ομότιμο» κομμάτι του συστήματος κατανεμημένης επεξεργασίας αφορά την χρήση των πόρων των χρηστών (επεξεργαστική ισχύς, αποθηκευτικός χώρος, εύρος σύνδεσης, …), όλο το υπόλοιπο σύστημα είναι μορφής εξυπηρετούμενου-εξυπηρέτη. Ένας ή περισσότεροι εξυπηρέτες τροφοδοτούνε τους χρήστες με δεδομένα προς επεξεργασία (επίσης γνωστά και ως μονάδες δεδομένων (data blocks) ή μονάδων εργασίας (work units)), οι κόμβοι των χρηστών κάνουν τους απαραίτητους υπολογισμούς και ύστερα τα αποτελέσματα συλλέγονται από τους εξυπηρέτες, οι οποίοι με διάφορους μηχανισμούς ξεχωρίζουν τα ορθά από τα εσφαλμένα αποτελέσματα και πράττουν ανάλογα. Οι ομότιμοι κόμβοι δεν επικοινωνούν μεταξύ τους και για πολλά συστήματα η συνεχής σύνδεση στο δίκτυο δεν είναι απαραίτητη, αρκεί να υπάρχει κατά τη λήψη και την αποστολή των δεδομένων, δεν είναι καν απαραίτητο ο κόμβος να είναι ενεργός συνέχεια, οι υπολογισμοί μπορούν να γίνουν κατά στάδια, μέσα σε λογικά χρονικά πλαίσια. Πρέπει να επισημανθεί ότι δεν είναι όλα τα προβλήματα κατάλληλα για να επιλυθούν με την υπολογιστική κοινών πόρων. Κατάλληλα θεωρούνται οι προσομοιώσεις φυσικών ή άλλων συστημάτων, οι λύσεις με γενετικούς αλγόριθμους και γενικά τα προβλήματα στα οποία χρειάζεται να αναλυθούν μεγάλες ποσότητες δεδομένων. Σε αντιδιαστολή με την ομότιμη υπολογιστική, η Grid υπολογιστική προϋποθέτει ιδιόκτητους πόρους, με κεντρική διαχείριση, ενεργούς συνέχεια και συνδεδεμένους συνέχεια με συνδέσεις μεγάλου εύρους. Στα Grids υπάρχει μια συμμετρία και μια οργάνωση που λείπει από τα ομότιμα συστήματα.

Μια τρίτη εφαρμογή που μπορεί να αξιοποιήσει τα ομότιμα δίκτυα είναι η άμεση επικοινωνία (instant communication). Είτε αναφερόμαστε σε συνομιλίες φωνής (Voice Over IP - VOIP) ή άμεσου κειμένου (instant messaging - chat) τα ομότιμα δίκτυα επιτρέπουν την άμεση επικοινωνία ανάμεσα σε δύο ή περισσότερους χρήστες, ενώ στις περιπτώσεις που η άμεση σύνδεση είναι ανέφικτη, λόγω αναχωμάτων ασφαλείας ή NATs, η επικοινωνία δρομολογείται μέσα από άλλους ομότιμους κόμβους. Συνήθως υπάρχουν κεντρικοί εξυπηρέτες για να παρέχουν υπηρεσίες αυθεντικοποίησης χρηστών και άλλες παρόμοιες. Πέρα από τις υπηρεσίες σύγχρονης επικοινωνίας, προσφέρονται και άλλες υπηρεσίες πρόσθετης αξίας, όπως η ανταλλαγή αρχείων ή η ασύγχρονη επικοινωνία.

Πάνω από ένα ομότιμο δίκτυο μπορεί να τρέξει και μια εφαρμογή κατανεμημένης συνεργασίας (distributed collaboration). Τέτοιες εφαρμογές στη γενική περίπτωση δημιουργούν διαμοιραζόμενους χώρους μέσα στους οποίους εξουσιοδοτημένοι χρήστες μπορούν να επικοινωνούν, να ανταλλάσουν δεδομένα και να χρησιμοποιούν ταυτόχρονα μια εφαρμογή. Οι επικοινωνίες και πάλι είναι σύγχρονες και άμεσες, αλλά επίσης μπορεί να παρέχονται και υπηρεσίες ασύγχρονης επικοινωνίας. Στη κατανεμημένη συνεργασία ένα βασικό στοιχείο είναι η εξουσιοδότηση ενός χρήστη να συμμετάσχει στο

διαμοιραζόμενο χώρο, συνέπεια αυτής της ανάγκης είναι η ύπαρξη μηχανισμών πρόσκλησης (inviting) ενός χρήστη να συμμετάσχει καθώς και απομάκρυνσης (uninviting) του.

Μια τελευταία εφαρμογή που θα αναφέρω είναι η κατανεμημένη αποθήκευση (distributed storage). Οι ειδικοί οραματίζονται ένα μέλλον πανταχού παρούσας υπολογιστικής (ubiquitous computing), αλλά προκειμένου να λάβει μέρος ένας υπολογισμός θα πρέπει από κάπου να έρθουν δεδομένα και ύστερα τα αποτελέσματα του υπολογισμού να αποθηκευτούν κάπου, γεννιέται λοιπόν το ερώτημα πού εδρεύει αυτή η διαρκής και συνεχής (persistent) πληροφορία. Τα ομότιμα συστήματα κατανεμημένης αποθήκευσης καλούνται να δώσουν μια κομψή απάντηση σε αυτό το ερώτημα με τη δημιουργία μιας διαφανούς υποδομής διαχείρισης των δεδομένων.

1.4 Ασφάλεια υπό το Πρίσμα των Ομότιμων Δικτύων

Οι αριθμοί αποδεικνύουν περίτρανα την επίδραση της τεχνολογίας των ομότιμων δικτύων (peer-to-peer technology) στην ηλεκτρονική μας καθημερινότητα, γίνεται έτσι εμφανές το γιατί η ασφάλεια (security) είναι μια σημαντική πλευρά αυτής. Μια ανασφαλής P2P τεχνολογία δεν θα μπορέσει να πείσει τις επιχειρήσεις να την ενσωματώσουν στις δοσοληψίες τους και κατά συνέπεια ο όρος P2P θα παραμείνει συνώνυμο της παράνομης ανταλλαγής ξένης πνευματικής ιδιοκτησίας.

Ένας από τα κυριότερα ζητήματα που αντιμετωπίζουν οι ειδικοί ασφάλειας είναι ότι κάθε ξεχωριστή εφαρμογή ή υπηρεσία και κάθε διαφορετικό άτομο θέτει τις δικές του απαιτήσεις ασφάλειας. Παραδείγματος χάρη, στις εφαρμογές ανταλλαγής αρχείων πάνω από ομότιμα δίκτυα η ανάγκη για ανωνυμία (anonymity) συχνά υπερέχει αυτής για εμπιστευτικότητα (confidentiality), καθότι άπαξ ο χρήστης καταστεί ανώνυμος η ιδιωτικότητά (privacy) του κατοχυρώνεται. Ενώ όσον αφορά τα ανταλλασσόμενα αρχεία, συνήθως αυτά δεν είναι εμπιστευτικά ή απόρρητα (confidential, classified) οπότε και δεν συντρέχει λόγος για την κρυπτογράφησή τους προκειμένου να προστατευτούν από αδιάκριτα μάτια. Αντίθετα στις εφαρμογές άμεσης επικοινωνίας η ιδιωτικότητα (privacy) επιτυγχάνεται με τη κρυπτογράφηση της επικοινωνίας, δηλαδή προστατεύοντας την εμπιστευτικότητα των δεδομένων (data confidentiality), και τις περισσότερες φορές δεν ζητείται ανωνυμία, τουναντίον κάποιοι χρήστες μπορεί να θέλουν αυθεντικοποίηση χρηστών (user authentication) και μη αποποίηση (non-repudiation) της επικοινωνίας. Επίσης οι ανάγκες ασφάλειας διαφέρουν από χρήστη σε χρήστη. Ένας παρανοϊκός χρήστης θα μπορούσε να εμμένει στη παροχή ανωνυμίας, αυθεντικοποίησης,

εμπιστευτικότητας και μη αποποίησης (ίσως με χρήση ψηφιακών πιστοποιητικών και ψευδωνύμων;) προκειμένου να εκκινήσει μια ομότιμη σύνοδο γραπτής συνομιλίας (peer-to-peer chatting session) με τον φίλο του που βρίσκεται στην άλλη άκρη της αίθουσας υπολογιστών. Και φυσικά υπάρχουν χρήστες που θέλουν να μοιραστούν σπιτικές συνταγές για κουλουράκια με τον υπόλοιπο κόσμο, οι οποίοι δεν ενδιαφέρονται ούτε για την ιδιωτικότητα τους, ούτε για τη μυστικότητα (secrecy) των συνταγών.

Το πρώτο που πρέπει να γίνει είναι να αναγνωριστούν οι ανάγκες ασφάλειας και οι τρόποι με τους οποίους θα καλυφθούν. Ύστερα θα πρέπει να υπολογιστεί το κόστος (cost) τους. Το κόστος δεν είναι ανάγκη να είναι χρηματικό, ένα αντίμετρο ασφάλειας (security countermeasure) μπορεί να εισαγάγει μια απαράδεκτη καθυστέρηση στην επικοινωνία, να επιβαρύνει με υπολογιστικό φόρτο το σύστημα ή απλώς να αυξήσει τη πολυπλοκότητα του ομότιμου συστήματος σε σημείο που να αποτρέπει τους χρήστες. Ακολουθεί η αξιολόγηση της αποτελεσματικότητά τους ως προς το κόστος εφαρμογής τους, συμβιβαζόμαστε ανάμεσα στο πόση ασφάλεια θέλουμε και πόσα είμαστε διατεθειμένοι να πληρώσουμε για να την αποκτήσουμε. Και τέλος υπάρχει η ίδια η υλοποίηση και ενσωμάτωση των αντιμέτρων στα ομότιμα συστήματα, καθώς και η εκπαίδευση των χρηστών στη χρήση αυτών. Όπως έχω ήδη αναφέρει η διπλωματική αναφέρεται μόνο στο πρώτο μέρος αυτής της διαδικασίας

Προκειμένου να διευκολυνθεί η μελέτη των χαρακτηριστικών ασφάλειας είναι απαραίτητο να διευκρινιστεί ο τρόπος χρήσης κάποιων ορισμών από το πεδίο της ασφάλειας των πληροφοριακών και επικοινωνιακών συστημάτων και από το πεδίο της κρυπτογραφίας [45], [46], [47], [48], [49].

Διαθεσιμότητα (Availability), Επιβιωσιμότητα (Survivability) και Αξιοπιστία (Reliability)

Διαθεσιμότητα ονομάζεται η ιδιότητα ενός συστήματος ή ενός πόρου συστήματος, συμπεριλαμβανομένων και των πληροφοριών, να είναι προσβάσιμο και έτοιμο προς χρήση όποτε το επιθυμήσει μια εξουσιοδοτημένη οντότητα. Κακόβουλοι χρήστες μπορούν να μειώσουν την αξιοπιστία ενός συστήματος με το να καταστήσουν δεδομένα ή πόρους μη διαθέσιμους, καθυστερώντας ή και εμποδίζοντας την πρόσβαση των εξουσιοδοτημένων χρηστών σε αυτά. *Επιβιωσιμότητα* είναι η ικανότητα ενός συστήματος να παραμένει εν λειτουργία ή να συνεχίσει να υφίσταται παρά τις όποιες αντίξοες συνθήκες, συμπεριλαμβανομένων των φυσικών συμβάντων, ατυχημάτων και επιθέσεων στο σύστημα. Η *αξιοπιστία* αφορά την ικανότητα ενός συστήματος να εκτελεί απαραίτητες λειτουργίες κάτω από ορισμένες συνθήκες για καθορισμένη χρονική περίοδο.

Ιδιωτικότητα (Privacy)

Το δικαίωμα μιας οντότητας (συνήθως ενός ατόμου), που δρα για λογαριασμό της, να καθορίζει το βαθμό στον οποίο θα αλληλεπιδρά με το περιβάλλον της, συμπεριλαμβανομένου του βαθμού στον οποίο η οντότητα θέλει να μοιράζεται πληροφορία σχετική με τον εαυτό της με τρίτους. Η *ιδιωτικότητα* δεν είναι ένας μηχανισμός ασφαλείας αλλά μάλλον ένας από τους λόγους για τους οποίους ζητάμε ασφάλεια. Ένας μηχανισμός που προστατεύει την *ιδιωτικότητα* είναι η *ανωνυμία* (*anonymity*), δηλαδή η κατάσταση κατά την οποία το όνομα μιας οντότητας παραμένει άγνωστο ή κρυφό για όλους ή μόνο για μη εξουσιοδοτημένες οντότητες. Για να αποκρυφτεί το όνομα μιας οντότητας μπορεί να χρησιμοποιηθεί ένα *ψευδώνυμο* (*alias*).

Εμπιστευτικότητα (Confidentiality).

Ως *εμπιστευτικότητα δεδομένων* ορίζουμε την μη αποκάλυψη, την απόκρυψη πληροφοριών και δεδομένων προς μη εξουσιοδοτημένες οντότητες. Τα δεδομένα, ανάλογα με την εφαρμογή, μπορεί να είναι διαμοιραζόμενα αρχεία, συνομιλία (φωνή ή κείμενο), ερωτήματα, αποτελέσματα και οτιδήποτε άλλο έχει αρκετή αξία για τα επικοινωνούντα μέρη ώστε να δικαιολογείται το κόστος της προστασίας τους. Η εμπιστευτικότητα εφαρμόζεται και στην ύπαρξη των δεδομένων, σε περιπτώσεις όπου η απλή γνώση της ύπαρξης προσφέρει πληροφορία που θα έπρεπε να προστατεύεται. Τέλος μπορεί να ζητηθεί και εμπιστευτικότητα στους πόρους, ήτοι *απόκρυψη πόρων*.

Η εμπιστευτικότητα επιτυγχάνεται με τη χρήση μηχανισμών *ελέγχου πρόσβασης* (*access control*). Ένας τέτοιος μηχανισμός είναι η *κρυπτογραφία* (*cryptography*), που μετασχηματίζει τα δεδομένα ώστε να τα καταστήσει ακατανόητα. Ένα *κρυπτογραφικό κλειδί* (*cryptographic key*) ελέγχει την πρόσβαση στα ακατανόητα δεδομένα. Ακόμα μπορεί το ίδιο το σύστημα να παρέχει μηχανισμούς που να αποτρέπουν την μη εξουσιοδοτημένη πρόσβαση στην πληροφορία. Τέτοιοι μηχανισμοί δύνανται να προστατεύσουν τη *μυστικότητα* (*secrecy*) των δεδομένων (ύπαρξη και περιεχόμενο) καθώς και να αποκρύψουν πολύτιμους πόρους, έχουν, όμως, το μειονέκτημα ότι εάν αποτύχουν η πληροφορία ή οι πόροι γίνονται ορατοί.

Ακεραιότητα (Integrity)

Η *ακεραιότητα* αναφέρεται στο πόσο *αξιόπιστα* (*trustworthy*) και *ακριβή* (*accurate*) είναι τα δεδομένα ή οι πόροι και συνήθως εξαντλείται στη αποφυγή μη ορθών (improper) και μη εξουσιοδοτημένων αλλαγών. Ο όρος *ακεραιότητα* περιλαμβάνει την προστασία, από αλλαγές, του περιεχομένου της πληροφορίας, γνωστή ως *ακεραιότητα δεδομένων* (*data integrity*) και την πιστοποίηση της προέλευσης των δεδομένων, γνωστή και ως

αυθεντικοποίηση προέλευσης δεδομένων (*data origin authentication*). Η πηγή της πληροφορίας είναι υπεύθυνη/εγγυάται για τη *ακρίβεια* (*accuracy*) και την *αξιοπιστία* (*credibility*) του περιεχομένου της πληροφορίας.

Υπάρχουν δύο κατηγορίες μηχανισμών που μας επιτρέπουν να διασφαλίσουμε την *ακεραιτότητα*. Η πρώτη κατηγορία περιλαμβάνει τους μηχανισμούς *πρόληψης* (*prevention*), οι οποίοι προστατεύουν την ακεραιότητα με το να εμποδίζουν τις μη εξουσιοδοτημένες προσπάθειες αλλαγής των δεδομένων ή τις μη ορθές αλλαγές αυτών. Οι μη εξουσιοδοτημένες προσπάθειες αλλαγής των δεδομένων συνήθως προστατεύονται μέσω της *αυθεντικοποίησης προέλευσης δεδομένων* και του *ελέγχου πρόσβασης*, ενώ οι μη ορθές είναι πιο δύσκολο να αποφευχθούν. Η δεύτερη κατηγορία αφορά μηχανισμούς εντοπισμού (detection). Τέτοιοι μηχανισμοί δεν εμποδίζουν την παραβίαση της ακεραιότητας αλλά ενημερώνουν τον χρήστη για την *αξιοπιστία* των δεδομένων.

Αυθεντικοποίηση (Authentication)

Ως *αυθεντικοποίηση* αναφέρεται η διαδικασία *επαλήθευσης* (*verification*) της ταυτότητας που μια οντότητα ισχυρίζεται ότι έχει ή της ανήκει. Ξεχωρίζουμε δύο είδη αυθεντικοποίησης, την *αυθεντικοποίηση προέλευσης δεδομένων* (βλ. ακεραιότητα) και την *αυθεντικοποίηση ομότιμης οντότητας* (*peer entity authentication*). Η δεύτερη αφορά την επιβεβαίωση ότι μια ομότιμη οντότητα σε μια σχέση είναι αυτή που ισχυρίζεται.

Ταυτοποίηση (Identification)

Η πράξη ή η διαδικασία η οποία παρουσιάζει μια *ταυτότητα* (*identifier - ID*) σε ένα σύστημα, ώστε αυτό να μπορεί να αναγνωρίζει μια οντότητα του και να τη ξεχωρίζει από άλλες οντότητες.

Απόδοση Ευθυνών (Accountability)

Η ιδιότητα ενός συστήματος να που διασφαλίζει ότι οι πράξεις μιας οντότητας μπορούν να αντιστοιχηθούν μοναδικά στην οντότητα εκείνη, καθιστώντας την υπεύθυνη για τις πράξεις της.

Εμπιστοσύνη (Trust)

Το μέτρο στο οποίο κάποιος μπορεί να είναι σίγουρος (*confidence*) ότι ένα σύστημα ή μια οντότητα ανταποκρίνεται στις προδιαγραφές του, ότι πράττει αυτά που υποστηρίζει ότι κάνει και δεν πραγματοποιεί μη επιθυμητές λειτουργίες. Ένας τρόπος για να χτιστεί *εμπιστοσύνη* σε ένα σύστημα είναι η διατήρηση ενός ιστορικού των

συναλλαγών, που θα επιτρέπει να κρίνεται η αξιοπιστία μιας οντότητας με βάση τη πρότερη συμπεριφορά της, δηλαδή με βάση την *υπόληψη* που έχει.

Υπόληψη (Reputation)

Η *υπόληψη* μιας οντότητας υπολογίζεται από τη συμπεριφορά που εκείνη επέδειξε στο σύνολο των συναλλαγών στις οποίες μετείχε. Εκ πρώτης όψεως φαίνεται ότι η έννοια της *υπόληψης* είναι ανταγωνιστική ως προς την έννοια της *ιδιωτικότητας*, όμως είναι δυνατό επιτρέποντας την *ταυτοποίηση* μέσω *ψευδωνύμων* να επιτευχθεί και η διατήρηση της *ανωνυμίας* της οντότητας αλλά και ο έλεγχος της σχετικά με τις πράξεις της.

1.5 Αδυναμίες & Επιθέσεις

Τα ομότιμα δίκτυα, όπως και κάθε πολύπλοκο σύστημα, έχουν *αλωσιμότητες*, ήτοι έχουν σχεδιαστικά προβλήματα και αδυναμίες αλλά και προβλήματα στην υλοποίηση και τη λειτουργία τους. Η εκμετάλλευση αυτών των *αδυναμιών* δημιουργεί *απειλές* (*threats*), όπου ως *απειλή* ορίζεται η πιθανή παραβίαση της ασφάλειας. Με τη σειρά τους οι δράσεις και διαδικασίες που πρέπει να λάβουν μέρος για να πραγματωθεί μια *απειλή* ονομάζεται *επίθεση* (*attack*). Πρέπει να τονιστεί ότι δεν καταλήγει κάθε *απειλή* σε *επίθεση* και ότι δεν είναι όλες οι *επιθέσεις* επιτυχείς, καθώς και ότι το μέγεθος της *ζημιάς* που μπορεί να έχει μια *επίθεση* πολλές φορές είναι αρκετά ασήμαντο ώστε η ύπαρξη της *αδυναμίας* να γίνεται υποφερτή.

1.5.1 Αδυναμίες

Όπως αναφέρθηκε τα ομότιμα δίκτυα έχουν αδυναμίες· ενδεχομένως οι σχεδιαστές ενός δικτύου να μην έλαβαν υπόψη κάποια σημαντική απαίτηση/παράμετρο, στη φάση του σχεδιασμού, και αναγκάστηκαν να την προσθέσουν αργότερα με έναν μη βέλτιστο τρόπο, ή το λογισμικό πελάτη να μην ακολουθεί πίστα το πρωτόκολλο ή να το υλοποιεί λανθασμένα. Πέραν αυτών των προβλημάτων τα ομότιμα δίκτυα, ως κατανεμημένα συστήματα, υποφέρουν από βυζαντινά σφάλματα (Byzantine faults) [50] και από το πρόβλημα των βυζαντινών στρατηγών [51].

Τα βυζαντινά σφάλματα μπορεί να είναι σφάλματα στο λογισμικό, λάθη του χειριστή/χρήστη ή και κακόβουλες επιθέσεις και είναι η κυριότερη αιτία διακοπής

παροχής υπηρεσίας. Στη γενική περίπτωση είναι ένα τυχαίο σφάλμα που συμβαίνει κατά την εκτέλεση του αλγορίθμου ενός κατανεμημένου συστήματος. Συμπεριλαμβάνονται σε αυτά, τα σφάλματα αποτυχίας (crash failures) και σφάλματα αποστολής και παράλειψης (send and omission failures). Όταν συμβαίνει μια βυζαντινή αποτυχία (Byzantine failure) το σύστημα μπορεί να ανταποκριθεί με απρόβλεπτο τρόπο.

Οι τυχαίες αυτές αποτυχίες μπορούν στοιχειωδώς να κατηγοριοποιηθούν ως εξής:

- Αποτυχία στην εκτέλεση του επόμενου βήματος του αλγορίθμου, γνωστό και ως σφάλμα αποτυχίας.
- Αποτυχία στην ορθή εκτέλεση ενός βήματος του αλγορίθμου.
- Τυχαία εκτέλεση ενός βήματος, διαφορετικού από αυτό που υποδεικνύει ο αλγόριθμος.

Πιο γενικά ένα βυζαντινό σφάλμα είναι αυτό στο οποίο ένα εξάρτημα του συστήματος όχι μόνο συμπεριφέρεται εσφαλμένα, αλλά επίσης αποτυγχάνει να συμπεριφερθεί με συνέπεια στις συναλλαγές του με τα υπόλοιπα εξαρτήματα. Σε ένα σύστημα ανθεκτικό στα βυζαντινά σφάλματα (Byzantine fault tolerant – BTF) τα εξαρτήματα που λειτουργούν ορθά θα μπορέσουν να καταλήξουν στις ίδιες ομαδικές αποφάσεις, ασχέτως της συμπεριφοράς των βυζαντινά εσφαλμένων εξαρτημάτων.

Το πρόβλημα των βυζαντινών στρατηγών περιγράφει μια ομάδα στρατηγών, ο καθένας από τους οποίους έχει υπό τις διαταγές του μια μεραρχία του βυζαντινού στρατού, που πολιορκούν μια πόλη. Οι στρατηγοί επιθυμούν να αποφασίσουν σε ένα σχέδιο επίθεσης. Στην πιο απλή μορφή του, καλούνται να αποφασίσουν εάν θα επιτεθούν ή θα υποχωρήσουν. Οι γνώμες διίστανται, αλλά θα πρέπει να ληφθεί μια απόφαση κοινή για όλους. Υπάρχουν δύο απαιτήσεις όλοι οι πιστοί στρατηγοί πρέπει να ακολουθήσουν το σχέδιο που θα αποφασιστεί και ένας μικρός αριθμός προδοτών στρατηγών δε πρέπει να μπορεί να αναγκάσει τους πιστούς στρατηγούς να ακολουθήσουν έναν μη βέλτιστο σχέδιο.

Η παρουσία προδοτών στρατηγών περιπλέκει το πρόβλημα, καθώς οι προδότες δύνανται όχι μόνο να ψηφίσουν μια μη βέλτιστη απόφαση, αλλά μπορούν να το κάνουν και επιλεκτικά. Η ψήφος ενός προδότη στρατηγού δε χρειάζεται να είναι η ίδια στα μάτια των υπολοίπων. Παραδείγματος χάρη στη περίπτωση τριών στρατηγών, από τους οποίους ένας ψηφίζει υποχώρηση και ένας επίθεση, ο τρίτος μπορεί να δώσει μια ψήφο επίθεσης στον ένα και υποχώρησης στον άλλο. Αυτός που θα λάβει τη ψήφο υποχώρησης θα υποχωρήσει ενώ ο άλλος (δυστυχώς για τον ίδιο) θα επιτεθεί.

Οι Lamport, Shostak, και Pease πρότειναν διάφορες λύσεις για την επίλυση του προβλήματος.

Ένας τρόπος για να επιτευχθεί η αξιοπιστία είναι η ύπαρξη πλεοναζόντων εξαρτημάτων του συστήματος και να λαμβάνεται η *απόφαση της πλειοψηφίας*. Για να είναι επιτυχές το σύστημα θα πρέπει όλα τα μη εσφαλμένα εξαρτήματα να αποφασίζουν με βάση τις ίδιες τιμές εισόδου και επιπλέον εάν η μονάδα εισόδου δεν είναι εσφαλμένη, όλα τα μη εσφαλμένα εξαρτήματα να χρησιμοποιούν τη τιμή που έδωσε ως είσοδο. Δεν υπάρχει λύση εάν οι πιστοί στρατηγοί είναι λιγότεροι ή ίσοι των ⅔ των στρατηγών.

Μια δεύτερη λύση χρησιμοποιεί *προφορικά μη υπογεγραμμένα μηνύματα*. Υπάρχουν τρεις απαιτήσεις. Πρώτον χρειάζεται ακεραιότητα και διαθεσιμότητα στα μήνυμα, ήτοι κάθε μήνυμα που στέλνεται πρέπει να παραδίδεται σωστά. Η αποτυχία στη παράδοση ενός μηνύματος λογίζεται ως ένα επιπλέον προδοτικό εξάρτημα. Δεύτερον ο παραλήπτης ενός μηνύματος ξέρει ποιος το έστειλε (δεν υπάρχουν ψηφιακές υπογραφές). Και τρίτον η απουσία ενός μηνύματος μπορεί να εντοπιστεί. Εάν λιγότεροι από το ⅓ των στρατηγών είναι προδότες το πρόβλημα έχει λύση.

Μια τρίτη χρησιμοποιεί *υπογεγραμμένα μηνύματα*. Υπάρχουν τρεις απαιτήσεις. Πρώτον χρειάζεται ακεραιότητα και διαθεσιμότητα στα μήνυμα, ήτοι κάθε μήνυμα που στέλνεται πρέπει να παραδίδεται σωστά. Η αποτυχία στη παράδοση ενός μηνύματος λογίζεται ως ένα επιπλέον προδοτικό εξάρτημα. Δεύτερον η υπογραφή ενός στρατηγού δε μπορεί να πλαστογραφηθεί και όλοι μπορούν να επιβεβαιώσουν την αυθεντικότητά της (χρήση ψηφιακών υπογραφών και Παρόχων Πιστοποίησης). Και τρίτον η απουσία ενός μηνύματος μπορεί να εντοπιστεί. Εάν υπάρχουν τουλάχιστον δύο πιστοί στρατηγοί υπάρχει λύση.

Η δεύτερη και τρίτη λύση μπορούν να τροποποιηθούν για να ισχύουν και σε περιπτώσεις όπου δεν υπάρχει κανάλι άμεσης επικοινωνίας ανάμεσα σε κάθε στρατηγό.

1 Η υπόθεση των βυζαντινών αποτυχιών μοντελοποιεί αρκετά καλά τα ομότιμα δίκτυα, όπου κόμβοι και δίκτυο μπορούν να συμπεριφερθούν με απρόβλεπτους τρόπους εξαιτίας αστοχιών υλικού, προβλημάτων στο δίκτυο (συμφόρηση, αποσύνδεση, ...) ή και κακόβουλων επιθέσεων.

1.5.2 Επιθέσεις

Επίθεση ονομάζονται οι δράσεις που πρέπει να πραγματοποιηθούν προκειμένου να πραγματοποιηθεί μια *απειλή*. Ο Shirey [52] προτείνει τη ταξινόμηση σε τέσσερις κατηγορίες. Η πρώτη περιλαμβάνει επιθέσεις *αποκάλυψης* (*disclosure*), ή *μη εξουσιοδοτημένης πρόσβασης σε πληροφορίες* (*unauthorized access to information*), η δεύτερη τις επιθέσεις *εξαπάτησης* (*deception*) ή *αποδοχής εσφαλμένων δεδομένων* (*acceptance of false data*), η τρίτη επιθέσεις *διάλυσης* (*disruption*) ή *διακοπής*

(interruption) ή παρεμπόδισης (prevention) της ορθής λειτουργίας και η τέταρτη επιθέσεις σφετερισμού (usurpation) ή μη εξουσιοδοτημένου ελέγχου κάποιου μέρους του συστήματος. Επιπλέον, για τις επιθέσεις στα ομότιμα δίκτυα, μπορούμε να κάνουμε έναν ακόμα διαχωρισμό ανάλογα με το πλήθος των επιτιθέμενων ή με το πλήθος των ομότιμων κόμβων που ένας επιτιθέμενος έχει υπό τον έλεγχό του, σε συνεργατικές (collaborated) και μη συνεργατικές (un-collaborated).

Οι περισσότερες εξ αυτών των επιθέσεων πλήττουν εν γένει τα πληροφορικά και επικοινωνιακά συστήματα στο Internet. Η παρατήρηση αυτή οδηγεί σε δύο συμπεράσματα. Αφενός μεν για πολλά από τα προβλήματα που θα εξεταστούν υπάρχουν ήδη έτοιμες λύσεις (out of the box/shelf), αφετέρου γίνεται εμφανές ότι ο χαρακτηρισμός των ομότιμων δικτύων ως ανασφαλή εξ ορισμού είναι άδικος, αφού τους προσάπτει προβλήματα της τρέχουσας υποδομής του Internet.

Σε κάθε περίπτωση από τη στιγμή που ένας ομότιμος κόμβος μπορεί ταυτόχρονα να αποτελεί μέρος ενός άλλου, ενδεχόμενος εταιρικού, δικτύου είναι απαραίτητο να ληφθούν αντίμετρα που είτε να εξαλείφουν τους κινδύνους είτε να μετριάζουν τη ζημιά που επιφέρει μια επιτυχημένη επίθεση.

Συνεργατικές και μη συνεργατικές επιθέσεις	**Επιθέσεις Αποκάλυψης**
	Επιθέσεις ταυτοποίησης χρήστη
	IP harvesting
	Παρακολούθηση επικοινωνίας
	Ανάλυση κίνησης
	Επιθέσεις Εξαπάτησης
	Δηλητηρίαση/Νόθευση των δεδομένων
	Εισαγωγή κακόβουλου κώδικα στα δεδομένα
	Ύπαρξη κακόβουλου κώδικα στο λογισμικό του ομότιμου δικτύου
	Επανεκπομπή πρότερης επικοινωνίας
	Ενδιαμέσου
	Masquerading
	Διάδοση εσφαλμένων πληροφοριών
	Επιθέσεις Διάλυσης/Διακοπής/Παρεμπόδισης
	Εσφαλμένες πληροφορίες δρομολόγησης
	Άρνηση υπηρεσίας
	Μονόπλευρη εκμετάλλευση των πόρων του δικτύου
	Λογοκρισία
	Αποστολή μη ζητηθείσας πληροφορίας

	Υπερχείλιση αποθηκευτικού χώρου
	Επιθέσεις Σφετερισμού
	Επίθεση Sybil
	Επίθεση Eclipse

Πίνακας 5. Επιθέσεις σε περιβάλλοντα ομότιμων δικτύων

1.5.2.1 Επιθέσεις Αποκάλυψης

Πολλά δίκτυα είναι σχεδιασμένα με τέτοιο τρόπο, όχι κατ' ανάγκη εσκεμμένα, ώστε να είναι εύκολη η ανακάλυψη πληροφοριών που άπτονται της ιδιωτικότητας των χρηστών. Δεδομένης της φύσης των ομότιμων δικτύων (άμεση επικοινωνία μεταξύ ομότιμων οντοτήτων) και της μη παροχής υπηρεσιών εμπιστευτικότητας από τα περισσότερα δίκτυα, ένας επιτιθέμενος μπορεί εύκολα να συλλέξει πληροφορίες όπως διευθύνσεις IP, λίστες με διαμοιραζόμενο περιεχόμενο, κίνηση χρηστών και με βάση αυτές τις πληροφορίες να προσπαθήσει να αντιστοιχήσει την ταυτότητα δικτύου ενός χρήστη με ένα άτομο του πραγματικού κόσμου.

1.5.2.2 Επιθέσεις Εξαπάτησης

Οι επιθέσεις κατά των δεδομένων που ανταλλάσσονται μέσω ομότιμων δικτύων και κατά του λογισμικού των εξαρτημάτων αυτών δεν αποτελούν μια πρωτότυπη μέθοδο παραβίασης της ασφάλειας, αντίθετα βασίζονται σε επιτυχημένες στρατηγικές που εφαρμόζονται κατά κόρον στο Internet, στο ηλεκτρονικό ταχυδρομείο και στα τοπικά δίκτυα. Αυτές περιλαμβάνουν τη δηλητηρίαση/νόθευση των ανταλλασσόμενων δεδομένων [53]. Η δηλητηρίαση/νόθευση επιτυγχάνεται με την ενσωμάτωση ιϊκών προγραμμάτων και γενικότερα προγραμμάτων κακόβουλης λογικής (malicious logic) στα ανταλλασσόμενα δεδομένα [54], ιδιαίτερα όταν τα τελευταία είναι εκτελέσιμα προγράμματα και με την παροχή ψευδών στοιχείων για τα ανταλλασσόμενα δεδομένα. Χαρακτηριστικό παράδειγμα, στο ομότιμο δίκτυο διαμοιρασμού αρχείων Napster, οι προσπάθειες της RIAA να αποτρέψει τους χρήστες από το κατέβασμα μουσικής, διαδίδοντας αρχεία mp3 που είτε δεν ανταποκρίνονται στην περιγραφή των, είτε είχαν πολύ χαμηλή ποιότητα. Σε ομότιμα δίκτυα υπολογιστικής κοινών πόρων η επίθεση αυτή συχνά έχει τη μορφή της μη επιστροφής των αποτελεσμάτων της επεξεργασίας ή της επιστροφής εσφαλμένων αποτελεσμάτων. Ενώ στα ομότιμα δίκτυα άμεσης επικοινωνίας

είναι η ίδια συνομιλία (φωνή ή κείμενο) που παραποιείται (συνήθως επιθέσεις ενδιάμεσου – man in the middle).

Όσων αφορά το λογισμικό πελάτη (client software) κάθε ομότιμου δικτύου είναι σχετικά εύκολο για έναν κακόβουλο χρήστη να διοχετεύσει στο Internet μια τροποποιημένη έκδοση αυτού. Ο λόγος πρέπει να αναζητηθεί στη πληθώρα των εκδόσεων λογισμικών, που κυκλοφορούν για κάθε ομότιμο δίκτυο, στις συνεχείς αναβαθμίσεις που αυτά υφίστανται καθώς και στην ύπαρξη πολλών μη έμπιστων σημείων διανομής. Σε αυτή την περίπτωση μια ατελείωτη γκάμα δυνατοτήτων ανοίγεται για των επιτιθέμενο, από την υλοποίηση ενός προγράμματος που δε θα σέβεται το πρωτόκολλο του ομότιμου δικτύου με αποτέλεσμα να προκαλείται όχληση στην ομαλή ροή των πληροφοριών (βλ. επιθέσεις στη δρομολόγηση και όχλησης του δικτύου), έως την ενσωμάτωση στο πρόγραμμα κακόβουλου κώδικα που να μολύνει τον υπολογιστή εγκατάστασης και ίσως και τους υπόλοιπους υπολογιστές που ανήκουν στο ίδιο underlay δίκτυο με αυτόν. Το κακόβουλο λογισμικό, εκμεταλλευόμενο σημεία ευπάθειας του λειτουργικού συστήματος, θα μπορούσε να δημιουργήσει κενά στην περιμετρική ασφάλεια του δικτύου στο οποίο ανήκει ο υπολογιστής, προκειμένου να προβεί αργότερα σε μια σειρά ζημιογόνων πράξεων, όπως είναι η παραβίαση της ιδιωτικότητας, η υποκλοπή αριθμών πιστωτικών καρτών, συνθηματικών, ευαίσθητων εταιρικών δεδομένων (λίστες πελατών, …) ή ακόμα και εκμετάλλευση εν αγνοία του χρήστη των υπολογιστικών του πόρων (ίσως προς τέλεση παρανόμων δραστηριοτήτων;)

Μια επίθεση μπορεί δυνητικά να συνδυάζει πάνω από ένα από τα χαρακτηριστικά που αναφερθήκαν και που θα αναφερθούν στα επόμενα δύο κεφάλαια. Μια τέτοια επίθεση ήρθε στη δημοσιότητα τον Οκτώβριο του 2004, επρόκειτο περί ενός δούρειου ίππου, επανέκδοση ενός παλαιότερου κακόβουλου προγράμματος με το όνομα iosdt. Το κακόβουλο αυτό πρόγραμμα κυκλοφορούσε σε ένα ομότιμο δίκτυο διαμοιρασμού αρχείων με την ψευδή ονομασία "Sims2 Crack.exe". Όταν, όμως, εκτελούνταν εγκαθιστούσε και έτρεχε, εν αγνοία του χρήστη, το λογισμικό πελάτη του ομότιμου δικτύου κατανεμημένης επεξεργασίας distributed.net υπό έναν υπάρχον λογαριασμό χρήστη, εκμεταλλευόμενο τους πόρους του κόμβου.

Πολλά ομότιμα δίκτυα διαμοιρασμού αρχείων προσφέρουν, τώρα πια, υπηρεσίες βαθμονόμησης και σχολιασμού των διαμοιραζόμενων εγγράφων καθώς και επιπλέον πληροφόρηση σχετικά με το περιεχόμενο (χρήση μεταδεδομένων). Άλλωστε η αυθεντικοποίηση των αρχείων με τη χρήση συνόψεων, έχει επιτρέψει σε πολλούς χρήστες να αναζητούν τις συνόψεις των αρχείων μέσα από παράπλευρα κανάλια εκτός του ομότιμου δικτύου, όπως για παράδειγμα σε σελίδες του Internet από όπου μπορούν να λάβουν εκτενείς πληροφορίες για το περιεχόμενο του αρχείου. Υπάρχουν ακόμα ομότιμα δίκτυα, συνήθως άμεσης επικοινωνίας, που αυθεντικοποιούν τα δεδομένα με

χρήση ψηφιακών υπογραφών. Τέλος αρκετά δίκτυα, προκειμένου να δώσουν ποιοτικότερες υπηρεσίες ανταλλαγής αρχείων, χρησιμοποιούν τεχνικές οι οποίες ευνοούν το κατέβασμα υποκομματιών του αρχικού αρχείου για τα οποία υπάρχει σύνοψη και άρα μπορούν να αυθεντικοποιηθούν (βλ. υποκεφάλαιο 2.3. eDonkey2000 για ένα παράδειγμα τέτοιου μηχανισμού). Χάρη σε αυτούς τους μηχανισμούς οι χρήστες έχουν τη δυνατότητα προτού κατέβει ολόκληρο το αρχείο να δουν ένα μέρος αυτού (preview) και να κρίνουν αν ανταποκρίνεται στις απαιτήσεις τους.

Όσον αφορά τη περίπτωση του λογισμικού των εξαρτημάτων του ομότιμου δικτύου (λογισμικό πελάτη, λογισμικό εξυπηρέτη, …), για να αντιμετωπιστεί η διάδοση κακόβουλων εκδόσεων αυτών, οι γνωστές ομάδες ανάπτυξης συνηθίζουν να υπογράφουν ψηφιακά το κώδικά τους ή τουλάχιστον να δίνουν μέσα από τη σελίδα τους τις συνόψεις των αυθεντικών εκτελέσιμων. Αρκεί τώρα ο χρήστης να εμπιστευτεί τη πηγή του κώδικα που εκτελεί [55]. Επίσης, επειδή, συνήθως το λογισμικό διανέμεται υπό τους όρους μιας άδειας ανοικτού κώδικα, τυχών λάθη και σφάλματα εντοπίζονται έγκαιρα από τη κοινότητα των χρηστών, όσο για τις περιπτώσεις κλειστού κώδικα ας μη το συζητάμε άλλο «πίστευε· και μη ερεύνα».

Ακόμα σε δίκτυα όπου η ταυτοποίηση των χρηστών δε είναι μονοσήμαντη και καλά ορισμένη και δεν παρέχονται υπηρεσίες αυθεντικοποίησης των χρηστών, κακόβουλες οντότητες μπορούν να εξαπολήσουν τις γνωστές πλέον επιθέσεις επανεκπομπής πρότερης επικοινωνίας (replay), ενδιάμεσου (man in the middle) και masquerading.

Τέλος ένα από τα σοβαρότερα προβλήματα που αντιμετωπίζουν τα ομότιμα δίκτυα, είναι αυτό της διάδοσης ψευδών πληροφοριών. Πέρα από τα ψευδή δεδομένα που αναφερθήκαν στην αρχή, μια οντότητα κατά τη διάρκεια της συμμετοχής της σε ένα ομότιμο δίκτυο καλείται να δώσει πληροφορία και στοιχεία για την ίδια τη τοπολογία του δικτύου, την ύπαρξη και διάθεση πόρων, την υπόληψη των συμμετεχόντων, τη συμπεριφορά των υπολοίπων (σε μερικά συστήματα παρέχεται η δυνατότητα να κατηγορήσει μια ομότιμη οντότητα κάποια άλλη ότι αθέτησε το συμβόλαιο παροχής υπηρεσιών, μέσα από τις αλληλοκατηγορίες χτίζεται η υπόληψη των κόμβων). Εάν αρκετές οντότητες αποφασίσουν να δώσουν ψευδείς απαντήσεις τότε διαταράσσεται η λειτουργία του δικτύου.

1.5.2.3 Επιθέσεις Διάλυσης, Διακοπής και Παρεμπόδισης

Σε αυτή τη κατηγορία επιθέσεων βρίσκουμε πρακτικές όπως η παροχή εσφαλμένων πληροφοριών δρομολόγησης [56]. Επειδή η δρομολόγηση στο overlay

δίκτυο εξαρτάται αποκλειστικά από τους ομότιμους κόμβους είναι δυνατόν αρκούντος μεγάλο πλήθος επιτιθέμενων, δρομολογώντας εσφαλμένα τους νόμιμους χρήστες να διαμορφώσει το δίκτυο έτσι ώστε να ελέγχει τη διακίνηση των δεδομένων (βλ. επίθεση Eclipse στο υποκεφάλαιο 1.5.2.4. Επιθέσεις Σφετερισμού), ή και ακόμα να εξαπολήσει μια επίθεση κατανεμημένης άρνησης υπηρεσίας (DDOS)) [57] ως προς κάποια οντότητα του Internet, για παράδειγμα εάν μπορούσε μια οντότητα να ξεγελάσει τους ομότιμους χρήστες ώστε να πιστέψουν ότι οι πόροι ή το περιεχόμενο, που ζητάνε, παρέχονται από αυτή την οντότητα, τότε η κίνηση που θα δημιουργούταν προς την οντότητα θα την εμπόδιζε από το να παράσχει υπηρεσίες στους νόμιμους χρήστες αυτών, η ίδια τεχνική θα μπορούσε να χρησιμοποιηθεί και για την μετάδοση μη ζητηθείσας πληροφορίας [7], σε αυτή τη περίπτωση ο επιτιθέμενος αντίς της τοποθεσίας των ζητηθεισών πόρων θα επέστρεφε ένα URL προς ένα web site όπου θα διαφήμιζε το προϊόν του. Ένα άλλο σημαντικό πρόβλημα είναι η άρνηση πολλών χρηστών να προσφέρουνε πόρους στο δίκτυο [58]. Αυτό έχει ως αποτέλεσμα να μειώνεται η ποιότητα των προσφερόμενων υπηρεσιών αλλά και να καταλύεται ο αποκεντρωμένος χαρακτήρας του δικτύου καθώς οι λιγοστοί κόμβοι που διαμοιράζονται πόρους αναδύονται ως το επίκεντρο του δικτύου. Επίσης στα περισσότερα δίκτυα διαμοιρασμού αρχείων δεν παρέχονται υπηρεσίες διασφάλισης της διαθεσιμότητας του εγγράφου, σε μερικά μάλιστα, λόγω του ότι χρησιμοποιούνται κεντρική εξυπηρέτες για την εξεύρεση πόρων, είναι πολύ εύκολη η λογοκρισία του περιεχομένου. Ένας επιτιθέμενος αρκεί να βρει την πηγή και να κινηθεί εναντίον της με νομικά ή άλλα μέσα. Τέτοια ήταν η περίπτωση του Supernova [5], ενός από τα πιο γνωστά web sites που παρείχαν αρχεία .torrents, τα οποία χρησιμοποιούνται από το δίκτυο Bittorrent για το διαμοιρασμό περιεχομένου, το οποίο έκλεισε το Δεκέμβριο του 2004, λόγω πίεσης που δέχτηκε ο διαχειριστής και ιδρυτής του.

1.5.2.4 Επιθέσεις Σφετερισμού

Η επίθεση Sybil [59] αναφέρεται στη περίπτωση που μια εσφαλμένη οντότητα δύναται να παρουσιάσει πολλαπλές ταυτότητες και άρα να έχει υπό τον έλεγχο της ένα αρκετά μεγάλο κομμάτι του συστήματος, υπονομεύοντας τοιουτοτρόπως την ικανότητα πλεονασμού του.

Η επίθεση Eclipse [60] είναι πιο γενική από την επίθεση Sybil, μάλιστα ένας επιτιθέμενος μπορεί να χρησιμοποιήσει την επίθεση Sybil για να εξαπολύσει μια επίθεση Eclipse με το να δημιουργήσει ένα μεγάλο αριθμό φαινομενικά διακριτών κόμβων. Η όλη επίθεση συνίσταται στο να μπορέσει ένας ή περισσότεροι κακόβουλοι χρήστες να ελέγξουν ένα αρκετά μεγάλο πλήθος γειτόνων ορθών κόμβων· κάθε κόμβος σε ένα

ομότιμο δίκτυο διατηρεί δείκτες προς ένα σύνολο γειτονικών κόμβων προκειμένου να δημιουργηθεί το overlay και να υλοποιηθούν οι υπηρεσίες του συστήματος. Κάτω από τον κακόβουλο έλεγχο οι γείτονες αυτοί κόμβοι μπορούν να «εκλείψουν» τους υπόλοιπους ορθούς κόμβους του δικτύου καταστρέφοντας ή αναδρομολογώντας τα μηνύματα που προσπαθούν να τους φτάσουν, διαταράσσοντας έτσι την ομαλή παροχή υπηρεσιών. Στην χείριστη περίπτωση ένα επιτιθέμενος μπορεί να ελέγξει όλη τη κίνηση στο δίκτυο.

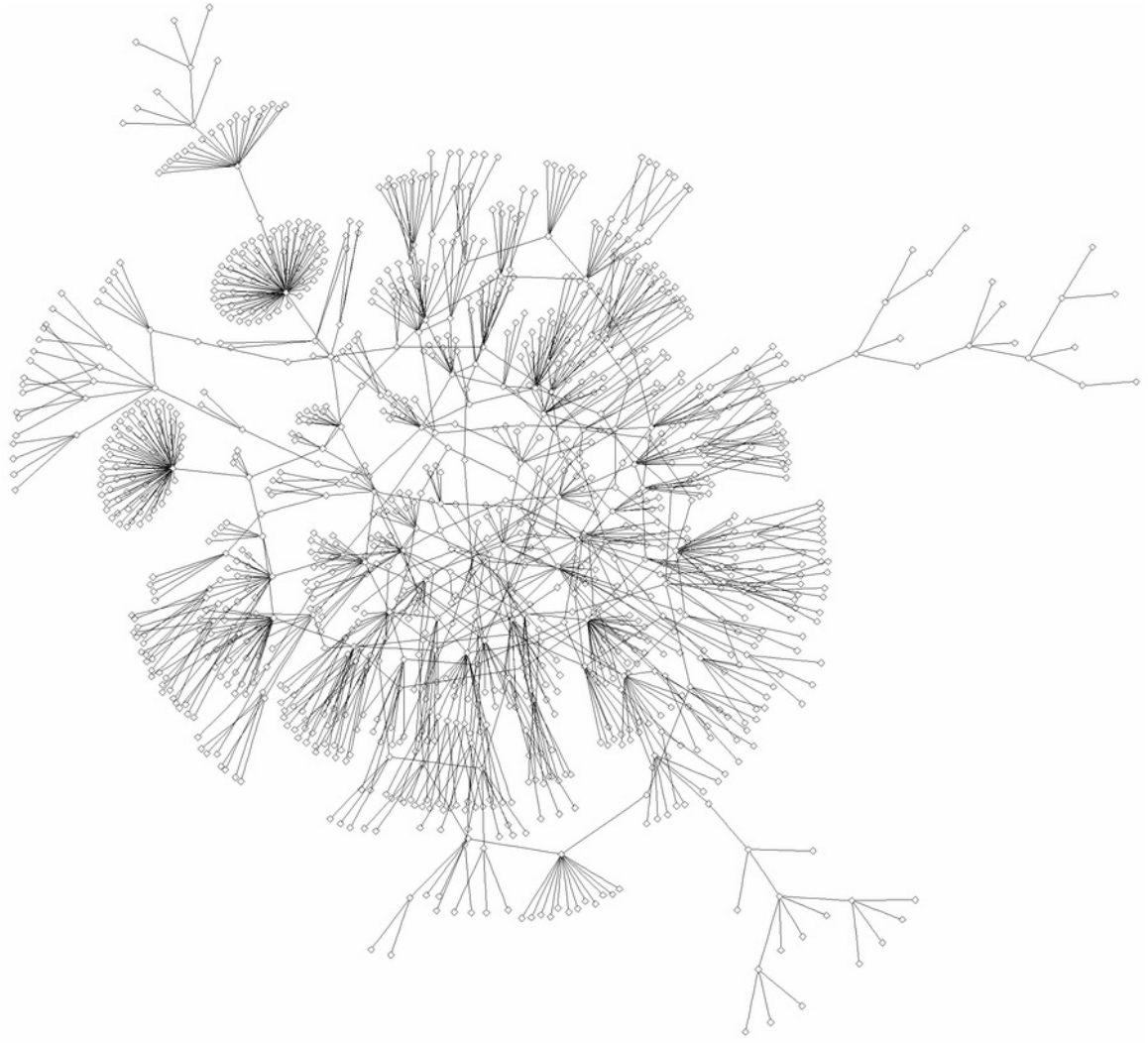
ΚΕΦΑΛΑΙΟ 2 - ΕΚΦΑΝΣΕΙΣ ΟΜΟΤΙΜΩΝ ΔΙΚΤΥΩΝ

2.1 Gnutella

Πλήρως Ομότιμο Δίκτυο Διαμοιρασμού Αρχείων

Σύντομη Περιγραφή

Το Gnutella [7], [61], [5], [8], [9], [10] είναι ένα απλό στη σύλληψη του πρωτόκολλο, που επιτρέπει σε υπολογιστές να επικοινωνούν μεταξύ τους με έναν αμιγώς απόκεντρικωποιημένο τρόπο (purely decentralized fashion). Το δίκτυο Gnutella (Gnutella Network, gNet) είναι ένα δίκτυο επιπέδου εφαρμογής (application-level network), ήτοι ένα overlay, το οποίο τρέχει πάνω από ένα TCP/IP προσανατολισμένο στη σύνδεση δίκτυο. Η υποδομή του gNet μεταβάλλεται αδιάκοπα, εφόσον οι κόμβοι που το αποτελούν συνδέονται και αποσυνδέονται κατά βούληση και για αυτό το πρωτόκολλο, αντίθετα από το Internet, δε ταυτοποιεί με έναν συνεχή (meaningful and persistent) τρόπο τους κόμβους του. Η underlying υποδομή του Internet είναι αδιαφανής στους ομότιμους χρήστες. Οι οποίοι δε χρειάζεται να απομνημονεύουν περίπλοκες διευθύνσεις προκειμένου να προσπελάσουν την πληροφορία, μια λέξη κλειδί είναι αρκετή για την αναζήτηση. Αυτό που τελικά προσφέρει το Gnutella είναι κατανεμημένο, πραγματικού χρόνου σύστημα ανάκτησης πληροφορίας (distributed, real-time information retrieval system) που προωθεί την ελεύθερη διερμήνευση και απάντηση σε ερωτήματα (free interpretation and response to queries) [7].



Εικόνα 3. Χάρτης του δικτύου Gnutella [62]

Ιστορία

Ο Justin Frankel και ο Tom Pepper της Nullsoft, ένα παράρτημα της AOL, ήταν οι εφευρέτες του Gnutella. Στις 14 Μαρτίου 2000, μια πρώιμη (beta) έκδοση του προγράμματος δόθηκε στο κοινό, ο πηγαίος κώδικας θα δημοσιευόταν αργότερα κάτω από την άδεια GNU General Public License (GPL). Την ίδια εποχή το δίκτυο Napster κατηγορούταν ότι επέτρεπε την παράνομη διάδοση πνευματικής ιδιοκτησίας. Σύμφωνα με τον Tom Pepper, to Gnutella αναπτύχθηκε με πρωταρχικό σκοπό τον διαμοιρασμό συνταγών recipes. Δυστυχώς, τα στελέχη της AOL δεν ήταν πρόθυμα να βελτιώσουν την κατάσταση του διαμοιρασμού συνταγών και διέκοψαν το πρόγραμμα. Εντός ολίγων ημερών το πρωτόκολλο έγινε reverse engineered, και συμβατά προγράμματα «κλώνοι» ανοικτού κώδικα γράφθηκαν. Η νομική καταδίκη του Napster (αρχές 2001) αύξησε τη δημοτικότητα του gNet με αποτέλεσμα να αποκαλυφθούν τα προβλήματα

επεκτασιμότητας (scalability) του δικτύου, τα οποία ανακουφίσθηκαν κάπως από παραλλαγές του αρχικού πρωτοκόλλου, αρχικά σε υλοποιήσεις κλειστού κώδικα. Λέγεται ότι παράλληλη ανάπτυξη διαφορετικών προγραμμάτων πελατών από διαφορετικές ομάδες είναι το *modus operandi* της ανάπτυξης του Gnutella, ταυτόχρονα όμως και ένας λόγος δημιουργίας ασυμβατοτήτων και προβλημάτων στο gNet.

Περιγραφή Πρωτοκόλλου

Η χρήστης Alice θέλει να συνδεθεί στο gNet. Πρώτα πρέπει να βρει και να κατεβάσει ένα από τα πολλά ανοικτού ή κλειστού, κώδικα συμβατά με το πρωτόκολλο Gnutella λογισμικά πελάτη (client software). Το λογισμικό αυτό κατά τη πρώτη του χρήση θα πρέπει να κάνει bootstrap και να βρει τουλάχιστον έναν άλλο Gnutella κόμβο. Οι συνήθεις μέθοδοι περιλαμβάνουν έτοιμες λίστες με διευθύνσεις πιθανά ενεργών κόμβων (pre-existing list of possibly working node addresses) που παρέχονται με το λογισμικό, τη χρήση σελίδων Internet τύπου Gwebcache (Gnutella web cache) που παρέχουν διευθύνσεις (πιθανά) ενεργών κόμβων, όπως και πληροφορία από στόμα σε στόμα (word of mouth). Εκτός και αν η χρήστης Alice είναι πολύ άτυχη, θα βρει τουλάχιστον έναν ενεργό κόμβο, έστω αυτόν του Bob. Μόλις η Alice συνδεθεί, ο Bob θα της στείλει τη δική του λίστα ενεργών κόμβων. Η χρήστης Alice θα προσπαθήσει να συνδεθεί σε αυτούς και σε όσους αυτοί τις στείλουν, μέσω των λιστών τους, εωσότου φτάσει κάποιο προκαθορισμό όριο (quota). Οι κόμβοι, που δε χρησιμοποίησε, αποθηκεύονται για μελλοντική χρήση και όσοι κόμβοι δεν λειτουργούσαν διαγράφονται από τη λίστα της. Από την επόμενη φορά που θα προσπαθήσει να επανασυνδεθεί θα χρησιμοποιεί τη δική της λίστα με πιθανά ενεργούς κόμβους, προκειμένου να διαφημίσει αναδρομικά το εαυτό της, όπως παραπάνω.

Η χρήστης Alice έχει κατορθώσει να συνδεθεί στο gNet, τώρα θέλει να κατεβάσει αρχεία σχετικά με κάτι, μέσω του λογισμικού της εκκινεί μια αναζήτηση με βάση κατάλληλες λέξεις κλειδιά. Στην αρχή θα ερωτηθούν όλοι οι κόμβοι με τους οποίους συνδέεται. Είναι πιθανό ένας αριθμός εξ αυτών να μη λειτουργούν πια, οπότε εκείνη προσπαθεί να συνδεθεί στους επόμενους κόμβους της λίστας της. Κάθε κόμβος θα προωθήσει την αίτηση της σε όλους τους κόμβους στους οποίους είναι συνδεμένος, εκείνοι με τη σειρά τους θα κάνουν το ίδιο και ούτο καθεξής. Θεωρητικά πάντα, η αίτηση θα φτάσει κάποια στιγμή σε κάθε χρήστη στο δίκτυο Gnutella. Στη πράξη, η αναζήτηση στο Gnutella είναι αναξιόπιστη. Κόμβοι έρχονται και παρέρχονται, και το δίκτυο μένει ποτέ σταθερό. Δεδομένου ότι το εύρος σύνδεσης κάθε χρήστη είναι περιορισμένο, μερικές αναζητήσεις μπορεί να απορριφθούν προτού φτάσουν σε όλο τι δίκτυο, το οποίο κατά μέσω όρο περιλαμβάνει 1.000.000 κόμβους κάθε στιγμή. Ως αποτέλεσμα οι περισσότερες αναζητήσεις δεν φθάνουν πάνω από το 50% του δικτύου [7].

Κάθε θετική απάντηση (positive response) δρομολογείται πίσω στην από το ίδιο μονοπάτι που έφτασε η ερώτηση. Έστω ότι ο χρήστης Godai-san διαμοιράζεται ένα αρχείο που φαίνεται να αντιστοιχεί στα κριτήρια της Alice και ότι η το επιλέγει προς κατέβασμα. Μια άμεση από σημείο σε σημείο σύνδεση (direct point to point connection) εγκαθιδρύεται μεταξύ του κόμβου της και του Godai-san, και το αρχείο κατεβαίνει χρησιμοποιώντας το πρωτόκολλο HTTP. Ο Godai-san σε αυτή τη περίπτωση λειτουργεί ως Web εξυπηρέτης ικανός να ανταποκριθεί σε αιτήσεις HTTP GET. Σε κάποιες επεκτάσεις του Gnutella εάν περισσότερα του ενός αντιγράφου του ίδιου αρχείου βρεθούν, εκκινείται ένα κατέβασμα τύπου σμήνους ("swarm" download), ήτοι κατέβασμα κομματιών του αρχείου από διαφορετικούς κόμβους, που επιταχύνει τη διαδικασία.

Εάν η Alice βρίσκεται πίσω από ένα ανάχωμα ασφαλείας, θα πρέπει να συνδεθεί με τους άλλους ομότιμους κόμβους μέσω μιας φιλικής, ως προς το ανάχωμα ασφαλείας, θύρας (port). Τυπικά αυτή είναι η θύρα 80, που έχει ανατεθεί στο πρωτόκολλο HTTP, το οποίο θεωρείται ασφαλές και μη κακόβουλο.

Εάν ο Godai-san δε μπορεί να δεχθεί HTTP συνδέσεις επειδή βρίσκεται πίσω από ανάχωμα ασφαλείας, η Alice έχει τη δυνατότητα να του στείλει ένα "Push-Request" πακέτο, ζητώντας του να πραγματοποιήσει μια εξωτερική (outbound) σύνδεση προς την Alice σε μια φιλική ως προς το ανάχωμα ασφαλείας θύρα, και να ανεβάσει (upload) τον αιτούμενο πόρο.

Εάν και οι δύο βρίσκονται πίσω από αναχώματα ασφαλείας το πρωτόκολλο καθίσταται μη λειτουργικό.

Χαρακτηριστικά και επεκτάσεις του πρωτοκόλλου

Το Gnutella βασίζεται σε ένα πρωτόκολλο τύπου πλημμυρίδας ερωτήσεων (query flooding). Η αρχική έκδοση του πρωτοκόλλου ορίζει 5 τύπους πακέτων για την σύνδεση και αναζήτηση στο Gnet.

- Ping: το πακέτο διαφήμισης (Advertisement packet) αποτελείται από το πλήθος των αρχείων που διαμοιράζεται ο χρήστης και το μέγεθός τους σε Kilobytes.
- Pong: απάντηση στο πακέτο Ping, που περιέχει ανάλογη πληροφορία.
- Query: αναζητά ένα αρχείο. Η σημασιολογία (semantics) των παραμέτρων αναζήτησης δεν καθορίζονται στην τρέχουσα έκδοση του πρωτοκόλλου.
- Query Hit: απάντηση στο Query, περιλαμβάνει το URL (Uniform Resource Locator) του αρχείου.
- Push-Request: αίτηση απόκτησης αρχείου για κόμβους που βρίσκονται πίσω από ανάχωμα ασφαλείας.

Οι επικεφαλίδες του Gnutella έχουν ως εξής:

Message ID (16 bytes)	**Function ID** (1 byte)	**TTL** (1 byte)	**Hops** (1 byte)	**Payload length** (4 bytes)

Πίνακας 6. Επικεφαλίδα Gnutella

- *Message ID (GUID):* ταυτοποιεί «μοναδικά» μια συναλλαγή σε συνάρτηση με την TCP/IP σύνδεση.
- *Function ID*: ένα από τα Ping, Pong, Query, Query Hit ή Push-Request.
- *TTL (time-to-live)*: χρόνος ζωής του μηνύματος, περιορίζει τη μέγιστη εξάπλωση του.
- *Hops*: Μετράει πόσες φορές ένα πακέτο προωθήθηκε, ήτοι πόσο μακριά είναι ο αρχικός αποστολέας.
- *Payload length*: Το μέγεθος των δεδομένων στο σώμα του μηνύματος.

Κλιμάκωση (Scalability)

Το Gnutella δε σχεδιάστηκε για δεκάδες χιλιάδες χρήστες. Έτσι όταν άρχισαν να συρρέουν ολοένα και περισσότεροι χρήστες το δίκτυο υπερφορτώθηκε και η ποιότητα υπηρεσιών του μειώθηκε αισθητά σε σημείο να άρνησης παροχής υπηρεσίας. Ιδέες που κατά τη σχεδίαση φαίνονταν καλές έδειξαν τα όριά τους από την αρχή κιόλας της λειτουργίας του δικτύου. Μια από αυτές είναι ο χρόνος ζωής των πακέτων. Σε κάποιες από τις αρχικές υλοποιήσεις το TTL δεν είχε ορισθεί καταλλήλως με αποτέλεσμα το δίκτυο να γεμίσει από “απέθαντα” πακέτα, που κατανάλωναν πολύτιμο εύρος ζώνης. Φυσικά αυτό είναι ένα σημείο ευπάθειας που ένας επιτιθέμενος θα μπορούσε να εκμεταλλευτεί. Ένα άλλο σημείο έχει να κάνει με τη ταχύτητα σύνδεσης των χρηστών. Οι χρήστες στο δίκτυο κάποια στιγμή θα κληθούν να παίξουν το ρόλο της πύλης (gateway) για να μπορέσουν άλλοι χρήστες να αναζητήσουν δεδομένα. Εάν το ρόλο της πύλης τον αναλάβει ένας χρήστης με χαμηλότερη ταχύτητα σε σχέση με τους υπόλοιπους, τότε δημιουργούνται bottlenecks. Επιπλέον όπως έχει ήδη αναφερθεί κανένας χρήστης δε μπορεί να εποπτεύσει όλο το δίκτυο. Χρησιμοποιώντας TTL = 5 και κρατώντας 4 συνδέσεις ανοιχτές κάθε στιγμή, κατά τη διαφήμιση και την αναζήτηση, μπορεί να φτάσει περίπου 4.000 κόμβους. Θα μπορούσε βέβαια να αυξήσει το TTL και το πλήθος των ανοικτών συνδέσεων. Αν υποθέσουμε ότι αυξάνουμε και τα δύο στο 8 τότε για μια απλή ερώτηση 18 Byte θα γεμίζαμε το δίκτυο με 1.2 Gigabytes (συναθροιζόμενων) δεδομένων. Ένα άλλο σημείο, σημαντικό από τη πλευρά της ασφάλειας, είναι η επιρρέπεια του Gnutella στης επιθέσεις άρνησης υπηρεσίας. Αρκεί ένας επιτιθέμενος να στείλει ένα μεγάλο πλήθος ερωτημάτων ή να αυξήσει το TTL για να αναλωθεί όλο το εύρος ζώνης στη γειτονιά του.

Τέλος ένα σημαντικό πρόβλημα είναι και η άρνηση πολλών χρηστών να προσφέρουνε πόρους στο δίκτυο [58]. Αυτό έχει ως αποτέλεσμα να καταλύεται ο αποκεντρωμένος χαρακτήρας του δικτύου και οι λιγοστοί κόμβοι που διαμοιράζονται περιεχόμενο να αναδύονται ως εξυπηρέτες. Τοιουτοτρόπως υποβιβάζεται η ποιότητα υπηρεσιών του δικτύου, αφού μεγάλη κίνηση περνάει από τις ίδιες γραμμές. Πέραν τούτου το δίκτυο εισέρχεται σε μια κατάσταση ομηρίας, όπου λίγοι ανασφαλείς εξυπηρέτες-χρήστες λειτουργούν ωσάν να ήταν οι διαχειριστές του δικτύου.

Εμπιστευτικότητα:

Δεν προβλέπεται από το πρωτόκολλο. Δε χρησιμοποιείται κρυπτογράφηση ούτε στα μηνύματα, ούτε στα ανταλλασσόμενα δεδομένα.

Ακεραιότητα:

Κάποιες ανεπίσημες επεκτάσεις του πρωτοκόλλου χρησιμοποιούν τη συνάρτηση σύνοψης SHA-1, προκειμένου να παράσχουν υπηρεσίες ακεραιότητας δεδομένων. Αναμένεται να υιοθετηθεί στην επόμενη έκδοση του πρωτοκόλλου.

Αυθεντικοποίηση:

Δε παρέχεται αυθεντικοποίηση του διαμοιραζόμενο περιεχόμενο του δικτύου. Δεδομένης και της ελεύθερης ερμηνείας των ερωτημάτων ο χρήστης δε μπορεί να ξέρει αν αυτό που κατεβάζει ανταποκρίνεται στις ανάγκες του, έως ότου αποκτήσει ένα αρκούντως μεγάλο μέρος αυτού για να κρίνει.

Επίσης στο αρχικό πρωτόκολλο δεν παρέχεται τρόπος αυθεντικοποίησης χρηστών. Αν και μερικά προγράμματα κλώνοι του Gnutella υλοποιούν επεκτάσεις του πρωτοκόλλου που επιτρέπουν τη δημιουργία ιδιωτικών δικτύων Gnutella. Για να συνδεθεί κάποιος σε ένα ιδιωτικό gNet θα πρέπει να γνωρίζει κάποιο μυστικό κωδικό ή άλλο αναγνωριστικό (προφανώς η κοινοποίηση αυτών στα νόμιμα μέλη αφήνεται ως άσκηση στους χρήστες). Τέτοια δίκτυα συνήθως προσφέρουν στα μέλη τους υπηρεσίες υψηλότερης ποιότητας, επιτρέπουν ορθολογικότερη διαχείριση και προσδίδουν την αίσθηση μεγαλύτερης εμπιστοσύνης.

Διαθεσιμότητα:

Καθώς οι ομότιμοι κόμβοι έρχονται και παρέρχονται, δε μπορεί να δοθεί καμία εγγύηση στο χρήστη ότι θα μπορέσει να βρει κάποιον κόμβο προκειμένου να συνδεθεί στο gNet, αν και υπάρχουν off-channel βοηθήματα όπως τα Gwebcaches. Επίσης η

διαθεσιμότητα πληροφορίας δεν εγγυάται. Ένα αρχείο παραμένει Gnet για όσο καιρό το επιτρέπει η δημοτικότητά του και η διακριτική ευχέρεια των κατόχων του. Παρά ταύτα η αμιγώς αποκεντρωποιημένη φύση και η ύπαρξη πλεονασμού στο δίκτυο κάνει πολύ δύσκολη τη καταστολή αυτού.

Ανωνυμία:

Όπως είδαμε οι ερωτήσεις περνάνε από κόμβο σε κόμβο και οι απαντήσεις επιστρέφουν διαμέσω των ίδιων κόμβων από όπου προήλθε η ερώτηση. Αυτό καθιστά ανούσια την καταγραφή των IP διευθύνσεων και των ερωτημάτων, καθότι δεν υπάρχει τρόπος να βεβαιωθούμε ποιος κόμβος εκκίνησε την ερώτηση. Κάθε κόμβος προωθεί αιτήσεις άλλων μελών του δικτύου, και ο μεμονωμένος χρήστης παραμένει κρυμμένος στο πλήθος των ομότιμων μελών του δικτύου. Εν κατακλείδι παρέχεται ασθενής ανωνυμία στις ερωτήσεις χάρη μέσω εύλογης αρνησικυρίας (plausible deniability).

Αντίθετα όσον αφορά την μεταφορά αρχείων, επειδή αυτή, για λόγους επίδοσης, παρέχεται μέσω άμεσων συνδέσεων HTTP χωρίς ενδιάμεσους, έχουμε αποκάλυψη της IP διεύθυνσης των δύο συμβαλλόμενων μελών. Αυτή η έλλειψη ανωνυμίας στη μεταφορά αρχείων οδήγησε στη δημιουργία του Wall of Shame. Μια λίστα σε ένα web site όπου αναρτιόνταν οι IP διευθύνσεις και τα domain ονόματα υπολογιστών που φέρονταν να κατέβασαν περιεχόμενο, το οποίο είχε ψευδός διαφημιστεί ότι περιέχει παιδική πορνογραφία.

Για αυτούς τους λόγους λέμε ότι το Gnutella παρέχει αδύναμη ανωνυμία (weak anonymity)

2.2 Gnutella 2 / Mike's Protocol

Υβριδικό Ομότιμο Δίκτυο Διαμοιρασμού Αρχείων

Ιστορία

Το Νοέμβριο του 2002 ο Michael Stokes ανακοίνωσε το πρωτόκολλο Gnutella2 [63]. Το Μάρτιο του 2003 κυκλοφόρησε μια πρώιμη έκδοση των προδιαγραφών του. Το πρωτόκολλο από τη πρώτη στιγμή της δημιουργίας του προκάλεσε ένα σχίσμα στους κόλπους των χρηστών του Gnutella, το οποίο ακόμα δεν έχει γεφυρωθεί.

Περιγραφή Πρωτοκόλλου

Το Gnutella2 χωρίζει τους κόμβους σε δύο κατηγορίες, φύλλα (leaves) και κέντρα (hubs). Κάθε φύλλο διατηρεί μια ή δύο συνδέσεις προς κέντρα, ενώ τα κέντρα δέχονται εκατοντάδες συνδέσεις από φύλλα και άλλα κέντρα. Όταν μια αναζήτηση εκκινήσει, ο κόμβος λαμβάνει μια λίστα από κέντρα αν χρειάζεται, και επικοινωνεί με κάθε κόμβο στη λίστα, έως ότου εξαντληθεί η λίστα ή φθάσει κάποιο προκαθορισμένο όριο. Αυτό επιτρέπει σε ένα χρήστη να βρει δημοφιλή αρχεία εύκολα χωρίς να επιβαρύνει το δίκτυο, ενώ ταυτόχρονα, στη θεωρία πάντα, επιτρέπει στο χρήστη να βρει ένα αρχείο οπουδήποτε και αν είναι στο δίκτυο.

Τα κέντρα καταγράφουν τι αρχεία έχει ένα φύλλο χρησιμοποιώντας ένα Πίνακα Δρομολόγησης Ερωτήματος (Query Routing Table), που περιέχει συνόψεις λέξεων κλειδί που το φύλλο ανέβασε στο κέντρο. Τα κέντρα συνδυάζουν τους πίνακες όλων των φύλλων τους και στέλνουν το αποτέλεσμα στα γειτονικά κέντρα, προκειμένου να μειωθεί ο αριθμός των ερωτημάτων που προωθούνται.

Το Gnutella2 βασίζει την επικοινωνία του, για τις αναζητήσεις, κυρίως στο UDP, καθώς η επιβάρυνση που θα πρόσφερε το TCP θα καθιστούσε το σύστημα αναζήτησης μη λειτουργικό. Η μορφή των πακέτων του πρωτοκόλλου είναι τέτοια που να επιτρέπει την μελλοντική επέκταση του και τη προσθήκη νέων υπηρεσιών. Αξίζει να σημειωθεί ότι χρησιμοποιείται συμπίεση στις δικτυακές συνδέσεις, σε μια προσπάθεια να μειωθεί η κατανάλωση του εύρους σύνδεσης.

Προκείμενου να παράσχει ακεραιότητα δεδομένων το Gnutella2 χρησιμοποιεί τη συνάρτηση σύνοψης SHA-1, η συνάρτηση χρησιμοποιείται με τέτοιο τρόπο που να κάνει δυνατό το ταυτόχρονο κατέβασμα (σμήνος) του ίδιου αρχείου από πολλές πηγές. Επίσης χρησιμοποιούνται Tiger-Tree συνόψεις για το αξιόπιστο ανέβασμα μερών ενός αρχείου που δεν έχει κατέβει ολόκληρο.

Για δημιουργήσει ένα πιο ρωμαλέο και ολοκληρωμένο σύστημα αναζήτησης το πρωτόκολλο υποστηρίζει ένα σύστημα μεταδεδομένων (metadata) που εμπλουτίζει τα αποτελέσματα της αναζήτησης με βαθμονομήσεις (ratings) και πληροφορία ποιότητας. Σημαντικό στοιχείο είναι ότι οι κόμβοι μπορούν να μοιράζονται αυτή τη πληροφορία και αφού έχουν σταματήσει να διαμοιράζονται ένα συγκεκριμένο αρχείο, γεγονός που επιτρέπει τη επισήμανση κακόβουλων προγραμμάτων και ιών στο δίκτυο χωρίς να χρειάζεται να κρατείται αντίγραφο αυτών.

Εμπιστευτικότητα:

Δεν παρέχεται. Δε χρησιμοποιείται κρυπτογράφηση ούτε στα μηνύματα, ούτε στα ανταλλασσόμενα δεδομένα.

Ακεραιότητα:

Χρησιμοποιείται η συνάρτηση σύνοψης SHA-1, προκειμένου να παρασχεθούν υπηρεσίες ακεραιότητας δεδομένων, μαζί με συνόψεις δέντρου τύπου Tiger-Tree.

Διαθεσιμότητα:

Όπως και στο Gnutella οι ομότιμοι κόμβοι έρχονται και παρέρχονται και άρα δε μπορεί να δοθεί καμία εγγύηση στο χρήστη ότι θα μπορέσει να βρει κάποιον κόμβο κέντρο (hub) προκειμένου να συνδεθεί στο δίκτυο, αν και μπορούν να βρεθούν λίστες με πιθανώς εν λειτουργία κέντρα. Επίσης η διαθεσιμότητα του περιεχομένου δεν εγγυάται. Ένα αρχείο παραμένει για όσο καιρό το επιτρέπει η δημοτικότητά του και η διακριτική ευχέρεια των κατόχων του.

Ανωνυμία:

Δεν περιλαμβάνεται στις προδιαγραφές του συστήματος

Αυθεντικοποίηση:

Το πρωτόκολλο χρησιμοποιεί μεταδεδομένα για να μπορέσουν οι χρήστες να βαθμονομούν τα αρχεία και να εισάγουν σχόλια και πληροφορία ποιότητας αυτών, χωρίς να είναι αναγκαίο να κατέχουν τα ίδια τα αρχεία. Όμως και πάλι υπάρχει το πρόβλημα της εμπιστοσύνης, καθώς ένας χρήστης θα πρέπει να εμπιστευτεί πληροφορία άγνωστης προέλευσης.

Επίσης δεν παρέχεται αυθεντικοποίηση χρηστών.

2.3 eDonkey2000 (eD2k)

Υβριδικό Ομότιμο Δίκτυο Διαμοιρασμού Αρχείων

Σύντομη Περιγραφή

Το δίκτυο **eDonkey2000** (**eDonkey**, **eD2k**) [15], [13] είναι ένα αποκεντρωμένο δίκτυο διαμοιρασμού αρχείων με κατανεμημένους εξυπηρέτες, που χρησιμοποιείται κυρίως για την ανταλλαγή μεγάλων σε μέγεθος πολυμεσικών αρχείων. Το δίκτυο διαχωρίζει τους κόμβους που το αποτελούν σε εξυπηρετούμενους, που τρέχουν κάποιο λογισμικό πελάτη, και εξυπηρέτες. Τα αρχεία βρίσκονται αποθηκευμένα στους υπολογιστές των ομότιμων εξυπηρετούμενων, ενώ οι eDonkey εξυπηρέτες λειτουργούν ως κέντρα επικοινωνίας (communication hubs) για τους εξυπηρετούμενους, ως κεντρικοί καταχωρητές πληροφορίας σχετικά με το διαμοιραζόμενο περιεχόμενο και τη θέση των κόμβων που το κατέχουν, κάνοντας δυνατή την αναζήτηση αρχείων στο δίκτυο. Οι εξυπηρέτες δεν είναι στατικοί και αλλάζουν διαρκώς, είναι ανεξάρτητοι μεταξύ τους και βρίσκονται διασκορπισμένοι ανά την υφήλιο.

Το eDonkey χρησιμοποιεί τη συνάρτηση σύνοψης MD4 για να ταυτοποιεί (identify) τα αρχεία, το οποίο επιτρέπει την ταυτοποίηση αντιγράφων ενός αρχείου ανεξαρτήτως του ονόματός τους. Ένα άλλο δημοφιλές χαρακτηριστικό του δικτύου είναι ότι επιτρέπει το διαμοιρασμό τμημάτων του αρχείου (file segments) πριν ακόμα ολοκληρωθεί το κατέβασμα όλου του αρχείου, αυτό έχει σαν αποτέλεσμα την γρήγορη εξάπλωση των αρχείων στο δίκτυο αλλά επίσης σημαίνει ότι διαδίδονται ημιτελή αρχεία.

Ιστορία

Το eDonkey δημιουργήθηκε από την Metamachine και το 2004, ξεπέρασε σε δημοτικότητα το δίκτυο FastTrack για να γίνει ένα από τα τρία μεγαλύτερα δίκτυα διαμοιρασμού αρχείων στο Internet. Υπολογίζεται ότι στα μέσα του 2005 φιλοξενούσε καθημερινά μερικές εκατομμύρια χρηστών, κάμποσα εκατοντάδες εκατομμύρια αρχείων και εκατοντάδες εξυπηρετών.

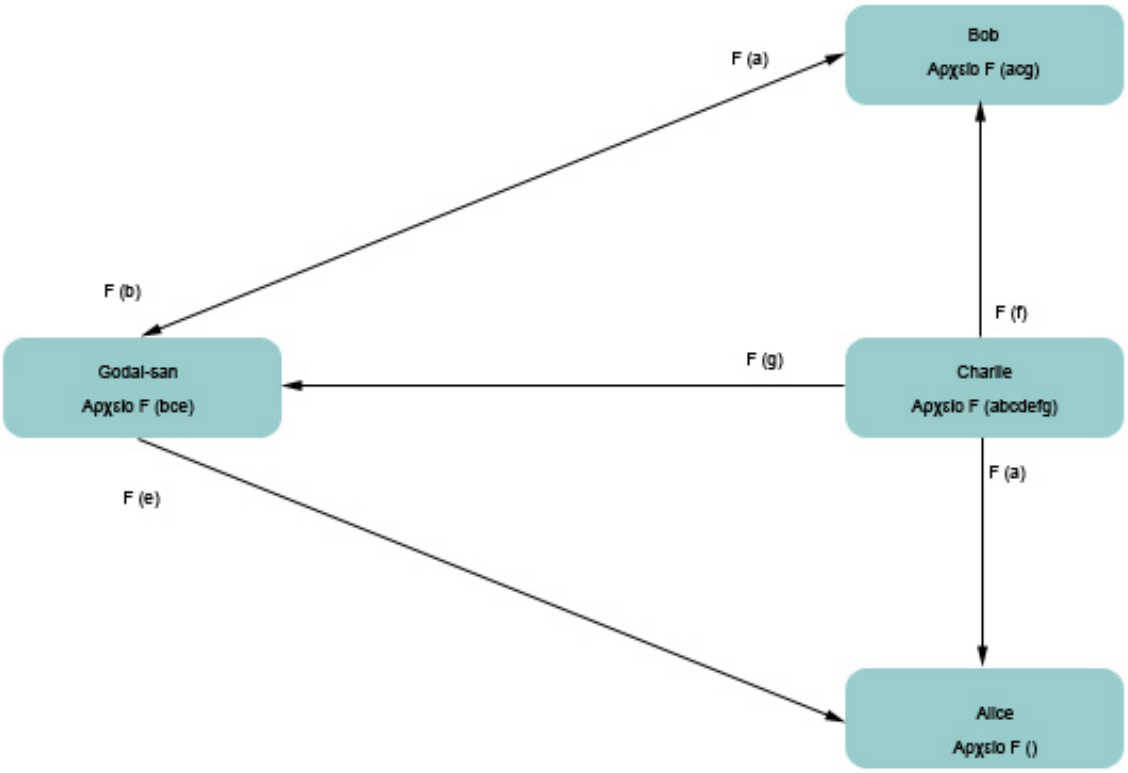
Περιγραφή Πρωτοκόλλου

Ο χρήστης Bob θέλει να συνδεθεί στο δίκτυο eDonkey2000. Πρώτα θα εγκαταστήσει λογισμικό πελάτη που να υλοποιεί το eD2k πρωτόκολλο (e-mule, edonkey2000, mldonkey, …). Ύστερα θα πρέπει να συνδεθεί σε έναν από τους εξυπηρέτες του δικτύου. Οι λίστες με διευθύνσεις εξυπηρετών (serverlists) περιέχονται στα αρχεία server.met που μπορούν να αποκτηθούν μέσα από πληθώρα σελίδων του Internet, είτε από κάποιον γνωστό εξυπηρέτη. Ο Bob θα εκκινήσει μια TCP σύνδεση με έναν εξυπηρέτη. Ύστερα θα στείλει ένα πακέτο τύπου “Hello” που περιέχει το ψευδώνυμο του Bob, τη σύνοψη χρήστη (user hash), την IP διεύθυνση και θύρα στην οποία ακούει το λογισμικό πελάτη και την έκδοση του πρωτοκόλλου που χρησιμοποιεί. Ακολούθως ο εξυπηρέτης αναθέτει μια ταυτότητα πελάτη (client ID) και τη στέλνει πίσω στον Bob. Εάν ο χρήστης Bob βρίσκεται πίσω από ανάχωμα ασφαλείας θα αποκτήσει ένα “low-id”, αυτό συμβαίνει διότι συνήθως τα IDs είναι της τάξης των εκατομμυρίων, ως μια αριθμητική αναπαράσταση της IP διευθύνσεως, όταν όμως παρεμβάλλεται ανάχωμα ασφαλείας ο εξυπηρέτης αναθέτει μια IP που αρχίζει από το 1. Στη συνέχεια ο εξυπηρέτης στέλνει ένα μήνυμα χαιρετισμού «greeting» και προαιρετικά επιπλέον πληροφορία για τον εαυτό του και για το πλήθος των χρηστών που είναι συνδεδεμένοι και των αρχείων που διαμοιράζονται.

Έχοντας πλέον συνδεθεί στο δίκτυο eD2k, ο Bob δύναται να πραγματοποιήσει αναζητήσεις αρχείων (ή συλλογών αρχείων). Ο Bob στέλνει μια συμβολοσειρά αναζήτησης και προαιρετικά κάποιες παραμέτρους αναζήτησης στον εξυπηρέτη. Οι παράμετροι συνήθως περιλαμβάνουν μέγιστο και ελάχιστο μέγεθος αρχείου, είδος αρχείου (ήχος, κινούμενη εικόνα, εικόνα, συλλογή, …) και ανάλογα με το είδος του αρχείου ο χρήστης μπορεί να αναζητήσει συγκεκριμένη διάρκεια, ανάλυση, αλγόριθμο κτλ. Για κάθε αρχείο που ταιριάζει στην αναζήτηση ο εξυπηρέτης απαντά με μια τριπλέτα της μορφής [όνομα αρχείου, σύνοψη αρχείου, μέγεθος αρχείου] και αν υπάρχει διαθέσιμη επιπλέον πληροφορία ανάλογα με το είδος του αρχείου (διάρκεια, αλγόριθμος κωδικοποίησης, …). Εναλλακτικά ο Bob μπορεί να βρει μια τέτοια τριπλέτα στο Internet με τη μορφή ενός eD2k συνδέσμου (link). Έχοντας βρει κάποιο αρχείο που να τον ενδιαφέρει, ο Bob στέλνει στον εξυπηρέτη ένα “query sources” μήνυμα, που περιέχει μόνο τη σύνοψη του ζητούμενου αρχείου. Ο εξυπηρέτης απαντάει με ένα μήνυμα που περιέχει τις πιθανές πηγές του αρχείου. Η πηγές δίνονται ως ζεύγη ID\θύρας για κάθε χρήστη που ισχυρίζεται ότι διαθέτει το αρχείο, το πλεονέκτημα αυτής της μορφής είναι ότι ο Bob μπορεί από το ID να καταλάβει ποιοι χρήστες βρίσκονται πίσω από αναχώματα ασφαλείας και να ζητήσει από τον εξυπηρέτη να στείλει σε αυτούς μια εντολή “push”, προκειμένου να του ανεβάσουν το αρχείο. Σε αυτό το σημείο αρχίζει η ομότιμη

επικοινωνία μεταξύ των χρηστών του δικτύου. Ο Bob στέλνει σε κάθε πηγή μια αίτηση για το κατέβασμα του αρχείου. Το δίκτυο χρησιμοποιεί το MFTP για το κατέβασμα των αρχείων.

Το MFTP (multisource file transmission protocol) έχει σχεδιαστεί έστω να βοηθάει στην ταχεία εξάπλωση μεταξύ πολλών ενδιαφερόμενων χρηστών. Δίνει τη δυνατότητα σε ένα χρήστη να κατεβάζει το ίδιο αρχείο από πολλές πηγές ταυτόχρονα, επιτυγχάνοντας έτσι υψηλές ταχύτητες κατεβάσματος μέσω της συνάθροισης των επί μέρους ταχυτήτων. Ένας χρήστης που κατεβάζει γίνεται ο ίδιος πηγή για το αρχείο μόλις κατορθώσει να συμπληρώσει ένα τμήμα (part) αυτού.



Εικόνα 4. Λειτουργία MFTP

Στο διάγραμμα ο ομότιμος χρήστης Charlie έχει αποθηκευμένο στον κόμβο του ολόκληρο το αρχείο F (το F αποτελείται από τα τμήματα a, b, c, d, e, f, g). Οι Bob, Alice και Godai-san θέλουν να κατεβάσουν το F. Αφού οι Bob και Godai-san έχουν από διαφορετικά τμήματα του αρχείου F μπορούν όχι μόνο να κατεβάσουν τα εναπομείναντα τμήματα από τον Charlie, αλλά και να προσφέρουν τα δικά τους στο δίκτυο. Με αυτό τον τρόπο το αρχείο διαδίδεται πιο γρήγορα, χωρίς να καταναλώνεται πολύ εύρος σύνδεσης από τον Charlie. Η Alice μπορεί να αρχίσει το κατέβασμα του F ακόμα και εάν ο Charlie δε μπορεί να την εξυπηρετήσει (βρίσκεται προσωρινά εκτός δικτύου ή δεν επαρκεί το

εύρος της σύνδεσής του). Δυστυχώς εάν ο Charlie σταματήσει να διαμοιράζεται το F, πριν διαδοθούν όλα τα τμήματά του, οι υπόλοιποι χρήστες θα συνεχίσουν να διαμοιράζονται το ημιτελές αρχείο.

Τμήμα θεωρούμε κομμάτια των 9500KByte ενός αρχείου, προφανώς εάν το αρχείο είναι μικρότερο ή εάν δε διαιρείτε ακριβώς με το 9500 χρησιμοποιούνται μικρότερα κομμάτια. Για κάθε τέτοιο τμήμα υπολογίζουμε μια σύνοψή του, με τον αλγόριθμο MD4, η οποία χρησιμοποιείται για τον έλεγχο της ακεραιότητας. Όταν εξακριβωθεί η ακεραιότητα ενός τμήμα, αυτό συνδυάζεται με άλλα τμήματα για την ανάπλαση του αρχικού αρχείου.

Μια άλλη μορφή ομότιμης επικοινωνίας που λαμβάνει χώρα είναι αυτή μεταξύ των εξυπηρετών. Οι εξυπηρέτες επικοινωνούν μεταξύ τους μόνο με μηνύματα UDP. Κάθε εξυπηρέτης που συνδέεται στο δίκτυο στέλνει μηνύματα αναγελλείας (“Announce”) για να γνωστοποιήση την ύπαρξή του στους ομότιμούς του. Επιπλέον οι εξυπηρέτες μπορούν να ανταλλάσσουν λίστες γνωστών εξυπηρετών, να ερωτούν και να απαντούν σχετικά με τη κατάστασή τους (“ping” και “pong” μηνύματα) και να στέλνουν τη περιγραφή τους

Μια αστοχία του πρωτοκόλλου είναι ότι επιτρέπει και στους εξυπηρετούμενους να ανταλλάσουν μηνύματα ping για να επιβεβαιώσουν την ορθότητα των λιστών εξυπηρετών τους. Αυτό έχει σαν αποτέλεσμα τα μηνύματα ping/pong να αποτελούν ένα υψηλό ποσοστό της όλης κυκλοφορίας στο δίκτυο.

Χαρακτηριστικά και επεκτάσεις του πρωτοκόλλου

Το edonkey2000 πρωτόκολλο ορίζει ότι οι TCP ροές ανάμεσα στον εξυπηρέτη και τον εξυπηρετούμενο χωρίζονται σε λογικά μηνύματα στο επίπεδο εφαρμογής.

Οι επικεφαλίδες είναι της μορφής:

e3 (1 byte)	Μέγεθος Μηνύματος (4 byte)	Τύπος Μηνύματος (1 byte)

Πίνακας 7. Επικεφαλίδα eD2k

- **e3**: υποδηλώνει ότι ακολουθεί μήνυμα του eD2k (Το λογισμικό πελάτη υποστηρίζει πολλαπλά πρωτόκολλα, πχ. Το e-mule παρέχει σύνδεση στο δίκτυο eD2k και στο Kademlia [12])

- **Μέγεθος Μηνύματος**: μέγεθος σε bytes του σώματος

- **Τύπος Μηνύματος**: δείκτης είδους μηνύματος

Το σώμα είναι της μορφής:

Σταθερό μέρος	Πλήθος Ετικετών	Ετικέτες	Δεδομένα

Πίνακας 8. Σώμα eD2k

- **Σταθερό Μέρος**: προκαθορισμένα δεδομένα ανάλογα με το τύπο του μηνύματος

- **Πλήθος Ετικετών (tags)**: δηλώνει πόσες ετικέτες ακολουθούν

- **Ετικέτες**: Μια ετικέτα (tag) είναι μια δομική αναπαράσταση συγκεκριμένου είδους δεδομένων. Οι ετικέτες έχουν την εξής μορφή:

Τύπος Δεδομένων (1 byte)	Μέγεθος Περιγραφής (2 bytes)	Περιγραφή (Μέγεθος Περιγραφής)

Πίνακας 9. Μορφή ετικέτας eD2k

Η λεπτομερής καταγραφή των ετικετών του eD2k ξεφεύγει από τους στόχους αυτής της διπλωματικής. Αρκεί να σημειωθεί ότι με αυτό τον τρόπο δίνεται η δυνατότητα επέκτασης του πρωτοκόλλου χωρίς να χάνεται η προς τα πίσω συμβατότητα, οι ετικέτες που δεν αναγνωρίζονται από ένα πρόγραμμα απλώς αγνοούνται.

(Σημείωση οι ακόλουθες προδιαγραφές ακολουθούνται από το λογισμικό πελάτη e-mule έκδοση v.44a+)

Ασφαλής Ταυτοποίηση Χρήστη και Σύστημα Ανταμοιβών

Κάθε χρήστης στο δίκτυο ταυτοποιείται από μια μοναδική τιμή, τη σύνοψη χρήστη, η οποία προστατεύεται με τη χρήση ασύμμετρης κρυπτογράφησης. Αυτή η ταυτότητα, που δεν έχει σχέση με τη ταυτότητα πελάτη, χρησιμοποιείται, κατόπιν επιλογής του χρήστη, μαζί με το ασύμμετρο κρυπτοσύστημα RSA για την υλοποίηση ενός ασφαλούς συστήματος ανταμοιβών (credit). Πιο αναλυτικά έστω ότι δύο χρήστες η Alice και ο Bob χρησιμοποιούν την ασφαλή ταυτοποίηση. Κάθε χρήστης έχει δημιουργήσει ένα ζεύγος RSA κλειδιών μεγέθους 384 bits (μέγεθος ικανοποιητικό για την υπηρεσία που υλοποιείται). Όταν πάνε να ανταλλάξουν δεδομένα για πρώτη φορά θα στείλουν ο ένας στον άλλο τα δημόσια κλειδιά τους και από ένα τυχαίο αριθμό έκαστος.

Οι τυχαίοι αριθμοί έχουν διάρκεια ζωής όσο και το χρονικό διάστημα που οι χρήστες παραμένουν στο δίκτυο, ενώ τα δημόσια κλειδιά αποθηκεύονται για μετέπειτα χρήση. Εάν η χρήστης Alice θέλει να ταυτοποιηθεί στον χρήστη Bob κρυπτογραφεί με το ιδιωτικό της κλειδί το δημόσιο κλειδί του Bob και τη τυχαία τιμή του Bob. Ο χρήστης Bob αρκεί να αποκρυπτογραφήσει το κρυπτογράφημα με το δημόσιο κλειδί της Alice και να ελέγξει την ακρίβεια των περιεχομένων. Αν όντως περιέχεται το δημόσιο κλειδί του και ο τυχαίος αριθμός του τότε η χρήστης Alice έχει ταυτοποιηθεί.

Το σύστημα ανταμοιβών επιβραβεύει χρήστες που συνεισφέρουν περιεχόμενο και πόρους στο δίκτυο. Τα λογισμικά πελάτη χρησιμοποιούν ουρές αναμονής (queues) για το κατέβασμα, όπου οι παλαιότεροι χρήστες έχουν προτεραιότητα. Με την εισαγωγή ανταμοιβών τροποποιείτε ο χρόνος αναμονής ανάλογα με τη συνεισφορά του χρήστη, ήτοι το λόγο ανεβάσματος και κατεβάσματος ανάμεσα στους δύο χρήστες. Όσο περισσότερο ανεβάζει αρχεία η χρήστης Alice στον χρήστη Bob, τόσο προωθείται στην αρχή της ουράς αναμονής του Bob. Για να υπολογιστεί η συνεισφορά της Alice στον Bob λαμβάνεται υπόψη το ελάχιστο δύο τιμών:

$$συνεισφορά = \min\left(\frac{\text{Ολικό Ανέβασμα} \times 2}{\text{Ολικό Κατέβασμα}}, \sqrt{\text{Ολικό Ανέβασμα} + 2}\right)$$

Εάν

$\text{Ολικό Ανέβασμα} < 1MB \Rightarrow συνεισφορά = 1$,

$\text{Ολικό Κατέβασμα} = 0 \Rightarrow συνεισφορά = 10$,

όπου $συνεισφορ.ά \in [1,10]$

Συνόψεις Αρχείων και ICH (Έξυπνος Χειρισμός Παραφθορών, Intelligent Corruption Handling)

Σε κάθε διαμοιραζόμενο αρχείο αντιστοιχίζεται μια μοναδική, που παράγεται με τη συνάρτηση σύνοψης MD4. Η *σύνοψη αρχείου* υπολογίζεται χωρίζοντας το αρχείο σε τμήματα των 9.28 MB. Για κάθε τμήμα υπολογίζεται με τον αλγόριθμο MD4 η *σύνοψη τμήματος* (Part Hash). Το σύνολο των συνόψεων τμημάτων ονομάζεται *Hashset*, και από αυτό παράγεται η τελική σύνοψη αρχείου.

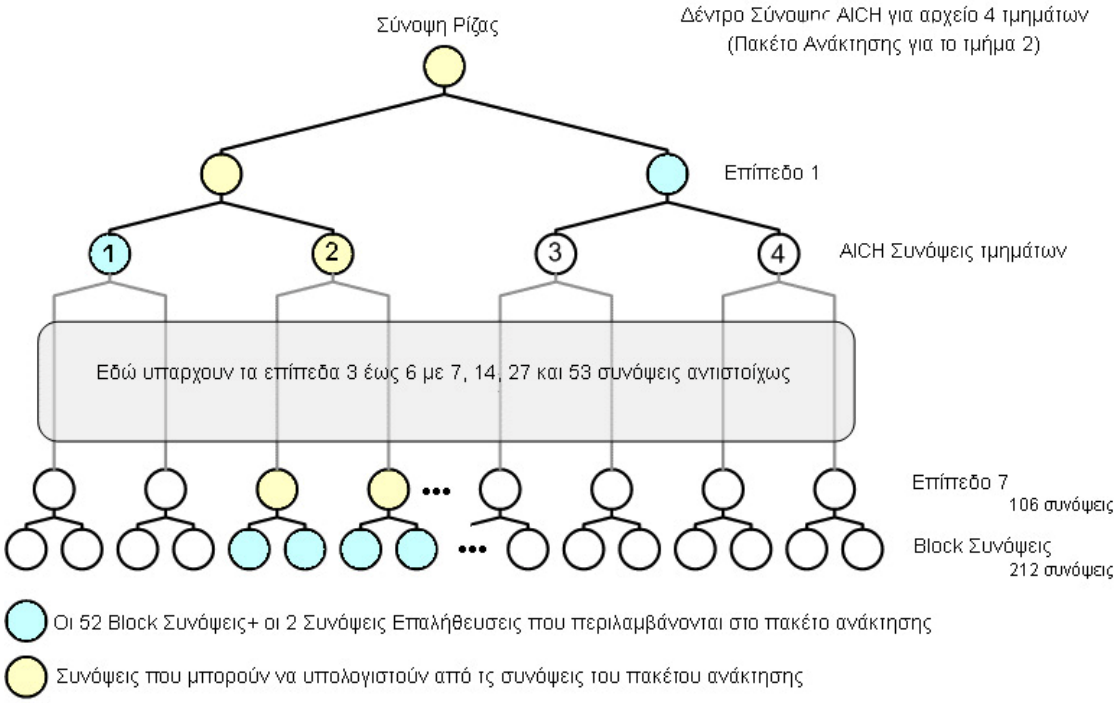
Κάθε φορά που ολοκληρώνεται το κατέβασμα ενός τμήματος ελέγχεται ως προς την αντίστοιχη σύνοψη τμήματος. Εάν ο έλεγχος είναι επιτυχής το τμήμα προσφέρεται για ανέβασμα. Διαφορετικά θα πρέπει να ξανακατέβει. Προκειμένου να αποφευχθεί η επανάληψη και των 9,28MB του τμήματος, το ICH ξανακατεβάζει τα πρώτα 180KB και επαναλαμβάνει των έλεγχο. Εάν πάλι υπήρξε αποτυχία κατεβάζονται τα επόμενα 180KB και ούτο καθεξής. Στην καλύτερη περίπτωση χρειάζεται να ξαναληφθούν μόνο 180KB, στη χειρότερη όλο το αρχείο.

AICH (Προηγμένος Έξυπνος Χειρισμός Παραφθορών, Advanced Intelligent Corruption Handling)

Το ICH είναι μεν αποτελασματικο, αλλά έχει περιορισμούς καθώς δεν μπορούν να επαληθευτούν κομμάτια μικρότερα των 9.28 MB. Έτσι εάν περισσότερα του ενός σημεία έχουν πρόβλημα ή εάν ένας κακόβουλος χρήστης διαδίδει παρεφθαρμένα δεδομένα ή εάν ο σύνδεσμος περιέχει μια εσφαλμένη συνάρτηση σύνοψης το ICH γίνεται μη λειτουργικό. Το AICH προσφέρει βελτιωμένες υπηρεσίες ακεραιτότητας δεδομένων με ελάχιστο κόστος με τη δημιουργία συνόψεων μικρότερων κομματιών.

Root Hash, Block Hashes & AICH Hashset

Κάθε τμήμα των 9.28 MB ενός αρχείου χωρίζεται σε blocks των 180KB, ήτοι 53 blocks ανά τμήμα, για κάθε block υπολογίζεται η τιμή σύνοψής του με χρήση του αλγόριθμου SHA1. Οι τιμές αυτές ονομάζονται *Block Συνόψεις (Block Hashes)* και αποτελούν το χαμηλότερο επίπεδο του τελικού *AICH Hashset*.



Εικόνα 5. Δένδρο Σύνοψης AICH

Στην εικόνα βλέπουμε πως ένα πλήρες δέντρο σύνοψης (hash tree) χτίζεται πάνω από τα blocks 4 τμημάτων ενός αρχείου. Κάθε τμήμα περιέχει 53 blocks, δηλαδή 212 *Block Συνόψεις* που σχηματίζουν ένα δέντρο 7 επιπέδων μέχρι τη Σύνοψη Ρίζας. Τα δέντρο αυτό ονομάζεται *AICH Hashset*. Οι κυανές και κίτρινες κουκίδες δείχνουν τις μαθηματικές εξαρτήσεις των μικρότερων *Block Συνόψεων* ως προς τη Σύνοψη Ρίζας.

Όποτε παρατηρηθεί μια παραφθορά σε ένα τμήμα το λογισμικό πελάτη ζητά ένα Πακέτο Ανάκτησης (Recovery Packet) από έναν τυχαίο χρήστη που έχει ολόκληρο το AICH Hashset. Αυτό το πακέτο θα περιέχει τις 53 *Block Συνόψεις* του παρεφθαρμένου τμήματος καθώς και τις *Συνόψεις Επαλήθευσης* (*Verifying Hashes*) ολόκληρου του δέντρου. Στην εικόνα φαίνεται το Πακέτο Ανάκτησης για ένα αρχείο 4 τμημάτων. Το πλήθος τον Συνόψεων Επαλήθευσης καθορίζεται το τμήμα που επαληθεύεται σύμφωνα με τον εξής τύπο:

$$2^{\pi\lambda\acute{\eta}\theta o\varsigma\Sigma\upsilon\nu\acute{o}\psi\varepsilon\omega\nu\Sigma\pi\alpha\lambda\acute{\eta}\theta\varepsilon\upsilon\sigma\eta\varsigma o\varsigma} \geq \Alpha\rho\iota\theta\mu\acute{o}\Tau\mu\acute{\eta}\mu\alpha\tau o\varsigma$$

Αφού ληφθεί το Πακέτο Ανάκτησης ελέγχονται οι *Συνόψεις Επαλήθευσης* ως προς τη Σύνοψη Ρίζας, την οποία εμπιστευόμαστε. Αν αντιστοιχούνται, ελέγχονται και τα 53 blocks του παρεφθαρμένου τμήματος με βάση τις *Block Συνόψεις* του Πακέτου Ανάκτησης. Τα blocks που βρέθηκαν παρεφθαρμένα ξανακατεβάζονται.

Προκειμένου να αποκτήσουμε μια έμπιστη Σύνοψη Ρίζας (*Trusted Root Hash*) θα πρέπει να βρούμε έναν eD2k σύνδεσμο από ένα φορέα που εμπιστευόμαστε. Διαφορετικά, στη περίπτωση που δεν έχουμε τη Σύνοψη Ρίζας, θα πρέπει να εμπιστευτούμε αυτή που μας στέλνουν οι πηγές του αρχείου. Προκειμένου να πειστούμε για την εγκυρότητα αυτής θα πρέπει η ίδια τιμή να στέλνεται από τουλάχιστον 10 διαφορετικές πηγές και ταυτόχρονα το 92% των πηγών να συμφωνούν με αυτή τη τιμή. Επειδή στη δεύτερη περίπτωση η Σύνοψη Ρίζας δεν είναι και τόσο αξιόπιστη όσο η πρώτη, φυλάσσεται μόνο για τη διάρκεια της συνόδου. Μόλις το αρχείο ληφθεί επιτυχώς κατασκευάζεται *ολόκληρο το AICH Hashset*, και η Σύνοψη Ρίζας διαδίδεται στους άλλους χρήστες.

eD2k Σύνδεσμοι (eD2k Links)

Τα eD2k links είναι μια ειδική μορφή συνδέσμων που επιτρέπουν την άμεση εκκίνηση ενός κατεβάσματος αρχείου. Υποστηρίζονται οι εξής μορφές.

Βασικός σύνδεσμος eD2k
eD2k://|file|<όνομα αρχείου>|<μέγεθος αρχείου>|<σύνοψη αρχείου>|/ Η πιο βασική μορφή συνδέσμου, περιέχει τα απολύτως απαραίτητα όνομα, μέγεθος και σύνοψη αρχείου. Πχ: ed2k://|file|name|12043984|6744FC42EDA527B27F0B2F2538728B3E|/
Σύνδεσμος eD2k με hash set
eD2k://|file|<όνομα αρχείου>|<μέγεθος αρχείου>|<σύνοψη αρχείου>|p=<hash set>|/ Παρέχοντας ολόκληρο το hash set βοηθάει στη ορθή και ταχεία εξάπλωση καινούριων και σπάνιων αρχείων. Πχ: ed2k://|file|name|12043984|6744FC42EDA527B27F0B2F2538728B3E |p=264E6F6B587985D87EB0157A2A7BAF40:17B9A4D1DCE0E4C2B672DF257145E98A|/
Σύνδεσμος eD2k με πηγές
eD2k://|file|<όνομα αρχείου>|<μέγεθος αρχείου>|<σύνοψη αρχείου>|/|sources,<IP:θύρα>|/ Προσθέτει πολλαπλές ήδη γνωστές πηγές με τη μορφή <IP:θύρα> στο σύνδεσμο, επιτρέποντας το άμεσο κατέβασμα
Σύνδεσμος eD2k με ξενιστές
eD2k://|file|<όνομα αρχείου>|<μέγεθος αρχείου>|<σύνοψη αρχείου>|/|sources, <όνομα ξενιστή:θύρα>|/ Ίδιο με τις πηγές μόνο που αντί για την IP διεύθυνση χρησιμοποιεί το όνομα της πηγής
Σύνδεσμος eD2k σε HTML
<a href="eD2k://|file|<όνομα αρχείου>|<μέγεθος αρχείου>|<σύνοψη αρχείου>|/">κείμενο που εμφανίζεται στη σελίδα Σύνδεσμος έτοιμος για παρουσίαση σε web σελίδα
Σύνδεσμος eD2k με πηγές HTTP
eD2k://|file|<όνομα αρχείου>|<μέγεθος αρχείου>|<σύνοψη αρχείου>|s=<hyperlink αρχείου>|/ Επιτρέπει το κατέβασμα και από σελίδα web

Πίνακας 10. Είδη eD2k συνδέσμων

Σύνδεσμος eD2k με Σύνοψη Ρίζας (Root Hash)
eD2k://|file|<όνομα αρχείου>|<μέγεθος αρχείου>|<σύνοψη αρχείου>|h=<Σύνοψη Ρίζας>|/ Επιτρέπει τη χρήση AICH. Πχ: ed2k://|file|name|12043984|6744FC42EDA527B27F0B2F2538728B3E |h=A2NWOTYURUU3P3GCUB6KCNW3FTYYELQB|/

Εμπιστευτικότητα:

Δεν παρέχεται. Δε χρησιμοποιείται κρυπτογράφηση ούτε στα μηνύματα, ούτε στα ανταλλασσόμενα δεδομένα.

Ακεραιότητα:

Χρησιμοποιείται ο αλγόριθμος MD4 για την υπηρεσία ICH (Έξυπνος Χειρισμός Παραφθορών). Κάποιες επεκτάσεις του πρωτοκόλλου χρησιμοποιούν τη συνάρτηση σύνοψης SHA-1, προκειμένου να παράσχουν καλύτερες υπηρεσίες ακεραιότητας δεδομένων μέσω του AICH (Προηγμένος Έξυπνος Χειρισμός Παραφθορών)

Αυθεντικοποίηση:

Το πρόβλημα της αυθεντικοποίησης του διαμοιραζόμενου περιεχόμενου του δικτύου, λόγω των μηχανισμών παροχής ακεραιότητας, τρέπεται σε πρόβλημα εμπιστοσύνης της προέλευσης του eD2k συνδέσμου του αντίστοιχου αρχείου. Όπως δείχθηκε στη περιγραφή της υπηρεσίας AICH για να μπορέσει ένας κακόβουλος χρήστης να παραφθείρει ένα αρχείο θα πρέπει να ελέγχει πλέον του 92% των κόμβων που εμφανίζονται ως πηγές αυτού. Δυστυχώς δεν υπάρχει αξιόπιστος μηχανισμός που να εγγυάται ότι το περιεχόμενο ενός αρχείου συμβαδίζει με τη περιγραφή του, παρέχεται, όμως, η δυνατότητα βαθμονόμησης (rating) και εισαγωγής σχολίων σε ένα αρχείο από τους χρήστες που το διαμοιράζονται. Επίσης η αυθεντικοποίηση των χρηστών δεν κρίνεται επαρκής, αφού ναι μεν κάθε κόμβος χαρακτηρίζεται μοναδικά από τη σύνοψη χρήστη, αλλά δε παρέχεται κάποια άλλη πληροφορία που να μπορεί να χρησιμοποιηθεί για να αυξήσει το επίπεδο εμπιστοσύνης.

Διαθεσιμότητα:

Καθώς οι εξυπηρέτες αλλάζουν δυναμικά επαφίεται στο χρήστη να βρει λίστα με εξυπηρέτες εν λειτουργία. Για το σκοπό αυτό χρησιμοποιούνται παράπλευρα κανάλια (off-channel), όπως το web. Επίσης η διαθεσιμότητα πληροφορίας δεν εγγυάται. Ένα αρχείο παραμένει στο δίκτυο eD2k για όσο καιρό το επιτρέπει η δημοτικότητά του και η διακριτική ευχέρεια των κατόχων του. Η ύπαρξη πλεονασμού τόσο στους εξυπηρέτες όσο και στις πηγές αρχείων κάνουν το δίκτυο αρκετά εύχρηστο και ρωμαλέο.

Ανωνυμία:

Δε παρέχεται ανωνυμία. Η συνύπαρξη ταυτοτήτων πελάτη, που βασίζονται στην IP διεύθυνση, και συνόψεων χρήστη, δείχνουν ότι η ανωνυμία δεν είναι ένας από τους στόχους του πρωτοκόλλου.

Απονομή Ευθυνών:

Το σύστημα χρησιμοποιεί ένα σύστημα απονομής ανταμοιβών για την επιβράβευση των χρηστών που προσφέρουν πόρους στο δίκτυο, η επιβράβευση έχει τη μορφή της καλύτερης εξυπηρέτησης αυτών των χρηστών όταν βρίσκονται σε ουρές αναμονής. Δυστυχώς το σύστημα ενώ μπορεί με τη χρήση ασύμμετρης κρυπτογραφίας και την ασφαλή αυθεντικοποίηση των χρηστών να παρέχει ένα ικανοποιητικό επίπεδο ασφάλειας, με την έννοια ότι δύσκολα κάποιος θα κλέψει το σύστημα, ωστόσο ο κάθε ομότιμος κόμβος χρησιμοποιεί μόνο τοπική πληροφορία. Αυτό έχει σημαίνει ότι δεν υπάρχει μια συνολική και καθολική ενημέρωση για το ποιόν των χρηστών και άρα δε μπορούν να τιμωρηθούν οι κόμβοι που δε συμπεριφέρνονται. Ο κάθε κόμβος μπορεί να κρίνει μόνο από τις προηγούμενες συναλλαγές του και η καλή του συμπεριφορά δεν τον ακολουθεί σε όλες τις μετέπειτα συναλλαγές του.

2.4 Free Haven

Υβριδικό Ομότιμο Δίκτυο Αποθήκευσης Εγγράφων

Σύντομη Περιγραφή

Το Free Haven [7], [18], [19], [20] είναι ένα σύστημα έκδοσης που δύναται αντισταθεί στις προσπάθειες δυνατών αντιπάλων να βρουν ή να καταστρέψουν αποθηκευμένα δεδομένα. Προσφέρει ανωνυμία στους αναγνώστες και τους εκδότες, καθώς επίσης κρύβει την τοποθεσία των εξυπηρετών που αποθηκεύουν και προμηθεύουν τα έγγραφα. Σε αντιδιαστολή με το Freenet, ο εκδότης ενός εγγράφου, και όχι ο εξυπηρέτης που το φυλάσσει, καθορίζει τη διάρκεια ζωής του. Προκειμένου να αντιμετωπιστούν κακόβουλοι ή εσφαλμένοι εξυπηρέτες, ο εκδότης σπάει το έγγραφο σε μερίδια (shares), οιοιδήποτε αρκούντος μεγέθους υποσύνολο των οποίων είναι αρκετό για την ανάκτηση του αρχικού εγγράφου. Οι εξυπηρέτες μπορούν να ανταλλάσουν αυτά τα μερίδια, επιτρέποντας την απρόσκοπτη συμμετοχή και απόσυρση τους και επιπλέον παρέχοντας έναν κινούμενο στόχο στους επιτιθέμενους που αναζητούν ένα συγκεκριμένο μερίδιο.

Για να αποτραπούν κακόβουλοι χρήστες από το να γεμίσουν το διατιθέμενο αποθηκευτικό χώρο, όσοι εκδίδουν οφείλουν να παρέχουν εξυπηρετές, και με τη σειρά τους οι εξυπηρετές δημιουργούν συμβόλαια για να αποθηκεύουν ο ένας το περιεχόμενο του άλλου για ορισμένες χρονικές περιόδους. Επειδή κακόβουλοι εξυπηρετές μπορεί να αφαιρέσουν περιεχόμενο νωρίτερα, το Free Haven χρησιμοποιεί ένα σύστημα υπόληψης (reputation) για να περιορίσει τη ζημιά. Η επιτυχής ικανοποίηση ενός συμβολαίου αυξάνει την υπόληψη ενός εξυπηρέτη και άρα την ικανότητα του να αποθηκεύει τα δεδομένα του σε άλλους εξυπηρέτες.

Σε ένα τέτοιο δυναμικό και ανώνυμο περιβάλλον, είναι δύσκολο να αναγνωριστεί με βεβαιότητα το γεγονός ότι ένας εξυπηρέτης αφαίρεσε δεδομένα νωρίτερα. Μια λύση είναι να αναλάβει τον έλεγχο, της διαθεσιμότητας των εγγράφων του, ο εκδότης. Εναλλακτικά μπορούμε να θέσουμε έναν τυχαίο εξυπηρέτη ως ποιμένα ενός εγγράφου, ή ακόμα να χρησιμοποιήσουμε ένα “buddy system”, όπου οι εκδότες βάζουν δύο αντίγραφα κάθε μεριδίου, και τα αντίγραφα προσέχουν το ένα το άλλο και κάνουν broadcast ένα παράπονο εάν κάποιο εξαφανιστεί.

Ένα από τα ανοικτά προβλήματα του Free Haven είναι η επιβεβαίωση της αλήθειας μιας αίτησης κακής συμπεριφοράς (claim misbehaviour), η τρέχουσα προδιαγραφή του συστήματος χρησιμοποιεί υπογεγραμμένα συμβόλαια. Όμως,

υπάρχουν πολλά σημεία ευπάθειας σε αυτή τη λύση, η οποία, σημειωτέων, έχει μεγάλη πολυπλοκότητα.

Ένα από τα πιο σημαντικά στοιχεία του συστήματος υπόληψης του Free Haven είναι ότι οι εξυπηρέτες δύνανται να εξαργυρώσουν την υπόληψή τους προκειμένου να αποκτήσουν πόρους από άλλους εξυπηρέτες. Για να μπορέσει να αποκτήσει την απαιτούμενη υπόληψη, ούτως ώστε να μπορέσει να αποθηκεύσει συγκεκριμένο μέγεθος δεδομένων σε άλλων εξυπηρέτη, ένας εξυπηρέτης θα πρέπει πρώτα να έχει επιτυχώς αποθηκεύσει το ίδιο μέγεθος ξένων εγγράφων.

Ωστόσο θα πρέπει να σημειωθεί ότι το Free Haven υστερεί κατά πολύ σε απόδοση και ευκολία χρήσης. Ο βασικός κατασκευαστικός του στόχος είναι η ανωνυμία και η διαθεσιμότητα των εγγράφων και όχι η συχνή αναζήτηση. Οι κατασκευαστές του θεωρούν ότι τα έγγραφα του Free Haven θα διαμοιράζονται μέσω ενός πιο εύχρηστου (ομότιμου) δικτύου. Και ότι οι αναγνώστες θα χρησιμοποιούν το Free Haven για να προμηθευτούν ένα έγγραφο μόνο στη περίπτωση που απαιτείται ισχυρή ανωνυμία ή που το συγκεκριμένο έγγραφο δεν είναι διαθέσιμο από αλλού.

Ιστορία

Το πρόγραμμα Free Haven Project άρχισε το Δεκέμβριο του 1999 ως ένα ερευνητικό πρόγραμμα από φοιτητές του MIT για το σχεδιασμό, την υλοποίηση ενός λειτουργικού καταφύγιου δεδομένων (data haven). Η ανάπτυξη του συνεχίζεται ακόμα με την έρευνα να εστιάζει στην δημιουργία μιας ισχυρότερης υποδομής ανώνυμης επικοινωνίας και το σχεδιασμό ενός λειτουργικού συστήματος απόδοσης ευθυνών.

Περιγραφή Πρωτοκόλλου

Όπως προείπα το Free Haven Project έχει ως στόχο την ανάπτυξη ενός συστήματος που να προσφέρει την απαραίτητη υποδομή για ανώνυμη έκδοση εγγράφων. Συγκεκριμένα ο εκδότης ενός εγγράφου πρέπει να παραμένει άγνωστος, ο χρήστης που ζητά το έγγραφο δε θα πρέπει να ταυτοποιείται σε κανέναν και η τρέχουσα τοποθεσία του εγγράφου θα πρέπει να μένει κρυφή. Επιπλέον θα πρέπει να ελαχιστοποιούνται οι δυνατότητες ενός τρίτου να αποδείξει ότι ένα συγκεκριμένο έγγραφο πέρασε από ένα συγκεκριμένο κόμβο.

Ο σχεδιασμός του είναι βασισμένος σε μια κοινότητα εξυπηρετών (servnet), όπου κάθε εξυπηρέτης φιλοξενεί δεδομένα άλλων εξυπηρετών με αντάλλαγμα την δυνατότητα να αποθηκεύσει τα δικά του δεδομένα στο servnet. Κάθε εξυπηρέτης έχει ένα ζεύγος δημοσίου και ιδιωτικού κλειδιου και μια ή περισσότερες remailer reply blocks (διεύθυνση στο δίκτυο ανώνυμης επικοινωνίας mixnet), που μαζί μπορεί να προσφέρουν ασφαλής, αυθεντικοποιημένη, ψευδώνυμη επικοινωνία με εκείνον τον εξυπηρέτη.

Όταν κάποιος θέλει να εκδώσει ένα έγγραφο, το διασπά σε μερίδια σύμφωνα με τον αλγόριθμο διασποράς πληροφορίας (information dispersal) του Rabin [64], εκ των οποίων ένα υποσύνολο k από τα n είναι αρκετό για την ανάκτηση του εγγράφου. Παράγει ένα ζεύγος δημοσίου και ιδιωτικού κλειδιού ($PK_{αρχείου}$, $SK_{αρχείου}$) και υπογράφει ψηφιακά με το ιδιωτικό κλειδί ($SK_{αρχείου}$) κάθε μερίδιο. Ύστερα διαπραγματεύεται με έναν εξυπηρέτη την έκδοση των μεριδίων στο servnet, ο τρόπος που θα επικοινωνήσει ο εκδότης με τον εξυπηρέτη δεν καλύπτεται από το πρωτόκολλο. Ο εξυπηρέτης με τη σειρά του ανταλλάσσει παρασκηνιακά τα μερίδια με άλλους.

Όταν η Alice θελήσει να ανακτήσει ένα κείμενο πρέπει να γνωρίζει το $PK_{αρχείου}$, με το οποίο θα επιβεβαιώσει την αυθεντικοποίηση και ακεραιότητα των μεριδίων και άρα του εγγράφου, και πάλι ο τρόπος εύρεσης του κλειδιού δε προσδιορίζεται από το πρωτόκολλο. Ύστερα θα πρέπει να δημιουργήσει ένα ζεύγος δημοσίου και ιδιωτικού κλειδιού (PK_{Alice}, SK_{Alice}), καθώς και μια διεύθυνση remailer reply block μιας χρήσης, αυτά τα δύο θα χρησιμοποιηθούν για να προστατευτεί η ιδιωτικότητα της. Τέλος ζητάει από έναν οποιοδήποτε εξυπηρέτη, και πάλι ο τρόπος εύρεσης ενός πρόθυμου εξυπηρέτη δεν περιγράφεται στο πρωτόκολλο, να βρει το συγκεκριμένο αρχείο και του στέλνει μαζί με την αίτηση το PK_{Alice} και τη διεύθυνση μιας χρήσεως. Ο εξυπηρέτης θα κάνει broadcast την εξής αίτηση *"('request', $PK_{αρχείου}$* PK_{Alice}*, reply block)"* σε όλους τους άλλους εξυπηρέτες. Οι εξυπηρέτες του δικτύου ελέγχουν με το *$PK_{αρχείου}$* εάν κατέχουν ένα μερίδιο αυτού του εγγράφου και εάν ναι το κρυπτογραφήσουν με το PK_{Alice} και το στέλνουν μέσω του remailer στην προκαθορισμένη τοποθεσία.

Η τρέχουσα υλοποίηση υποδομής ανώνυμης επικοινωνίας βασίζεται στο Mixmaster Type II remailer, που προτάθηκε από τον Lance Cottrell (βλ. Electronic Frontiers Georgia (EFGA), Anonymous remailer information, http://anon.efga.com/Remailers/). Κάθε Mixmaster remailer έχει ένα RSA ζεύγος δημοσίου και ιδιωτικού κλειδιού και χρησιμοποιεί υβριδική συμμετρική κρυπτογραφία. Κάθε μήνυμα κρυπτογραφείται με ένα ξεχωριστό 3DES κλειδί για κάθε mix κόμβο στην αλυσίδα ανάμεσα στον αποστολέα και το δέκτη. Τα 3DES κλειδιά με τη σειρά τους κρυπτογραφούνται με τα δημόσια κλειδιά των αντίστοιχων κόμβων. Όλα τα μηνύματα συμπληρώνονται ώστε να έχουν το ίδιο μήκος.

Αξίζει να σημειωθεί μια πιθανή επίθεση κατά της ανωνυμίας του συστήματος που προκύπτει από τη χρήση ορισμένων κρυπτοσυστημάτων δημοσίου κλειδιού. Ορίζεται ως κρυπτοσύστημα δημοσίου κλειδιού απόκρυψης-παραλήπτη (recipient-hiding) το σύστημα εκείνο για το οποίο, δεδομένου ενός κρυπτογράφημα, είναι αδύνατο να προσδιορίσει κανείς το δημόσιο κλειδί που χρησιμοποιήθηκε για τη δημιουργία του. Τα περισσότερα κρυπτοσυστήματα δε σχεδιάστηκαν για να προσφέρουν ανωνυμία, για παράδειγμα το PGP ενσωματώνει τη ταυτότητα του παραλήπτη στις κεφαλίδες του, ενώ

στην περίπτωση του RSA, επειδή διαφορετικά δημόσια κλειδιά έχουν διαφορετικά moduli, το αποτέλεσμα της πράξης (ροή κρυπτογράφηματος) modulo (modulus “λάθος κλειδιού”) τείνει να δώσει διαφορετική κατανομή από ότι αν πέρναμε το module ως προς το “σωστό” modulus. Αυτό επιτρέπει σε έναν επιτιθέμενο να ψάξει μέσα από ένα σύνολο γνωστών δημοσίων κλειδιών για να βρει αυτό που ταιριάζει καλυτερα στο κρυπτογράφημα. Έχουν προταθεί διάφορες λύσεις για αυτό το πρόβλημα. Ο Rivest πρότεινε τη χρήση randomized κρυπτοσυστημάτων, όπως αυτό των Goldwasser-MIcalli, ενώ οι Lysyanskaya και Wagner πρότειναν (ανεξάρτητα) μια έκδοση του ElGamal στην οποία όλα τα μέρη μοιράζονται το ίδιο modulus. Τέλος ο Hopwood [65] λέει ότι μια παραλλαγή του Bellare, Abdalla και Rogaway DHAES scheme έχει αυτή την ιδιότητα, όπως και μια παραλλαγή του RSA, στην οποία τα moduli παράγονται κοντά το ένα στο άλλο.

Παρασκηνιακά κάθε μερίδιο χρησιμοποιεί το “buddy system” για να επιτύχει απονομή ευθυνών. Όσοι εξυπηρέτες διαγράφουν ένα μερίδιο πριν από τη λήξη του χρόνου παραμονής τους, κάποια στιγμή γίνονται αντιληπτοί και η εμπιστοσύνη προς αυτούς μειώνεται. Κάθε εξυπηρέτης διατηρεί μια βάση δεδομένων, στην οποία αποθηκεύονται το δημόσιο κλειδί, το remailer block και η υπόληψη (βασιζόμενος στην πρότερη εμπειρία καθώς στο τι έχουν πει άλλοι) άλλων εξυπηρετών. Η επικοινωνία ανάμεσα στους εξυπηρέτες και του servnet και των αναγνωστών παρέχεται πάνω από κάποια υποδομή ανώνυμης επικοινωνίας, τύπου mixnet [66].

Εμπιστευτικότητα:

Εάν ένας εκδότης επιθυμεί να αποκρύψει ένα έγγραφο μπορεί να το κρυπτογραφήσει, πριν την εισαγωγή του. Ενώ ένα τέτοιο σενάριο είναι εφικτό ωστόσο το σύστημα δε προσφέρει κάποια ιδιαίτερη υπηρεσία εμπιστευτικότητας (διαχείριση μυστικών κλειδιών,…), ούτε δίνει συγκεκριμένες προδιαγραφές (χρήση συγκεκριμένων κρυπτογραφικών αλγορίθμων,…). Αντίθετα το σύστημα προβλέπει την κρυπτογράφηση στα εξής σημεία:

Κρυπτογράφηση δημοσίου κλειδιού για την επικοινωνία μεταξύ των τμημάτων (modules) που αποτελούν το σύστημα (haven module, communication module, trust module, crypto module, user interface και το ίδιο το κανάλι επικοινωνίας) με τη χρήση ασύμμετρης κρυπτογραφίας.

Κρυπτογράφηση δημοσίου κλειδιού για την επικοινωνία μεταξύ της αναγνώστριας και των εξυπηρετών που φυλάνε τα αντίστοιχα μερίδια. Αυτή η κρυπτογράφηση περιλαμβάνει τη κρυπτογράφηση των δεδομένων που αποστέλλονται στην αναγνώστρια (παροχή εμπιστευτικότητα) καθώς και τη διαστρωματωμένη κρυπτογράφηση (layered

encryption) ή onion routing που απαιτεί το ανώνυμο δίκτυο επικοινωνίας (Chaumian mix-net).

Ακεραιότητα & Αυθεντικοποίηση:

Χρησιμοποιούνται ψηφιακές υπογραφές για να ελεγχθεί η εγκυρότητα των ανταλλασσόμενων δεδομένων. Βέβαια, η αναγνώστης θα πρέπει να εμπιστευτεί τη πηγή ότι το $PK_{αρχείου}$ που της έδωσε, αντιστοιχεί πράγματι σε αυτό που εκείνη αναζητά. Από εκείνο το σημείο και μετά μπορεί να είναι σίγουρη ότι είτε θα πάρει το ακριβές έγγραφο που αντιστοιχεί στο δημόσιο κλειδί, είτε ότι δε θα πάρει τίποτα (εάν πολλοί εξυπηρέτες αποφασίσουν να σπάσουν το συμβόλαιό τους).

Διαθεσιμότητα:

Μια από τις λειτουργικές απαιτήσεις του πρωτοκόλλου είναι να εξασφαλίζεται η διαθεσιμότητα ενός κειμένου για όλο το χρονικό διάστημα που ορίζει ο εκδότης του. Αυτό επιτυγχάνεται με την υπογραφή συμβολαίων μεταξύ των εξυπηρετών, τη χρήση του αλγορίθμου διασποράς πληροφορίας του Rabin και την ύπαρξη πλεονασμού στα μερίδια.

Ανωνυμία:

Ο κύριος σκοπός του Free Haven είναι να προσφέρει ισχυρή ανωνυμία σε όλα τα συμβαλλόμενα μέρη: στους εκδότες που εισάγουν έγγραφα στο δίκτυο, στους αναγνώστες που κατεβάζουν αυτά και στους εξυπηρέτες που τα φυλλάσουν, αλλά και στα ίδια τα έγγραφα (ο εξυπηρέτης δε ξέρει τι περιεχόμενο φυλάει). Για να επιτευχθεί η ανωνυμία αυτή είναι απαραίτητη η ύπαρξη μιας ιδανικής υποδομής ανώνυμης επικοινωνίας, η οποία δυστυχώς δεν υφίσταται. Οι σχεδιαστές του δικτύου προσπαθούν να αναπτύξουν δύο υποδομές ανώνυμης επικοινωνίας τη Mixminion [67] και την Onion Routing δεύτερης γενιάς [68], η τελευταία προσφέρει υπηρεσίες λιγότερο ισχυρής ανωνυμίας αλλά εισάγει λιγότερες καθυστερήσεις στην επικοινωνία.

Απονομή Ευθυνών (Accountability):

Ο κάθε εξυπηρέτης διατηρεί μια βάση δεδομένων, όπου πέραν των άλλων κρατείται και η άποψη του για την εμπιστοσύνη άλλων εξυπηρετών. Περιοδικά κάθε εξυπηρέτης διαλέγει έναν άλλο εξυπηρέτη από τη βάση και κάνει broadcast την άποψη του για την αξιοπιστία του. Μέσω αυτού του κουτσομπολιού οι εξυπηρέτες μένουν ενήμεροι για πράξεις που δεν τους επηρέασαν άμεσα. Κάθε εξυπηρέτης που λαμβάνει αυτή τη πληροφόρηση την ερμηνεύει με βάση τη πληροφορία που ήδη διαθέτει σχετικά με τον αποστολέα της πληροφορίας και το υποκείμενο αυτής. Πέραν από την άποψη των

εξυπηρετών και τα ίδια τα μερίδια είναι υπεύθυνα για την απονομή ευθυνών στο δίκτυο. Μέσα από το την υπηρεσία buddy check ένα μερίδιο περιοδικά ελέγχει για το αντίγραφό του εκκινώντας μια διαδικασία απολύτως όμοια με αυτήν της ανάκτησης (ώστε ο υπό έλεγχο εξυπηρέτης να μη μπορεί να διαφοροποιεί τη συμπεριφορά του). Εάν ένα μερίδιο συμπεράνει ότι το αντίγραφό του δεν είναι πια διαθέσιμο κάνει ένα buddy broadcast, που περιέχει την άποψή του για τον εξυπηρέτη που φύλασσε το αντίγραφο. Κάθε εξυπηρέτης που λαμβάνει αυτή τη πληροφόρηση την ερμηνεύει με βάση τη πληροφορία που ήδη διαθέτει σχετικά με τον αποστολέα της πληροφορίας και το υποκείμενο αυτής. Το παρεχόμενο σύστημα υπόληψης των εξυπηρετών τύπου κουτσομπολιού/αξιοπιστίας (gossip/credibility), δεν είναι λειτουργικό και αναμένεται να αντικαταστεί από ένα σύστημα επαληθεύσιμων συναλλαγών (verifiable transactions). Μια άλλη προσέγγιση υποστηρίζει την εξασθένιση του συστήματος υπόληψης. Συμφωνά με αυτήν κάθε εξυπηρέτης χρησιμοποιεί μόνο τοπική πληροφορία για τις συναλλαγές και ανταλλαγές του. Συνέπεια αυτού είναι ότι κάθε εξυπηρέτης θα πρέπει μόνος του να αποδείξει σε κάθε άλλον εξυπηρέτη που συναντά ότι είναι αξιόπιστος, ούτως ώστε να μπορέσει να εκδώσει κείμενα. Αυτή η μη διάδοση νέων, σχετικών με τη συμπεριφορά των εξυπηρετών, αναγκάζει τα μέλη του συστήματος να γίνουν πιο συντηρητικά ως προς τη παροχή πόρων σε καινούς κόμβους· ταυτόχρονα, για να μη μείνει στάσιμο το δίκτυο ζητείται από τα υφιστάμενα μέλη να διακινδυνεύσουν του πόρους τους προκειμένου να εισέλθουν νέοι κόμβοι. Αυτό όμως ανοίγει το δρόμο σε έναν κακόβουλο χρήστη να προκαλέσει ζημιά (σπαταλήσει τους πόρους) σε όλους τους κόμβους από μια φορά.

2.5 distributed.net

Υβριδικό Ομότιμο δίκτυο Υπολογιστικής Κοινών Πόρων

Σύντομη Περιγραφή

Το distributed.net [2] ήταν το πρώτο πρόγραμμα κατανεμημένης υπολογιστικής πάνω από το Internet. Αρχή γινομένης από το 1997, το δίκτυο μετράει σήμερα χιλιάδες χρήστες ανά τον κόσμο, που διαθέτουν υπολογιστικούς πόρους ακαδημαϊκών αλλά και γενικότερου ενδιαφέροντος προγραμμάτων. Το δίκτυο έχει τι τυπική δομή ενός υβριδικού ομότιμου δικτύου υπολογιστικής κοινών πόρων. Ο χρήστης εγκαθιστά στον υπολογιστή του ένα λογισμικό πελάτη που επικοινωνεί με το κεντρικό δίκτυο εξυπηρετών του distributed.net για να κατεβάσει μονάδες δεδομένων (data blocks) προς επεξεργασία, και να ανεβάσει το αποτέλεσμα των υπολογισμών του. Η μέθοδος που χρησιμοποιεί το δίκτυο για να λύσει τα προβλήματά και τις προκλήσεις που αναλαμβάνει είναι η εξαντλητική αναζήτηση (brute-force attack).

Προγράμματα

Δίνεται μια συνοπτική παρουσίαση των προβλημάτων και των προκλήσεων (για τις προκλήσεις βλ. RSA Labs, http://www.rsasecurity.com/rsalabs/) που απαντήθηκαν με τη χρήση του distributred.net:

RC5-56:

Η πρόκληση των RSA Labs RC5-56 τελείωσε στις 22 Οκτωβρίου του 1997, μετά από αναζήτηση 250 ημερών, όταν ανακοινώθηκε ότι βρέθηκε το κλειδί 0x532B744CC20999 που έδινε το κείμενο "The unknown message is: It's time to move to a longer key length".

DES II-1:

Η πρόκληση DES II-1 των RSA Labs είχε χρονικό περιορισμό. Ξεκίνησε το στις 13 Ιανουαρίου 1998 και τελείωσε στις 24 Φεβρουάριου του ίδιου έτους. Το κλειδί ήταν το 76 9E 8C D9 F2 2F 5D EA και το μήνυμα έλεγε "The unknown message is: Many hands make light work."

DES II-2:

Άλλη μια πρόκληση με χρονικό περιορισμό. Νικητής, όμως, ήταν η Electronic Frontier Foundation (EEF) που έκανε χρήση εξειδικευμένου υλικού (custom made).

DES-III

Ξεκίνησε στις 13 Ιανουάριου 1999, και έληξε επιτυχώς μετά από 22,5 ώρες με τη βοήθεια του Deep Crack, μιας μηχανής ειδικού σκοπού της EEF.

Πρόκληση CS-Cipher:

Η CS-Cipher Challenge οργανώθηκε από τη CS Communications & Systems και προοριζόταν να διαρκέσει ένα έτος. Το distributed.net βρήκε το κλειδί αποκρυπτογράφησης στις 16 Ιανουάριου 2000, αφού πρώτα έλεγξε 98% των πιθανών κλειδιών σε λιγότερο από 2 μήνες. Σκοπός της πρόκλησης ήταν να δειχθεί η αδυναμία ενός κλειδιού 56-bit απέναντι σε μια επίθεση εξαντλητικής αναζήτησης.

RC5-64:

Υστέρα από 1757 μέρες αναζήτησης βρέθηκε στις 14 Ιουλίου 2002 το κλειδί της πρόκλησης RC5-64. το κλειδί ήταν το 0x63DE7DC154F4D039, είχε βρεθεί από έναν ομότιμο χρήστη που είχε έναν Pentium3-450 στην Ιαπωνία με λειτουργικό Windows 2000®, και έδινε το κείμενο "The unknown message is: Some things are better left unread" had been found.

Optimal 24 Mark Golomb Rulers – Βέλτιστος Golomb Κανών με 24 σημεία:

Το OGR-24 πρόγραμμα τερματίστηκε την 1 Νοεμβρίου 2004, όταν αποδείχτηκε ότι η βέλτιστη λύση είναι η 24/9-24-4-1-59-25-7-11-2-10-39-14-3-44-26-8-40-6-21-15-16-19-22.

Τρέχοντα Προγράμματα

RC5-72:

Στις 3 Δεκεμβρίου 2002 ξεκίνησε η αναζήτηση για το 72-bit κλειδί του κρυπτογραφικού αλγορίθμου RC5.

Αθροιστικά Στατιστικά (12/9/2005)	
Σύνολο blocks προς αναζήτηση:	1,099,511,627,776
Σύνολο ελεγχθέντων blocks:	2,825,936,149
Ρυθμός:	32 Blocks/sec
Σύνολο κλειδιών προς αναζήτηση:	4,722,366,482,869,646,000,000
Σύνολο ελεγχθέντων κλειδιών:	12,137,303,340,539,182,000
Ρυθμός:	138,402,017,658 Keys/sec
Ολοκληρώθηκε:	0.257%
Χρόνος εργασίας:	1,015 days

3,121,338 blocks ελέγχθηκαν στις 11/9/2005 (0.000284% του χώρου κλειδιών)
με ρυθμό 36 Blocks/sec.

13,406,044,629,762,047 κλειδιά ελέγχθηκαν στις 11/9/2005 (0.000284% του χώρου κλειδιών)
με ρυθμό 155,162,553,585 Keys/sec.

Οι πιθανότητες είναι 1 στις 351,351 να βρεθεί το κλειδί
στις επόμενες 24 ώρες, τουτέστιν θα ελεγχθεί το
100% σε 351,351 μέρες.

Συμμετείχαν 66,722 χρήστες
από την αρχή του προγράμματος.
8,791 ήταν ενεργοί στις 11/9/2005
και από αυτούς, 22 ήταν νέοι χρήστες.

Υπάρχουν 4,484 καταχωρημένες ομάδες.
1,440 υπέβαλλαν μονάδες εργασίας στις 11/9/2005

Πίνακας 11. Πρόοδος RC5-72

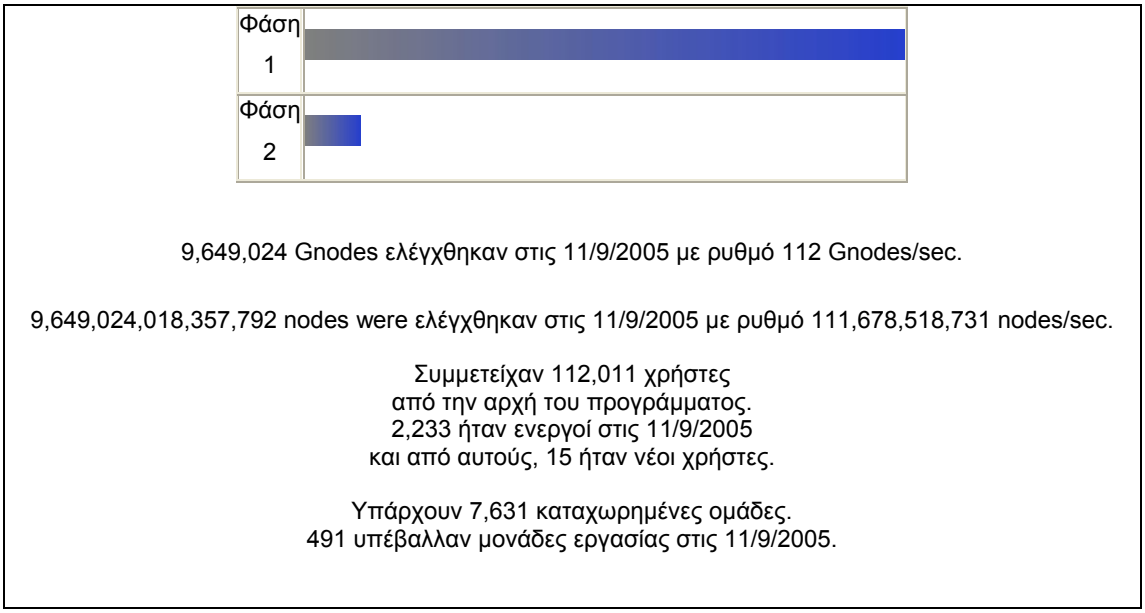
Optimal 25 Mark Golomb Rulers – Βέλτιστος Golomb Κανών με 25 σημεία:

Ένας Golomb Κανών (Golomb Ruler) είναι η μαθηματική ονομασία που δίνεται σε ένα σύνολο μη αρνητικών ακεραίων, στο οποίο δεν υπάρχουν δύο διαφορετικά ζευγάρια αριθμών από το σύνολο που να έχουν την ίδια διαφορά. Εάν το δούμε σχηματικά είναι ένας κανών στον οποίο δεν υπάρχουν δύο διαφορετικά ζευγάρια σημείων που να μετράνε την ίδια απόσταση. Ο Βέλτιστος Golomb Κανών είναι ο μικρότερος δυνατός για το συγκεκριμένο πλήθος σημείων. Οι Κανόνες Golomb πήραν το όνομά τους από τον δρ. Solomon W. Golomb.

Αθροιστικά Στατιστικά (12/9/2005)

Σύνολο ελεγχθέντων Gnodes:	30,715,136,224
Ρυθμός:	190 Gnodes/sec
Σύνολο ελεγχθέντων nodes:	30,715,136,224,408,455,000
Ρυθμός:	190,310,097,354 nodes/sec
Ολοκληρώθηκε (Φάση 1):	100.00%
Ολοκληρώθηκε (Φάση 2):	~9.2%
Time Working:	1,868 days

Progress Meters



Πίνακας 12. Πρόοδος εύρεσης Βέλτιστου Golomb Κανών 25 σημείων

Στα μελλοντικά προβλήματα που θα δοκιμάσει να λύσει το distributed.net περιλαμβάνονται η ανεύρεση δύο πρώτων παραγόντων μεγάλων αριθμών, που επιχορηγείτα από τα RSA Labs. Και η κρυπτανάλυση στο Elliptic Curve Cryptosystem - ECC.

Περιγραφή Πλατφόρμας

Το δίκτυο κατανεμημένης επεξεργασίας distributed.net έχει χτιστεί γύρω από μια πυραμοειδή διάταξη που αποτελείται από keyservers (εξυπηρέτες κλειδιών) και εξυπηρετούμενους. Στη κορυφή βρίσκεται ο κύριος εξυπηρέτης (*master keyserver)*, ο οποίος ελέγχει ποιες μονάδες δεδομένων (κλειδιά για το πρόγραμμα RC5 και στελέχη για το OGR) έχουν σταλεί προς έλεγχο, ποιες έχουν ελεγχθεί και επιστρέψει και ποιες μένουν να ελεγχθούν. Κάτω από τον κύριο εξυπηρέτη κλειδιών είναι οι κεντρικοί *proxy keyservers)*. Αυτοί παίζουν το ρόλο του ενδιαμέσου ανάμεσα στους εξυπηρετούμενους και τον master keyserver. Οι proxy keyservers ζητάνε μονάδες δεδομένων από τον master keyserver. Οι εξυπηρετούμενοι με τη σειρά τους ζητάνε τις μονάδες δεδομένων από τους proxy keyservers. Οι εξυπηρετούμενοι θα επιστρέψουν τις ελεγμένες μονάδες στους proxy keyservers, οι οποίοι θα τις στείλουν στην κορυφή. Με αυτό τον τρόπο πολλοί keyservers μπορούν να κατανέμουν μονάδες δεδομένων χωρίς τον κίνδυνο να στείλουν μια μονάδα περισσότερες φορές από ότι χρειάζεται. Επίσης ο συνδυασμός των proxy keyservers μαζί με ένα round-robin DNS προσδίδει στο σύστημα μεγαλύτερη ανθεκτικότητα στα σφάλματα, καθώς ναι μεν ο master keyserver αποτελεί μοναδικό σημείο αποτυχίας, αλλά δε συμβαίνει το ίδιο με τους υπόλοιπους keyservers.

Προαιρετικά μπορεί να υπάρξει και ένα τρίτο επίπεδο εξυπηρετών οι personal proxy keyservers - *pproxies*. Οι pproxies ζητάνε προκαθορισμένες μονάδες δεδομένων από τους proxy keyservers και αναλαμβάνουν να τις κατανείμουν στους εξυπηρετούμενους. Συνήθως χρησιμοποιούνται για να υπερκεραστούν αναχώματα ασφαλείας ή και από ομάδες εξυπηρετούμενων. Όλα τα μέλη μιας ομάδας λειτουργούν μέσα από έναν pproxy με αποτέλεσμα να έχουν μια μεγαλύτερη ευχέρεια στην συλλογή στατιστικών στοιχείων για την εργασία που έχουν κάνει, πράγμα που ελαφρύνει και το φόρτο εργασίας των proxy keyservers.

Τα προγράμματα του distributed.net έχουν κατά καιρούς πέσει θύματα επιθέσεων. Διακρίνονται δύο είδη επιθέσεων. Στο πρώτα οι επιτιθέμενοι είναι χρήστες ή ομάδες χρηστών που κατασκευάζουν μια τροποποιημένη έκδοση του λογισμικού πελάτη, η οποία επιστρέφει εσφαλμένα ή παραποιημένα αποτελέσματα και προσπαθεί να προσπεράσει τους ελέγχους των εξυπηρετών. Στόχος αυτών συνήθως είναι η απόκτηση υψηλότερης θέσης στη κατάταξη με τους χρήστες που παρέχουν πόρους στο δίκτυο, οπότε και η διαγραφή τους από αυτή λύνει το πρόβλημα. Έχουν υπάρξει και περιπτώσεις ομάδων που υπονόμευσαν την ασφάλεια του δικτύου για να αποδείξουν ότι αυτό είναι εφικτό, γεγονός που το παραδέχονται και οι ίδιοι οι διαχειριστές αλλά δε το θεωρούν σημαντικό. Υπήρξε και μια ομάδα που απειλούσε ότι θα διανείμει ένα τέτοιο τροποποιημένο λογισμικό πελάτη για να πιέσει την εταιρία να ασπαστεί τη φιλοσοφία του ανοικτού κώδικα. Οι σχεδιαστές του δικτύου είναι φειδωλοί όσον αφορά τα αντίμετρα που λαμβάνονται, πάντως υποστηρίζουν ότι η επιστροφή κλειδιών που δεν έχουν ελεγχθεί είναι εντοπίσιμη από αυτόματα και μη φιλτραρίσματα και ότι γίνεται έλεγχος των μοτίβων που έχει η εισροή εσφαλμένων δεδομένων. Ακόμη υποστηρίζεται ότι αυτές οι επιθέσεις αν και προκαλούν καθυστερήσεις και αποδιοργανώνουν το δίκτυο ωστόσο δεν καταλύουν την ακεραιότητα του προγράμματος.

Υπάρχουν τέλος χρήστες, οι οποίοι ορμώμενοι από κάποια αίσθηση κοινοκτημοσύνης, κατασκευάζουν ιούς, δούρειους ίππους, worms και άλλα κακόβουλα προγράμματα που εγκαθιστούν το λογισμικό πελάτη του distributed.net σε ανυποψίαστους χρήστες και διαμοιράζονται, χωρίς την άδειά των τελευταίων, τους υπολογιστικούς πόρους τους. Έχουν καταγραφεί πάνω από 15 τέτοια προγράμματα. Και ενώ μια τέτοια επίθεση φαίνεται να μη προκαλεί κάποια ζημιά στην υποδομή του δικτύου, ωστόσο παρουσιάζει ψευδείς εντυπώσεις για το τι είναι και τι δεν είναι η υπολογιστική κοινών πόρων.

2.6 Berkeley Open Infrastructure for Network Computing - Boinc (πρώην τίτλος SETI@home)

Υβριδικό Ομότιμο δίκτυο Υπολογιστικής Κοινών Πόρων και Κατανεμημένης Αποθήκευσης

Σύντομη Περιγραφή

Όταν πρωτοξεκίνησα τη διπλωματική η πρόθεση μου ήταν να αφιερώσω αυτό το υποκεφάλαιο στο πρόγραμμα SETI@home, που είναι το δημοφιλέστερο παράδειγμα κατανεμημένης επεξεργασίας, έχοντας ξεπεράσει τα 5,4 εκατομμύρια χρήστες σε 226 χώρες, έχοντας συγκεντρώσει πάνω από 2,3 εκατομμύρια χρόνια επεξεργαστικού χρόνου, με ρυθμό 35,22 TeraFLOPs/sec [6 (τελευταίος έλεγχος Δευτέρα 12 Σεπτεμβρίου 2005)].

12/9/2005	**Σύνολο**	**Τελευταίο 24ωρο**
Χρήστες	5436301	0 (new users)
Αποτελέσματα	2005637370	780175
Χρόνος CPU	2378563.061 years	539.796 years
Πράξεις κινητής υποδιαστολής	7.406171e+21	3.042682e+18 (35.22 TeraFLOPs/sec)
Μέσος χρόνος ανά μονάδα εργασίας	10hr 23min 19.7sec	6hr 03min 39.4sec

Πίνακας 13. Στατιστικά SETI@home

Όμως πρόσφατα η ομάδα του SETI@home αποφάσισε να εγκαταλείψει την υπάρχουσα εφαρμογή της και να αξιοποιήσει μια γενικευμένη πλατφόρμα ομότιμης υπολογιστικής. Ως εκ τούτου δε θα αναφερθώ στην ασφάλεια και τα σημεία ευπάθειας των προηγούμενων εφαρμογών του SETI@home αλλά ασχοληθώ με τη νέα αυτή πλατφόρμα.

Το BOINC (Berkeley Open Infrastructure for Network Computing) [69], [70], [71], [72] είναι ένα σύστημα που επιτρέπει την εύκολη δημιουργία και λειτουργία προγραμμάτων υπολογιστικής κοινών πόρων. Αναπτύσσεται στο U.C. Berkeley Spaces Sciences Laboratory από την ίδια ομάδα που σχεδίασε και συνεχίζει να διαχειρίζεται το SETI@home.

Το BOINC υποστηρίζει εφαρμογές με ποικίλες απαιτήσεις, όπως ανάγκη αποθήκευσης ογκωδών δεδομένων και μεταφοράς των. Επιτρέπει το διαμοιρασμό πόρων ανάμεσα σε αυτόνομα προγράμματα, Οι συμμετέχοντες χρήστες των ομότιμων κόμβων έχουν τη δυνατότητα να ορίσουν την κατανομή των πόρων τους ανάμεσα στα διάφορα προγράμματα στα οποίο συμμετέχουν. Και φυσικά παρέχει κίνητρα συμμετοχής στους συμμετέχοντες μέσα από ένα ρωμαλέο σύστημα λογιστικής ανταμοιβών (credit accounting) που επιβραβεύει τους χρήστες, κατατάσσοντάς τους σε ανάλογα με τους πόρους που έχουν παράσχει στα προγράμματα.

Τρέχοντα Προγράμματα

http://Climateprediction.net*:*

Μελετά τις κλιματολογικές αλλαγές.

http://einstein.phys.uwm.edu/:

Ερευνά βαρυτηκά σήματα προερχόμενα από pulsars.

http://athome.web.cern.ch/athome/:

Προσπάθεια βελτίωσης του σχεδιασμού του επιταχυντή σωματιδίων LHC του CERN.

http://predictor.scripps.edu/:

΄Ερευνα για ασθένειες που συνδέονται με πρωτεΐνες.

http://setiweb.ssl.berkeley.edu/:

Αναζήτηση για εξωγήινη νοημοσύνη.

http://www.cellcomputing.net/

Ιατρική έρευνα.

12/9/2005	Μονάδες Ανταμοιβής	Χρήστες	Ομάδες	Κόμβοι	Χώρες
ClimatePrediction.Net	810,286,416	50,174	2,421	96,939	154

http://einstein.phys.uwm.edu/	487,247,933	59,501	3,188	108,096	159
http://predictor.scripps.edu/	131,822,999	28,057	1,751	68,664	137
http://lhcathome.cern.ch/	42,189,396	12,830	1,055	30,741	113
http://setiathome.berkeley.edu/	18,611,892	79,393	24,166	429,607	202

Πίνακας 14. Στατιστικά Προγραμμάτων BOINC

Χαρακτηριστικά Συστήματος

Το BOINC υλοποιεί χαρακτηριστικά που απλοποιούν τη διαδικασία ανάπτυξης και λειτουργίας προγραμμάτων κατανεμημένης υπολογιστικής. Παρέχεται ευελιξία στην ενσωμάτωση υπαρχόντων επιστημονικών εφαρμογών, γραμμένες σε C, C++ ή Fortran, αλλά και στην αναβάθμιση αυτών. Παρόλο που το BOINC βρίσκεται κάτω την Lesser GNU Public License, η επιστημονική εφαρμογή δεν χρειάζεται να είναι ανοικτού κώδικα. Παρέχεται ασφάλεια μέσα από μηχανισμούς αντιμετώπισης συγκεκριμένων επιθέσεων,

όπως για παράδειγμα η χρήση ψηφιακών υπογραφών για την πιστοποίηση της εγκυρότητας του λογισμικού. Υποστηρίζονται πολλαπλοί εξυπηρέτες (πλεονασμός) και άλλοι μηχανισμοί που προσδίδουν ανοχή σε σφάλματα, τουτέστι ένας χρήστης αυτόματα δοκιμάζει εναλλακτικούς εξυπηρέτες, εάν κανένας δε λειτουργεί τότε για να αποφευχθεί η κατάκλυση (flooding) των εξυπηρετών, όταν εκείνοι επαναλειτουργήσουν, οι χρήστες εφαρμόζουν την εκθετική υποχώρηση (exponential backoff). Ακόμα οι χρήστες έχουν την δυνατότητα να καθορίσουν το πλήθος των μονάδων εργασίας (work units) που κατεβάζουν, ούτως ώστε να έχουν δεδομένα προς επεξεργασία ακόμα και όταν οι εξυπηρέτες δε λειτουργούν αλλά και για να μπορούν να προγραμματίζουν καλύτερα το χρόνο των συνδέσεών τους στο Internet.

Το σύστημα BOINC αποτελείται από ένα σύμπλεγμα υλικού και λογισμικού που διαχωρίζεται σε δύο επίπεδα.

Επίπεδο Εξυπηρέτη:

Λογισμικό:

Λογισμικό συστήματος που περιλαμβάνει τις Βάσεις Δεδομένων και την ιστοσελίδα του προγράμματος

Υλικό:

Οι εξυπηρέτες του προγράμματος

Επίπεδο Εξυπηρετούμενου:

Λογισμικό:

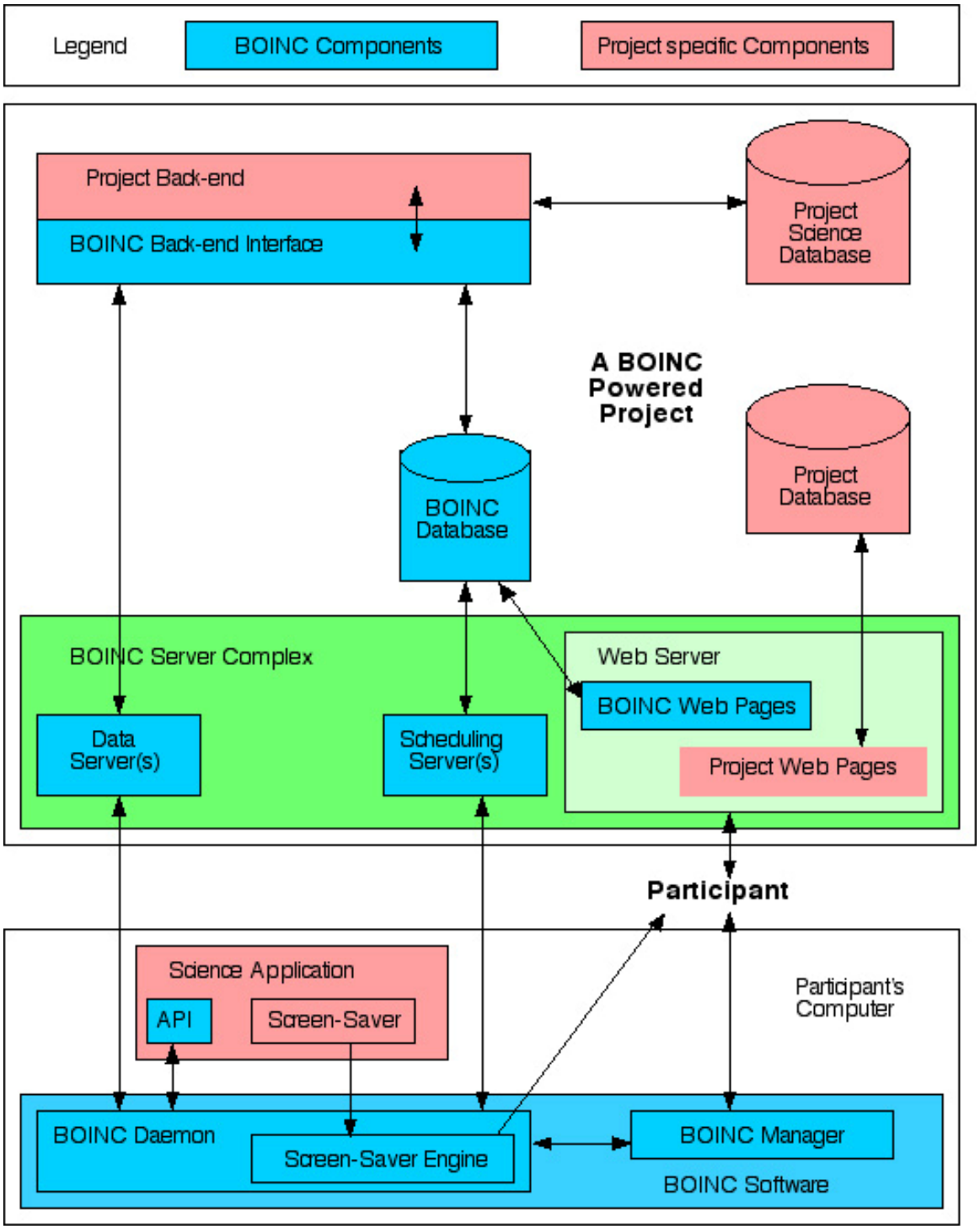
Ο Διαχειριστής BOINC (BOINC Manager)

Ο BOINC Daemon

Η επιστημονική εφαρμογή (Science Applications)

Υλικό:

Οι κόμβοι του συμμετέχοντος χρήστη

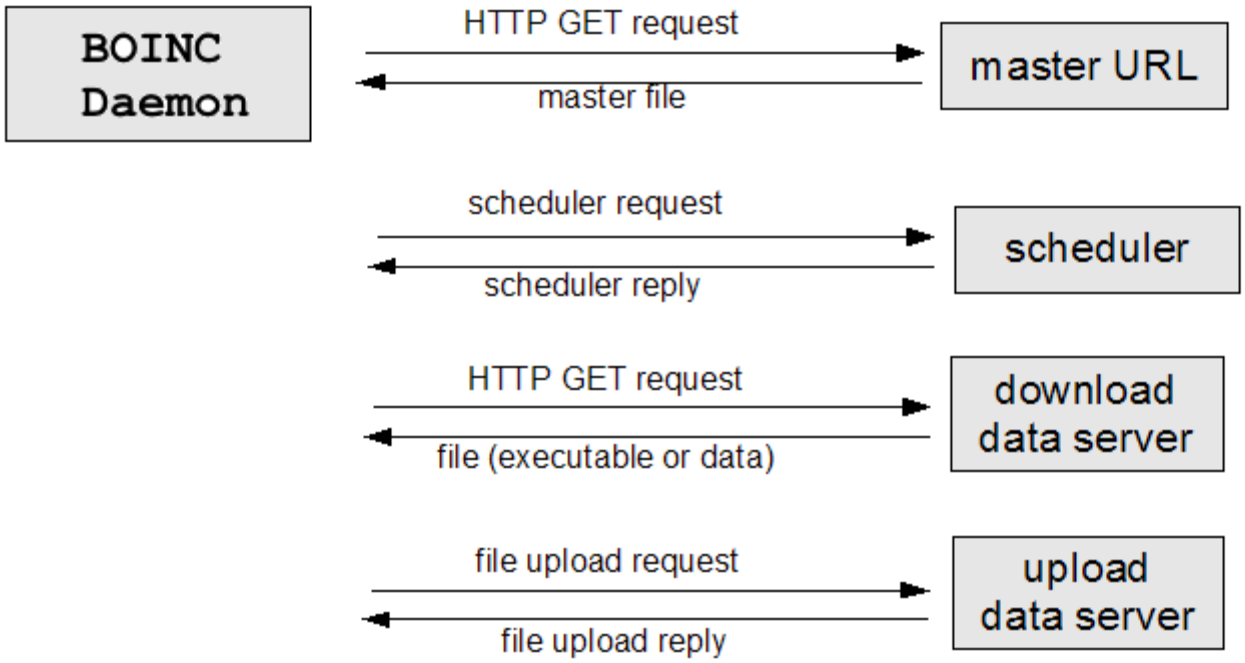


Εικόνα 6. Αρχιτεκτονική BOINC

Το λογισμικό πελάτη του BOINC επικοινωνεί με διάφορους εξυπηρέτες για να κατεβάσει μονάδες εργασίας και να επιστρέψει αποτελέσματα. Όλη η επικοινωνία γίνεται μέσω HTTP στη θύρα 80.

Πιο συγκεκριμένα η επικοινωνία χρήστη-εξυπηρετών περιλαμβάνει τα εξής στάδια. Στην αρχή ο χρήστης κατεβάζει τη σελίδα από το Master URL του

προγράμματος. Η σελίδα περιέχει ετικέτες (tags) σε XML που φέρουν πληροφορία για τα domain names των Schedulers (εξυπηρέτες προγραμματισμού). Ύστερα επιλέγεται ένας Scheduler με τον οποίο ο χρήστης ανταλλάσει αιτήσεις και απαντήσεις, οι απαντήσεις περιέχουν περιγραφές της δουλειάς που πρέπει να γίνει και λίστες με URLs των απαιτούμενων για την εργασία αρχείων. Αφού κατεβάσει αυτά τα αρχεία, που μπορεί να είναι το λογισμικό της επιστημονικής εφαρμογής ή τα δεδομένα των μονάδων εργασίας, από έναν ή περισσότερους εξυπηρέτες δεδομένων (Data Servers), αρχίζει η επεξεργασία. Όταν τελειώσει αυτή, ο χρήστης ανεβάζει το δεδομένα του αποτελέσματος, κάνοντας χρήση ενός ειδικού πρωτοκόλλου του BOINC προστατεύει τους εξυπηρέτες δεδομένων από επιθέσεις άρνησης υπηρεσίας (DOS Attacks). Τέλος ο χρήστης ξαναεπικοινωνεί με έναν scheduler για να δηλώσει την ολοκλήρωση της εργασίας και να ζητήσει επιπλέον μονάδες εργασίας.



Εικόνα 7. Μοντέλο επικοινωνίας BOINC

Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίζουν τα συστήματα υπολογιστικής κοινών πόρων είναι η επιβεβαίωση της εγκυρότητας των μονάδων εργασία και των αποτελεσμάτων. Ένας χρήστης μπορεί μεν να υπολογίσει σωστά και να παραδώσει το αποτέλεσμα, αλλά επίσης ενδέχεται να έχει κάνει λάθος υπολογισμούς ή

να έχει αποτύχει να κατεβάσει ή να ανεβάσει τα δεδομένα ή να έχει σταματήσει να παρέχει πόρους στο σύστημα ή ακόμα ο scheduler να μη μπορεί να βρει κάποιον χρήστη με τους απαιτούμενους πόρους για μια επεξεργασία. Προκειμένου να αντιμετωπίσει αυτά τα προβλήματα το BOINC εφαρμόζει μια μορφή πλεονάζουσας υπολογιστικής (redundant computing), σύμφωνα με την οποία κάθε επεξεργασία λαμβάνει μέρος σε πολλαπλούς χρήστες, τα αποτελέσματα αυτών συγκρίνονται και γίνονται δεκτά μόνον όταν υπάρξει συμφωνία (consensus), ειδάλλως τα δεδομένα ξαναστέλνονται προς επεξεργασία.

Η εγκυρότητα ελέγχεται μέσω συγκρίσεων. Όταν ένας ικανός αριθμός, μια απαρτία, αποτελεσμάτων επιστραφούν, συγκρίνονται μεταξύ τους. Η μέθοδος των συγκρίσεων, που μπορεί να λαμβάνει υπόψη λεπτομέρειες όπως οι διαφορές στην αριθμητική κινητής υποδιαστολής ανάμεσα σε αλλότριες υπολογιστικές πλατφόρμες, και η πολιτική, που ακολουθείται για να βρεθεί συμφωνία, καθορίζονται από την εφαρμογή. Όταν καταλήξει σε συμφωνία, το σύστημα ταξινομεί το αποτέλεσμα ως κανονικό (canonical). Κάθε αποτέλεσμα που θα φτάσει αργότερα θα συγκριθεί με το κανονικό προκειμένου αποφασιστεί εάν θα ανταμειφθεί ο χρήστης. Το σύστημα περιλαμβάνει και ένα μηχανισμό αφομοίωσης (assimilation) που εκτελείται μια φορά για κάθε (διαφορετική) μονάδα εργασίας, ασχέτως του επεξεργαστικού πλεονασμού. Ο μηχανισμός αυτός ειδοποιεί το πρόγραμμα για την επιτυχή ή αποτυχή ολοκλήρωση κάθε μονάδας εργασίας. Εάν μια μονάδα εργασίας ολοκληρωθεί επιτυχώς, ήτοι βρεθεί κανονικό αποτέλεσμα μια συνάρτηση προερχόμενη από το πρόγραμμα καλείται να χειριστεί τη πληροφορία, παραδείγματος χάρη μπορεί να τη καταγράψει σε μια βάση. Ειδάλλως, σε περίπτωση αποτυχίας, η συνάρτηση θα πράξει αναλόγως συμπληρώνοντας μια εγγραφή σε ένα ημερολόγιο, ειδοποιώντας τον διαχειριστή και ούτο καθεξής.

Η αναφορά μου στο BOINC θα κλείσει με μια αναφορά στα αντίμετρα που αυτό (δεν) υλοποιεί απέναντι σε κάποιες από τις κυριότερες επιθέσεις που εφαρμόζονται στα συστήματα κατανεμημένης επεξεργασίας.

Παραποίηση Αποτελεσμάτων:

Μπορεί να εντοπιστεί πιθανοτικά με χρήση πλεονάζουσας υπολογιστικής και επαλήθευσης των αποτελεσμάτων. Ειπώθηκε παραπάνω ότι εάν η πλειοψηφία των αποτελεσμάτων συμφωνεί, σύμφωνα με τους κανόνες που θέτει το κάθε πρόγραμμα, τότε ταξινομούνται ως ορθά.

Παραποίηση Ανταμοιβών:

Και αυτή μπορεί να πιθανοτικά με χρήση πλεονάζουσας υπολογιστικής και επαλήθευσης της ανταμοιβής. Για παράδειγμα σε κάθε συμμετέχοντα δίδεται η ελάχιστη ανταμοιβή από τα ορθά αποτελέσματα.

Διάδοση Κακόβουλων Εκτελέσιμων:

Το BOINC χρησιμοποιεί τη ψηφιακή υπογραφή του πηγαίου κώδικα. Σε κάθε πρόγραμμα αντιστοιχεί ένα ζεύγος δημοσίου και ιδιωτικού κλειδιού για την υπογραφή του κώδικα. Το ιδιωτικό θα πρέπει να κρατείται μυστικό σε έναν υπολογιστή χωρίς πρόσβαση στο Internet και να χρησιμοποιείται μόνο για την παραγωγή ψηφιακών υπογραφών για τα εκτελέσιμα αρχεία. Το δημόσιο διανέμεται και αποθηκεύεται στους κόμβους των χρηστών. Ακόμα και εάν ένας επιτιθέμενος καταφέρει να αποκτήσει πρόσβαση σε κάποιον από τους εξυπηρέτες BOINC του προγράμματος δε θα έχει τρόπο να ξεγελάσει τους χρήστες να δεχτούν κίβδηλο κώδικα.

Εάν για μεγαλύτερη ασφάλεια απαιτείται η περιοδική αλλαγή του ζεύγους κλειδιών, το λογισμικό πελάτης του BOINC καταλαβαίνει και δέχεται νέο δημόσιο κλειδί μόνο εάν αυτό σταλεί ψηφιακά υπογεγραμμένο από το προηγούμενο ιδιωτικό κλειδί.

Άρνηση Υπηρεσίας στους Εξυπηρέτες Δεδομένων:

Κάθε αρχείο αποτελέσματος σχετίζεται με ένα μέγιστο μέγεθος. Κάθε πρόγραμμα έχει ένα ζεύγος δημοσίου και ιδιωτικού κλειδιού για την αυθεντικοποίηση του ανεβάσματος. Το δημόσιο κλειδί αποθηκεύεται στους εξυπηρέτες δεδομένων του προγράμματος. Περιγραφές αρχείων αποτελέσματος στέλνονται στους χρήστες συνημμένες με μια ψηφιακή υπογραφή, η οποία στέλνεται στους εξυπηρέτες δεδομένων όταν το αρχείο ανεβάζεται. Οι εξυπηρέτες δεδομένων επαληθεύουν τη περιγραφή του αρχείου και βεβαιώνουν ότι το μέγεθος των δεδομένων που ανεβάστηκε δε υπερβαίνει το μέγιστο μέγεθος.

Κλοπή Πληροφοριών από τους Λογαριασμούς των Χρηστών:

Η επίθεση αυτή μπορεί να πραγματοποιηθεί είτε με απευθείας επίθεση στον εξυπηρέτη, είτε υποκλέπτοντας τη κίνηση στο δίκτυο. Το BOINC αυτό καθαυτό δεν παρέχει καμία λύση. Οι οργανισμοί που αναπτύσσουν τα προγράμματα είναι υπεύθυνοι για την ασφάλεια, μπορούν να χρησιμοποιήσουν κρυπτογράφηση σε επίπεδο εφαρμογής ή πιο χαμηλά (SSH, IPsec, …), να προστατεύουν του εξυπηρέτες με αναχώματα ασφαλείας (αποστρατιωτικοποιημένες ζώνες, …), να πραγματοποιούν συχνούς ελέγχους ασφάλειας (security audits),...

Κλοπή Αρχείων Προγράμματος:

Τα αρχεία εισόδου/εξόδου που χρησιμοποιούνται από τις εφαρμογές του BOINC δεν κρυπτογραφούνται. Κάτι τέτοιο μπορεί να γίνει σε επίπεδο εφαρμογής από τους ιδιοκτήτες του προγράμματος, αν και από τη στιγμή που τα αρχεία αυτά καταλήγουν στη RAM στην αρχική μη κρυπτογραφημένη μορφή τους, ένας επιτιθέμενος μπορεί να τα αποκτήσει με έναν αποσφαλματωτή (debugger).

Κατάχρηση των κόμβων των χρηστών από τα προγράμματα:

Εσκεμμένη ή κατά τύχη το BOINC πάλι δεν κάνει τίποτα. Οι χρήστες θα πρέπει να εμπιστευτούν τις καλές προθέσεις των ιδιοκτητών των προγραμμάτων και τους ελέγχους που έχουν κάνει.

Εμπιστευτικότητα:

Δεν παρέχεται από το σύστημα. Δε κρυπτογραφούνται ούτε τα αρχεία του προγράμματος, ούτε τα δεδομένα προς επεξεργασία.

Ακεραιότητα & Αυθεντικοποίηση:

Γίνεται χρήση ψηφιακών υπογραφών για την επίτευξη εγκυρότητας στα δεδομένα (αρχεία, μονάδες εργασίας, αποτελέσματα). Ειδικά για τα αποτελέσματα γίνεται χρήση πλεονασμού σε συνδυασμό με πολιτικές συναίνεσης.

Οι χρήστες συνήθως αυθεντικοποιούνται με τη χρήση συνθηματικών, που αποκτούν όταν δημιουργούν λογαριασμούς στα διαφορετικά προγράμματα. Κάθε λογαριασμός αντιστοιχεί σε μια διεύθυνση email.

Διαθεσιμότητα:

Δε παρέχεται κάποια εγγύηση ότι ένα πρόγραμμα θα είναι πάντα σε θέση να παρέχει μονάδες εργασίας ή οποιαδήποτε άλλη υπηρεσία. Υποστηρίζεται, όμως, ο καθορισμός του πλήθους των μονάδων εργασίας που ένας χρήστης μπορεί να κατεβάσει (για να έχει δουλειά όταν ο εξυπηρέτης βγει εκτός λειτουργίας) και επίσης το σύστημα επιτρέπει και προτρέπει τους χρήστες να γραφτούν σε πολλαπλά προγράμματα, έτσι ώστε εάν κάποιο δεν έχει διαθέσιμες μονάδες εργασίας να κατεβάζουν δεδομένα προς επεξεργασία από άλλο πρόγραμμα.

Ανωνυμία:

Δεν παρέχεται από το σύστημα, επαφίεται στους ιδιοκτήτες του προγράμματος να προστατεύσουν τα δεδομένα των λογαριασμών.

2.7 Skype

Υβριδικό Ομότιμο Δίκτυο Άμεσης Επικοινωνίας

Σύντομη Περιγραφή

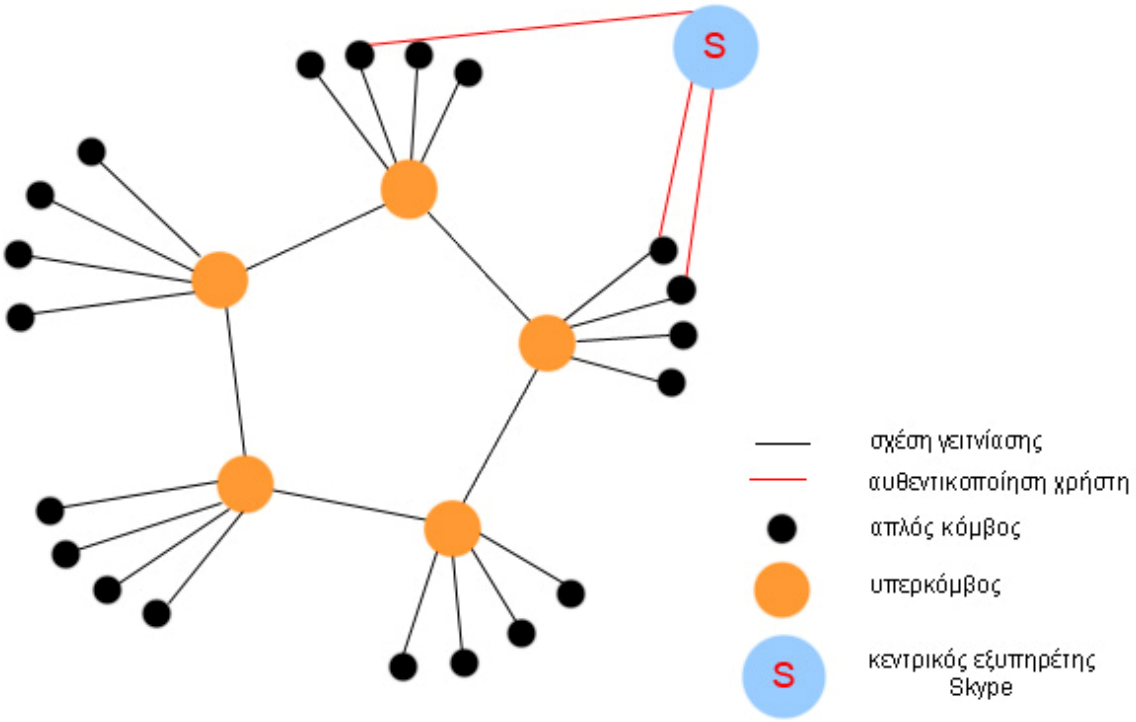
Το Skype [22], [23], [5] είναι ένα ιδιόκτητο ομότιμο δίκτυο παροχής υπηρεσιών τηλεφωνίας πάνω από το Internet (Voice Over IP - VOIP). Το Skype προσφέρει δωρεάν υπηρεσίες ομιλίας από υπολογιστή σε υπολογιστή, άμεσης γραπτής επικοινωνίας για ομάδες μέχρι και 48 ατόμων, συνδιασκέψεων και ανταλλαγής αρχείων, ανάμεσα στους χρήστες του. Καθώς και επί πληρωμή υπηρεσίες κλήσης προς σταθερά και κινητά τηλέφωνα (SkypeOut), λήψης κλήσεων από σταθερά και κινητά τηλέφωνα στον υπολογιστή (SkypeIn) και λήψης μηνυμάτων φωνής (voicemail messages). Οι δημιουργοί του ισχυρίζονται ότι το Skype μπορεί να προσφέρει υπηρεσίες ακόμα και πάνω από αναχώματα ασφαλείας και NATs και ότι η προσφερόμενη ποιότητα είναι ανώτερη των υπαρχόντων εφαρμογών άμεσης επικοινωνίας. Επίσης προσφέρεται αυξημένη ασφάλεια στις επικοινωνίες μέσω κρυπτογράφησης, ψηφιακών υπογραφών και πιστοποιητικών. Το Skype χρησιμοποιεί έναν κεντρικό εξυπηρέτη για τη δημιουργία λογαριασμών, τη παροχή πιστοποιητικών και την αυθεντικοποίηση των χρηστών κατά την σύνδεσή τους στο δίκτυο και gateways για τις επί πληρωμή υπηρεσίες. Όλες οι άλλες υπηρεσίες παρέχονται από τους ομότιμους υπερκόμβους και κόμβους του δικτύου.

Ιστορία

Το Skype αναπτύχθηκε από τους Niklas Zennström και Janus Friis, τους δημιουργούς του Kazaa, το 2003. Τον Οκτώβριο του 2004 είχε ένα εκατομμύριο χρήστες συνδεδεμένους ταυτόχρονα στο δίκτυο. Το Φεβρουάριο του 2005 οι συνδεδεμένοι χρήστες του διπλασιάστηκαν. Το Μάρτιο του ίδιου έτους καταγράφηκαν ένα εκατομμύριο χρήστες τις υπηρεσίας Skype-out σε σύνολο 29 εκατομμυρίων εγγεγραμμένων χρηστών, ενώ είχαν εξυπηρετηθεί 5,98 δισεκατομμύρια λεπτά ομιλίας. Τον Μάιο οι συνδεδεμένοι χρήστες έφτασαν τα τρία εκατομμύρια και τον επόμενο μήνα το πλήθος των εξυπηρετούμενων λεπτών συνομιλίας έφτασε τα δέκα δισεκατομμύρια. Τον Σεπτέμβριο του 2005 το λογισμικό πελάτης του Skype είχε κατεβαστεί πάνω από 162 εκατομμύρια φορές.

Περιγραφή Πρωτοκόλλου

Το Skype είναι ένα υβριδικό ομότιμο δίκτυο. Οι κόμβοι του χωρίζονται σε απλούς και σε υπερκόμβους. Ένας κόμβος είναι ένας υπολογιστής εγγεγραμμένου χρήστη που τρέχει το λογισμικό πελάτη του Skype και που αξιοποιεί τις υπηρεσίες του δικτύου. Ένας υπερκόμβος είναι ένας απλός κόμβος του δικτύου που έχει μια πραγματική IP διεύθυνση (δε βρίσκεται πίσω από NAT) αρκετή επεξεργαστική ισχύ, μνήμη και εύρος σύνδεσης. Κάθε κόμβος δύναται εάν πληρεί κάποια κριτήρια να επιλεγεί για υπερκόμβος, ασχέτως των επιθυμιών του χρήστη. Ένα κόμβος πρέπει να συνδεθεί σε ένα υπερκόμβο και να δηλωθεί (register) στον κεντρικό εξυπηρέτη. Τα ονόματα χρήστη και τα συνθηματικά των χρηστών αποθηκεύονται στον κεντρικό εξυπηρέτη, ο οποίος είναι υπεύθυνος για την αυθεντικοποίηση των κατά την είσοδό τους στο δίκτυο και για τη μοναδικότητα των ονομάτων χρηστή. Δεν υπάρχει άλλος εξυπηρέτης στο δίκτυο Skype, οι πληροφορίες για τους εγγεγραμμένους χρήστες, τα ερωτήματα και τα δεδομένα, αποθηκεύονται ή/και διαδίδονται αποκεντρωμένα στο δίκτυο.

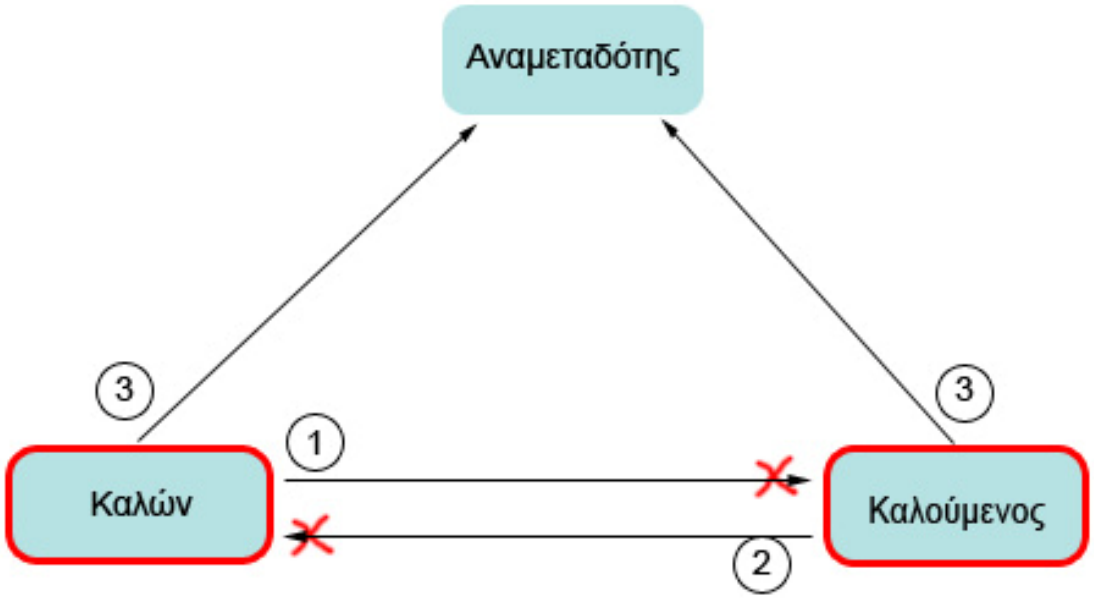


Εικόνα 8. Απεικόνιση δικτύου Skype

Για να συνδεθεί ένας χρήστης θα πρέπει να βρει πρώτα έναν υπερκόμβο, για αυτό το σκοπό χρησιμοποιούνται λίστες με γνωστούς υπερκόμβους. Αφού συνδεθεί με τον υπερκόμβο, ακολουθεί η αυθεντικοποίησή του στον κεντρικό εξυπηρέτη με τη χρήση του ονόματος χρήστη και του συνθηματικού. Όταν ο χρήστης αυθεντικοποιηθεί επιτυχώς ο κεντρικός εξυπηρέτης θα εκδώσει ένα πιστοποιητικό για το δημόσιο κλειδί του χρήστη, υπογεγραμμένο με το δικό του ιδιωτικό κλειδί.

Το Skype χρησιμοποιεί ένα κατάλογο χρηστών, γνωστό ως Global Index, για τις αναζητήσεις. Ο κατάλογος αυτός είναι αποθηκευμένος αποκεντρωμένα στο δίκτυο και η πληροφορία που περιέχει ανανεώνεται δυναμικά. Οι πληροφορίες για ένα συγκεκριμένο χρήστη αποθηκεύονται για 72 ώρες μετά τη τελευταία σύνδεσή του στο δίκτυο.

Όταν ένας χρήστης καλέσει έναν άλλο χρήστη χρησιμοποιεί τους ενεργούς κόμβους του δικτύου προκειμένου να δρομολογήσει τα δεδομένα του. Στην αρχή ο καλών προσπαθεί να επικοινωνήσει με τον καλούμενο άμεσα. Εάν ο καλούμενος βρίσκεται πίσω από ανάχωμα ασφαλείας (ή NAT), μέσω του Global Index ζητείται από τον καλούμενο να επικοινωνήσει με τον καλών. Στην περίπτωση που μια από τις δύο αυτές προσπάθειες επιτύχει η κλήση εγκαθιδρύεται πάνω από μια άμεση σύνδεση. Διαφορετικά το Global Index θα επιδείξει έναν κόμβο που θα λειτουργήσει ως αναμεταδότης και η κλήση θα δρομολογηθεί μέσω αυτού.



Εικόνα 9. Επικοινωνία χρηστών στο Skype

Και στις τρεις περιπτώσεις τα δεδομένα κρυπτογραφούνται από άκρη σε άκρη (end-to-end, e2e) χρησιμοποιώντας ένα μυστικό κλειδί συνόδου 256-bit για τον αλγόριθμο συμμετρικής κρυπτογράφησης AES. Το κλειδί συνόδου ανταλλάσσεται κρυπτογραφημένο με το ιδιωτικό κλειδί του εκκινούντα την επικοινωνία. Το δημόσιο κλειδί αυτού στέλνεται μαζί με το πιστοποιητικό που είχε υπογράψει ο κεντρικός εξυπηρέτης.

Host Caches

Οι Host Caches είναι δυναμικές λίστες που απαρτίζονται από IP διευθύνσεις και θύρες επικοινωνίας υπερκόμβων. Οι λίστες αυτές περιλαμβάνουν εκατοντάδες εισαγωγές και αποθηκεύονται στους κόμβους των χρηστών.

Codecs

Το Skype χρησιμοποιεί για την κωδικοποίηση/αποκωδικοποίηση των δεδομένων φωνής τον iLBC και τον iSAC της GlobalIPSound και έναν τρίτο άγνωστο.

Εμπιστευτικότητα, Ακεραιότητα & Αυθεντικοποίηση:

Η εμπιστευτικότητα, ακεραιότητα και αυθεντικοποίηση των μεταδιδόμενων δεδομένων από κόμβο σε κόμβο, ανάμεσα σε χρήστες του Skype, επιτυγχάνεται μέσω κρυπτογράφησης. Οι κλήσεις προς σταθερά και κινητά τηλέφωνα είναι κρυπτογραφημένες μέχρι να φτάσουν στο Public Switched Telephone Network (PSTN). Τέλος οι κλήσεις από σταθερό ή κινητό τηλέφωνο όπως και οι συνδιασκέψεις όπου ένας από τους συμμετέχοντες βρίσκεται στο PSTN δεν κρυπτογραφούνται.

Το Skype χρησιμοποιεί για την κρυπτογράφηση των ανταλλασσόμενων δεδομένων (φωνή, μηνύματα, αρχεία, ερωτήματα,…) τον συμμετρικό αλγόριθμο AES (Advanced Encryption Standard), επονομαζόμενο και Rijndael με μυστικά κλειδιά συνόδου μεγέθους 256-bit. Το μυστικό κλειδί AES είναι μοναδικό για κάθε σύνοδο και δεν διατηρείται ούτε από το χρήστη, ούτε από κάποια τρίτη οντότητα ως εχέγγυο (key escrow).

Για τη διανομή των κλειδιών αξιοποιείται ψηφιακός φάκελος (digital envelope) με ασύμμετρο κρυπτογραφικό αλγόριθμο RSA και μέγεθος κλειδιού τα 1024-bit. Σε κάθε χρήστη αντιστοιχεί ένα ζεύγος δημοσίου και ιδιωτικού κλειδιού 1024-bit, όμως, περισσότερες λεπτομέρειες για τη παραγωγή και το χρόνο ζωής αυτού ζεύγους δεν δίνονται, και επίσης τα 1024-bit κρίνονται ανεπαρκή ή τουλάχιστον δυσανάλογα λίγα σε σχέση με τα 256-bit του κλειδιού συνόδου. Τα δημόσια κλειδιά των χρηστών αυθεντικοποιούνται με τη χρήση RSA πιστοποιητικών που υπογράφονται από τον

κεντρικό εξυπηρέτη, στην περίπτωση υπηρεσιών επί πληρωμή ο εξυπηρέτης χρησιμοποιεί ιδιωτικό κλειδί 2048-bit για την υπογραφή των πιστοποιητικών, ειδάλλως το κλειδί έχει μήκος 1536-bit. Τα πιστοποιητικά αυτά έχουν μικρό χρόνο ζωής και εκδίδονται κατά την είσοδο του χρήστη στο δίκτυο.

Ειδικά για την αυθεντικοποίηση χρήστη το δίκτυο δεν κάνει υποθέσεις σχετικά με την πραγματική του ταυτότητα. Επίσης οι πληροφορίες χρήστη για όσο διάστημα κρατούνται στο δίκτυο είναι ψηφιακά υπογεγραμμένες από τον κεντρικό εξυπηρέτη.

Διαθεσιμότητα:

Για να μπορέσει να χρησιμοποιήσει ένας χρήστης τις υπηρεσίες του Skype αρκεί να έχει στη host cache του τουλάχιστον ένα εν λειτουργία υπερκόμβο. Στη συνέχεια ο χρήστης πρέπει να αυθεντικοποιηθεί στο κεντρικού εξυπηρέτη. Η εταιρία του Skype βεβαιώνει ότι η διαθεσιμότητα του δικτύου είναι πρωτεύον μέλημά της.

Ανωνυμία:

Όταν ένας χρήστης συνδεθεί για πρώτη φορά στο δίκτυο, πρέπει να επικοινωνήσει με τον κεντρικό εξυπηρέτη και να ανοίξει ένα λογαριασμό με ένα μοναδικό όνομα χρήστη και ένα συνθηματικό. Προκειμένου να διευκολυνθεί η αναζήτηση χρηστών στο κατάλογο του Skype, ζητείται προαιρετικά από το χρήστη να δώσει επιπλέον πληροφορία για τον εαυτό του, όπως το email του. Για να αποτραπεί η δυνατότητα σύνταξης λιστών με τα email των χρηστών, στον κατάλογο αποθηκεύεται μια σύνοψη του email, ούτως ώστε μόνο χρήστες που ήδη ξέρουν ολόκληρο το email να μπορούν να το αναζητήσουν, καθώς ερωτήματα με χαρακτήρες μπαλαντέρ δεν έχουν νόημα. Πέραν των άλλων οι πληροφορίες χρηστών δεν μένουν μόνιμα, κάθε πληροφορία, με εξαίρεση το όνομα χρήστη, κρατείται για περίπου 72 ώρες από τη τελευταία χρήση των υπηρεσιών του δικτύου.

2.8 Groove

Ομότιμο Δίκτυο Κατανεμημένης Συνεργασίας

Σύντομη Περιγραφή

Το Groove [24], [7] είναι μια πλατφόρμα πάνω στο Internet που παρέχει βασική υποστήριξη για συνεργασία, ειδικά στους τομείς της ασφάλειας και του συγχρονισμού. Χρησιμοποιώντας το Groove ομάδες συνεργατών μπορούν να δημιουργήσουν στιγμιαίους διαμοιραζομένους χώρους (spontaneous shared spaces), μέσα στους οποίους συλλέγονται έγγραφα, μηνύματα, εφαρμογές και δεδομένα εφαρμογών που σχετίζονται με την εργασία της ομάδας. Το underlay δίκτυο οποιαδήποτε μορφή και να έχει (LAN, αναχώματα ασφαλείας, NATs, δυναμικές διευθύνσεις IP, …) παραμένει διαφανές για τους χρήστες. Τα κυριότερα χαρακτηριστικά των διαμοιραζόμενων χώρων είναι η *στιγμαιότητά* τους, δεν χρειάζονται διαχειριστές ή πρότερη προετοιμασία, η *ασφάλειά* τους, μπορούν να παρομοιαστούν με στιγμιαία virtual private networks – VPNs, το *περιεχόμενό* τους, ο διαμοιραζόμενος χώρος παρέχει επεξηγηματικό περιεχόμενο στους χρήστες σχετικά με τη φύση, το σκοπό και την ιστορία των μηνυμάτων και εγγράφων, ο *συγχρονισμός* τους, οι διαμοιραζόμενοι χώροι συγχρονίζονται αυτόματα ανάμεσα σε όλες τις Groove συσκευές (υπολογιστικές συσκευές στις οποίες τρέχει λογισμικό του Groove) που ανήκουν σε κάθε μέλος, εάν κάποιος χρήστης βρίσκεται εκτός δικτύου οι αλλαγές θα συγχρονιστούν όταν ξανασυνδεθεί, και επιπλέον το Groove παρέχει τη δυνατότητα, αντί να ανταλλάσουν ολόκληρα έγγραφα, οι χρήστες να ενημερώνονται μόνο για τις τελευταίες αλλαγές που έγιναν.

Περιγραφή Πλατφόρμας

Σε αυτή την ενότητα θα περιγράψω κυρίως τα χαρακτηριστικά ασφάλειας του Groove, που αποτελούν και το πιο ενδιαφέρον στοιχείο της εφαρμογής. Το Groove εξ ορισμού χρησιμοποιεί κρυπτογραφικές μεθόδους για να επιτύχει την εμπιστευτικότητα (συμμετρική και ασύμμετρη κρυπτογράφηση, ψηφιακοί φάκελοι), ακεραιότητα (Message Authentication Code – MAC, ψηφιακές υπογραφής) και αυθεντικοποίηση των δεδομένων (ανταλλασσόμενων ή αποθηκευμένων) και την ταυτοποίηση των χρηστών (ψηφιακές υπογραφές), και δεν επιτρέπει την απενεργοποίηση αυτών των μηχανισμών.

Ένας διαμοιραζόμενος χώρος Groove είναι ένα αντίγραφο μιας αποθήκης αντικειμένου XML (XML object store). Οι επαυξητικές (incremental) αλλαγές στα

αντικείμενα μεταδίδονται σε όλες τις Groove συσκευές, οι οποίες μετέχουν στο διαμοιραζόμενο χώρο), με τη μορφή Groove-δέλτα μηνυμάτων (Groove delta messages). Αυτά τα μηνύματα μπορεί να περιέχουν δεδομένα εφαρμογών (πχ. προσθήκη ενός γραφήματος σε ένα έγγραφο, μια γραμμή σε μια γραπτή επικοινωνία – chat, …) ή διαχειριστικά δεδομένα (πχ. προσκλήσεις συμμετοχής, κρυπτογραφικά κλειδιά, …). Η κατανεμημένη μηχανή επικοινωνίας βεβαιώνει ότι τα δέλτα μηνύματα θα φτάσουν σε όλους τους παραλήπτες· στις περιπτώσεις που κάποια συσκευή είναι χωρίς σύνδεση ή πίσω από κάποιο ανάχωμα ασφαλείας ενδέχεται να απαιτηθεί η χρήση ενός κεντρικού εξυπηρέτη αναμετάδοσης (relay). Πάντως το μόνο που βλέπουν οι χρήστες είναι ένας αυτόματος συγχρονισμός, οι μηχανισμοί είναι αδιαφανείς.

Η προσφερόμενη ασφάλεια αν και δε μπορεί να απενεργοποιηθεί εντελώς, όπως ήδη αναφέρθηκε, ωστόσο δύναται να διαφέρει ανάμεσα σε δύο διαμοιραζόμενους χώρους. Ο ένας λόγος είναι ότι οι χρήστες μπορούν να συμφωνήσουν να χρησιμοποιήσουν διαφορετικούς κρυπτογραφικούς αλγορίθμους και μήκη κλειδιών από τα προεπιλεγμένα. Ο άλλος έχει να κάνει με τη δυνατότητα ενός διαμοιραζόμενου χώρου να λειτουργεί σε δύο διακριτές καταστάσεις, τη κατάσταση αμοιβαίας εμπιστοσύνης (mutual trust), στην οποία τα μηνύματα αυθεντικοποιούνται ότι προήλθαν από κάποιο αόριστο μέλος του διαμοιραζόμενου χώρου, και αμοιβαίας δυπιστίας (mutual suspicion), στην οποία τα μηνύματα αυθεντικοποιούνται ότι προήλθαν από ένα συγκεκριμένο μέλος του διαμοιραζόμενου χώρου. Όσον αφορά τους προεπιλεγμένους κρυπτογραφικούς αλγορίθμους αυτοί είναι για την αυθεντικοποίηση και την ανταλλαγή κλειδιών (ψηφιακός φάκελος) ο ασύμμετρος κρυπταλγόριθμος ελλειπτικής καμπύλης ElGamal με modulus των 1536-bit και για την κρυπτογράφηση κατά την ανταλλαγή ο MARC4 με κλειδί 192-bit (MARC4 είναι ο Modified-Alleged-RC4. το Alleged αναφέρεται σε ένα ελεύθερα παρεχόμενο αλγόριθμο που είναι συμβατός με τον RC4 της RSA Security Inc., ενώ το Modified υποδηλώνει ότι τα πρώτα 256 bytes του keystream δεν χρησιμοποιούνται, για να αποτραπεί μια αλωσιμότητα του αλγορίθμου που παρήγαγε αδύναμα κλειδιά).

Διαμοιραζόμενοι χώροι αμοιβαίας εμπιστοσύνης

Κάθε χρήστης διατηρεί ένα λογαριασμό σε μια ή περισσότερες συσκευές. Ένας λογαριασμός είναι ένας φορέας ταυτότητας, και ένας χρήστης μπορεί να επιλέξει να έχει διαφορετικές ταυτότητες σε διαφορετικούς διαμοιραζόμενους χώρους. Κάθε ταυτότητα ορίζεται από δύο ζεύγη δημοσίου-ιδιωτικού κλειδιού, το ένα για τη ψηφιακή υπογραφή, ονομαζόμενα και κλειδιά ταυτότητας (identity keys), και το άλλο για την κρυπτογράφηση/αποκρυπτογράφηση. Τα δύο ιδιωτικά κλειδιά φυλλάσονται στον λογαριασμό, ενώ τα δύο δημόσια στον διαμοιραζόμενο χώρο, ενώ πιθανώς να υπάρχουν και σε λίστες επαφών, για να μπορούν να αυθεντικοποιούνται μηνύματα

γραπτών επικοινωνιών που λαμβάνουν χώρα έξω από το πλαίσιο ενός διαμοιραζόμενου χώρου. Στην περίπτωση αυτή τα μηνύματα κρυπτογραφούνται με μυστικά κλειδιά συνόδου, τα οποία ανταλλάσσονται με χρήση ψηφιακών φακέλων και των κλειδιών ταυτότητας. Σε αντίθεση τα δέλτα μηνύματα κρυπτογραφούνται με ένα μυστικό κλειδί που βρίσκεται αποθηκευμένο στο διαμοιραζόμενο χώρο.

Όταν ένα μέλος εισέρχεται ή εξέρχεται από ένα διαμοιραζόμενο χώρο χρησιμοποιείται το ζεύγος κλειδιών ταυτότητας για την αυθεντικοποίηση του μηνύματος πρόσκλησης ή απομάκρυνσης. Το ζεύγος κρυπτογράφησης/αποκρυπτογράφησης χρησιμοποιείται για την κρυπτογράφηση/αποκρυπτογράφηση του μυστικού κλειδιού, το οποίο με τη σειρά του χρησιμοποιείται για την κρυπτογράφηση/αποκρυπτογράφηση του μηνύματος πρόσκλησης ή απομάκρυνσης.

Όταν ο Bob κάνει μια αλλαγή σε ένα αντικείμενο του διαμοιραζόμενου χώρου, το δέλτα μήνυμα αποθηκεύεται στη κρυπτογραφημένη αποθήκη αντικειμένων στον δίσκο του και ταυτόχρονα στέλνεται, πάντα κρυπτογραφημένο, στις υπόλοιπες κρυπτογραφημένες αποθήκες αντικειμένων, μέσα από το δίκτυο. Ενώ ασφαλώς εγγυάται και η ακεραιότητα του μηνύματος με τη χρήση Message Authentication Code – MAC [45]. Για τη κρυπτογράφηση των αρχείων στο δίσκο χρησιμοποιείται ένα ανά χρήστη, ανά διαμοιραζόμενο χώρο μυστικό κλειδί. Το κλειδί αυτό προστατεύεται από το κύριο μυστικό κλειδί (master secret key), το οποίο αποθηκεύεται στο λογαριασμό του χρήστη, δημιουργημένο τη στιγμή που εγκαταστάθηκε το λογισμικό του Groove, ο λογαριασμός με τη σειρά του προστατεύεται με ένα συνθηματικό που επέλεξε ο χρήστης. Για τη κρυπτογράφηση των δέλτα μηνυμάτων που στέλνονται πάνω από το δίκτυο χρησιμοποιούνται δύο μυστικά κλειδιά, που αναφέρονται ως ομαδικά κλειδιά (group keys), και βρίσκονται αποθηκευμένα στο διαμοιραζόμενο χώρο και είναι προσβάσιμα από όλα τα μέλη. Το ένα είναι ένα MAC κλειδί, το L_G, για τη προστασία της ακεραιότητας και της αυθεντικοποίησης και το άλλο το K_G, κρυπτογραφεί/αποκρυπτογραφεί τα μηνύματα (Το G αναφέρεται στο σύνολο των μελών της ομάδα – Group του διαμοιραζόμενου χώρου). Ο προεπιλεγμένος αλγόριθμος MAC είναι ο HMAC-SHA1 (η συνάρτηση σύνοψης SHA-1 [73], [45] χρησιμοποιείται για να παραχθεί η σύνοψη της επικεφαλίδας και του σώματος του μηνύματος και ο HMAC [74], [45] προσφέρει αυθεντικοποίηση και ακεραιτότητα στην σύνοψη).

Η ύπαρξη κοινών ομαδικών κλειδιών, ήτοι κοινών L_G και K_G, επιτρέπει μεν σε οποιοδήποτε μέλος του διαμοιραζόμενου χώρου να επιβεβαιώσει την ακεραιότητα και την Αυθεντικοποίηση του δέλτα μηνύματος, στη πραγματικότητα, όμως αυτό που επιβεβαιώνεται είναι ότι κανείς έξω από την ομάδα δεν έχει παραποιήσει το μήνυμα. Δε προστατευόμαστε από επιθέσεις εκ των έσω, καθώς δε μπορούμε να

αυθεντικοποιήσουμε τον ακριβή αποστολέα. Απλώς όλα τα μέλη συμφωνούν να εμπιστεύεται το ένα το άλλο.

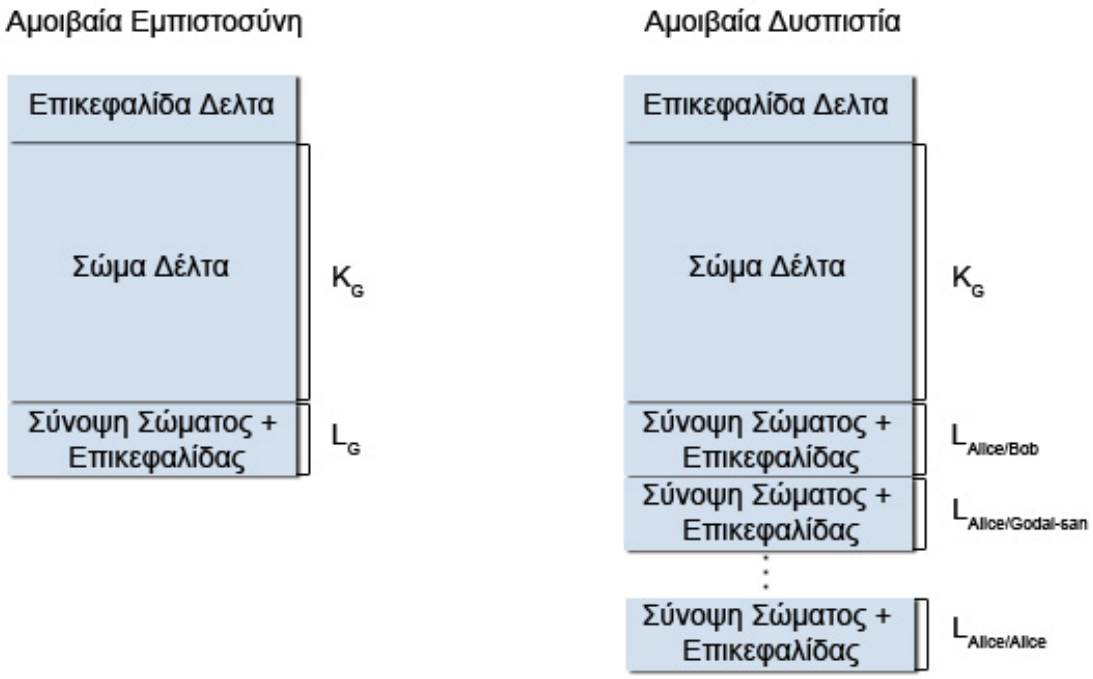
Διαμοιραζόμενοι χώροι αμοιβαίας δυσπιστίας

Υπάρχουν περιπτώσεις όπου η αμοιβαία εμπιστοσύνη δε μπορεί να θεωρηθεί δεδομένη, όπως σε μια συνεργασία ανάμεσα σε άτομα από διαφορετικές εταιρίες, όπου συμβαίνουν οικονομικές συναλλαγές. Για αυτές τις περιπτώσεις το Groove προσφέρει διαμοιραζόμενους χώρους, όπου κάθε μέλος θεωρείται ένας εν δυνάμει επιτιθέμενος και άρα πρέπει να εγγυάται η μη αποποίηση των μηνυμάτων (των αλλαγών που το κάθε μέλος επιφέρει στο διαμοιραζόμενο χώρο). Προκειμένου να μπορέσει να ικανοποιήσει τις επιπλέον απαιτήσεις ασφάλειας το Groove εισάγει ορισμένα καινούρια κλειδιά.

Κάθε μέλος στο διαμοιραζομενο χώρο έχει ένα επιπλέον ζέυγος Diffie-Hellman δημόσιου-ιδιωτικού κλειδιού. Αυτά τα Diffie-Hellman ζεύγη, που αυθεντικοποιούνται με χρήση του ζεύγους κλειδιών ταυτότητας, χρησιμοποιούνται για την εγκαθίδρυση μυστικών κλειδιών ανά ζεύγος χρηστών (pairwise secret keys). Χάρη στον αλγόριθμο των Diffie-Hellman τα κλειδιά δεν χρειάζεται να σταλούν μέσα από το δίκτυο, αλλά αντίθετα παράγονται ξεχωριστά από κάθε χρήστη. Ο Bob υπολογίζει τα μυστικά κλειδιά Bob/Alice για τον ίδιο και την Alice από το ιδιωτικό Diffie-Hellman κλειδί του και το δημόσιο Diffie-Hellman κλειδί της Alice. Και αντιστοίχως πράττει η Alice. Και επίσης υπολογίζει τα μυστικά κλειδιά Bob/Godai-san και ούτο καθεξής. Ακόμα σε κάθε χρήστη αντιστοιχούν και μυστικά κλειδιά ζεύγους χρηστών για τον εαυτό του, έτσι ο Godai-san πρέπει να υπολογίσει και τα μυστικά κλειδιά για το ζεύγος Godai-san/Godai-san, αυτό γίνεται για να μπορέσει να συγχρονίσει όλες τις Groove συσκευές που του ανήκουν.

Υπάρχουν, κατά αντιστοιχία με τη περίπτωση αμοιβαίας εμπιστοσύνης, δύο μυστικά κλειδιά ζεύγους χρηστών. Έτσι στη περίπτωση του ζεύγους Bob/Alice υπάρχει ένα μυστικό κλειδί $K_{Alice/Bob}$ που χρησιμοποιείται για την ασφαλή διανομή (κρυπτογράφηση) των ομαδικών κλειδιών K_G και L_G. Και ένα MAC κλειδί, το $L_{Alice/Bob}$, που εγγυάται την αυθεντικοποίηση της πηγής και την ακεραιότητα.

Σε αυτή τη κατάσταση καχυποψίας, αντί να χρησιμοποιείται ένα MAC επιπέδου ομάδας ({δέλτα μηνύματος}L_G), προστίθενται ένα σύνολο ανεξάρτητων MACs, που ονομάζονται αυθεντικοποιητές (authenticators) ή πολυαυθεντικοποιητές (multiauthenticators). Έτσι η Alice θα προσθέτει τις εξής MACs στο τέλος κάθε δέλτα μηνύματος που στέλνει: {δέλτα μηνύματος}$L_{Alice/Bob}$|{δέλτα μηνύματος}$L_{Alice/Godai-san}$|…|{δέλτα μηνύματος}$L_{Alice/Alice}$. Κάθε MAC παράγεται και αυθεντικοποιείται με τον αλγόριθμο HMAC-SHA1. Ο κάθε χρήστης που θα λάβει το δέλτα της Alice θα επιβεβαιώσει μόνο την MAC που τον αφορά και θα πετάξει τις υπόλοιπες.



Εικόνα 10. Μορφή δέλτα μηνυμάτων

Στην εικόνα βλέπουμε πώς μεταδίδεται ένα δέλτα μήνυμα. Στην περίπτωση αμοιβαίας εμπιστοσύνης το μέγεθος του αυθεντικοποιητή είναι το ίδιο ανεξαρτήτως του πλήθος των συμμετεχόντων χρηστών, χρησιμοποιείται το K_G για τη κρυπτογράφηση του σώματος και το LG για την MAC. Στη περίπτωση αμοιβαίας δυσπιστίας αυτό που αλλάζει είναι ότι χρησιμοποιούνται πολύ-αυθεντικοποιητές, τον οποίων το πλήθος αυξάνει γραμμικά ως προς το πλήθος των συμμετεχόντων χρηστών, κάθε MAC παράγεται ανά ζεύγος χρηστών με το αντίστοιχο LΑποστολέας/Παραλήπτης.

Το Groove προτιμά τη χρήση πολύ-αυθεντικοποιητών αντίς της λύσεις των ψηφιακών υπογραφών. Ο λόγος είναι ότι το μέγεθος και ο χρόνος που απαιτείται για τη παραγωγή ενός αυθεντικοποιητή είναι σταθερά και μικρά, σε αντίθεση με τις πιο ογκώδεις και χρονοβόρες ψηφιακές υπογραφές. Οπότε για μικρές ομάδες, που είναι και το κοινό στο οποίο απευθύνεται αυτή η πρώτη έκδοση του Groove, οι πολύ-αυθεντικοποιητές είναι λειτουργικοί.

Unicast, Mutlicast, Εξυπηρέτες Αναμετάδοσης και το Σύστημα της Βεντάλιας

Το σύστημα Groove υποστηρίζει την αμιγώς ομότιμη επικοινωνία. Στην απλούστερη των περιπτώσεων αρκεί ένα καλώδιο δικτύου, Ethernet RJ-45 συνεστραμμένου ζεύγους 10BaseT να συνδέει δύο υπολογιστές που τρέχουν το

λογισμικό του Groove. Όμως, για να μπορέσει να παράσχει τις υπηρεσίες του ακόμα και σε περιπτώσεις όπου ένας ή περισσότεροι χρήστες μένουν εκτός δικτύου κατά χρονικά διαστήματα ή που βρίσκονται πίσω από αναχώματα ασφαλείας ή σε NATs, το Groove υποστηρίζει και την υβριδική ομότιμη επικοινωνία. Αρκεί ένας πάροχος να προσφέρει υπηρεσίες αναμετάδοσης, με χρήση ενός ή περισσότερων εξυπηρετών που να κατανοούν το πρωτόκολλο Groove. Ένας εξυπηρέτης αναμετάδοσης δε γνωρίζει τίποτα σχετικά με χρήστες-μέλη μπορεί να αντιληφθεί μόνο διαμοιραζόμενους χώρους και Groove συσκευές.

Το Groove χρησιμοποιεί το πρωτόκολλο Device Presence Protocol – DPP για την ονοματοδοσία και την ανακάλυψη συσκευών και το Simple Symmetric Transport Protocol – SSTP για την σύνδεση χρήστη με χρήστη (ομότιμη), χρήστη με αναμεταδότη και αναμεταδότη με αναμεταδότη.

Ο Bob θέλει να στείλει ένα δέλτα μήνυμα (έκανε μια αλλαγή στο διαμοιραζόμενο χώρο και πρέπει να ενημερώσει όλες τις Groove συσκευές), ας θεωρήσουμε ότι λειτουργεί σε κατάσταση αμοιβαίας δυσπιστίας.

- Εάν όλες οι συσκευές είναι προσβάσιμες, δηλαδή συνδεδεμένες και με τη θύρα 2492 (η θύρα αυτή χρησιμοποιείται από το ιδιόκτητο πρωτόκολλο του Groove SSTP) ανοικτή, τότε μπορεί να στείλει το δέλτα με μια ομότιμη σύνδεση. Έχει να επιλέξει ανάμεσα σε:
  - Μια αποστολή multicast, οπότε και όλοι θα πάρουν το ίδιο ακριβώς δέλτα. Αυτό σημαίνει ότι το μήνυμα θα πρέπει να περιλαμβάνει όλες τις MACs για κάθε ζεύγος χρηστών.
  - Μια αποστολή unicast, οπότε και μπορεί να στείλει σε κάθε συσκευή το δέλτα με την αντίστοιχη MAC.
- Εάν κάποιες από τις συσκευές είναι προσωρινά εκτός δικτύου ή δεν δέχονται συνδέσεις στη θύρα 2492. Ο Bob θα πρέπει να χρησιμοποιήσει το σύστημα τη βεντάλιας με έναν εξυπηρέτη αναμετάδοσης. Το σύστημα της βεντάλιας ορίζει ότι ο αποστολέας στέλνει ένα μήνυμα στον εξυπηρέτη και ο εξυπηρέτης αναλαμβάνει να το διαδώσει στις υπόλοιπες συσκευές. Ο εξυπηρέτης μπορεί, ανάλογα με τη πληροφορία που διαθέτει (χρήση ετικετών) να χρησιμοποιήσει τη multicast ή τη unicast αποστολή. Στη δεύτερη επιλογή θα πρέπει να παρασχεθούν ετικέτες που να δείχνουν ποια MAC αντιστοιχεί σε ποιον προορισμό, για να μπορέσει να τροποποιήσει τα αρχικά δεδομένα.

- Σε περίπτωση αδυναμίας και των δύο άνωθεν ενεργειών (πχ, δεν είναι συνδεδεμένος) μπορεί να αποθήκευση το δέλτα και να το στείλει αργότερα.

Ο Bob επιλέγει τον τρόπο αποστολής με βάση τα εξής κριτήρια. Πρώτον η άμεση αποστολή είναι προτιμότερη από την χρήση εξυπηρετών και αυτή με τη σειρά της είναι προτιμότερη από την προσωρινή αποθήκευση. Δεύτερον πραγματοποιείται ένας ευρεστικός υπολογισμός (heuristic calculation) της σχετικής αποδοτικότητας των τριών επιλογών. Λαμβάνονται υπόψη στοιχεία όπως το εύρος σύνδεσης και το μέγεθος για να αποφασιστεί αν η multicast αποστολή ή το σύστημα της βεντάλιας είναι καλύτερο. Επίσης πρέπει να ληφθεί υπόψη αν κάποιος παραλήπτης είναι μη προσβάσιμος.

Ας υποθέσουμε ότι το μήνυμα του Bob δεν φτάνειι ποτέ στην Alice. Το λογισμικό του Groove στη πλευρά της Alice θα καταλάβει ότι λείπει ένα δέλτα, επειδή τα μηνύματα έχουν αριθμούς ακολουθίας, και θα προσπαθήσει να το ανακτήση. Αρχικά θα προσπαθήσει να το πάρει από τον Bob, εάν αυτό αποτύχει θα δοκιμάσει να το πάρει από το Godai-san. Ο Godai-san θα κρυπτογραφήσει το δέλτα με το $K_{Alice/Godai-san}$ κλειδί, αλλά θα χρησιμοποιήσει την MAC με το κλειδί $L_{Alice/Bob}$, που τη κράτησε για μια τέτοια περίσταση.

Δημιουργία Διαμοιραζόμενων Χώρων και Έμπιστη Αυθεντικοποίηση

Ο χρήστης που θα στείλει τη πρόσκληση (invitation) για συμμετοχή σε ένα διαμοιραζόμενο χώρο ορίζεται ως ο προεδρεύων (chair) του χώρου εκείνου. Η πρόσκληση είναι το πρώτο μήνυμα που θα λάβει ένα μέλος και που θα προστεθεί στο διαμοιραζόμενο χώρο. Τα μέλη ενός υπό δημιουργία χώρου, τα οποία τρέχουν ήδη το λογισμικό του Groove, μπορούν να προσκληθούν μέσα από την υπηρεσία άμεσων μηνυμάτων (με ένα μήνυμα του προεδρεύοντος προς τους κληθέντες). Στη περίπτωση που ένα υποψήφιο μέλος δεν έχει το λογισμικό του Groove, θα πρέπει να χρησιμοποιηθεί κάποιο εναλλακτικό (ανασφαλές) κανάλι επικοινωνίας. Το πιο πιθανό είναι η πρόσκληση να σταλεί μέσω email, και μαζί της να σταλούν οδηγίες σχετικά με τη χρήση του Groove και πληροφορίες σχετικές με τη φύση και το σκοπό του εν δημιουργία διαμοιραζόμενου χώρου. Μια πρόσκληση πάντα υπογράφεται ψηφιακά με το ιδιωτικό κλειδί ταυτότητας του προεδρεύοντος, από το λογισμικό του Groove.

Κάθε χρήστης δημιουργεί μια ταυτότητα, η δημόσια προβολή της ονομάζεται επαφή (contact). Η επαφή περιλαμβάνει το όνομα του χρήστη, σε ελεύθερη μορφή μαζί με τα δημόσια κλειδιά των δύο ζευγών δημοσίου-ιδιωτικού κλειδιού, που αναθέτει αυτόματα το Groove στο χρήστη (τα ζεύγη παράγονται στον υπολογιστή του χρήστη, αλλά μελλοντικά θα παρέχεται η δυνατότητα να εισάγει ο χρήστης ζεύγη από τρίτες

οντότητες (Παρόχους Πιστοποιητικών, Έμπιστες Τρίτες Οντότητες, …). Επίσης ένας χρήστης μπορεί να προσθέσει επιπλέον πληροφορία σε μια επαφή, όπως λόγου χάρη ένα τηλέφωνο επικοινωνίας ή μια φυσική διεύθυνση. Προκειμένου να ταυτοποιηθεί και να αυθεντικοποιηθεί ένας χρήστης θα πρέπει να χρησιμοποιηθεί ένα έμπιστο παράπλευρο κανάλι (πχ. συνάντηση πρόσωπο με πρόσωπο και ανταλλαγή δημοσίων κλειδιών).

Πρόσκληση σε Διαμοιραζόμενο Χώρο

Ας υποθέσουμε ότι ο Bob είναι ο προεδρεύων ενός διαμοιραζόμενου χώρου, που ήδη έναν άλλον χρήστη τον, Godai-san, και ότι θέλει να προσκαλέσει την Alice. Θα πρέπει να γίνουν τα εξής βήματα:

- Ο Bob και η Alice πρέπει να ολοκληρώσουν τα βήματα που ορίζει το πρωτόκολλο πρόσκλησης.
- Θα δημιουργηθούν νέα κλειδιά για τα καινούρια ζεύγη χρηστών, δηλαδή για τα ζεύγη Alice/Bob και Alice/Godai-san.
- Το μυστικό ομαδικό κλειδί (K_G) πρέπει να γνωστοποιηθεί στην Alice.

Η πρόσκληση περιέχει όλη τη πληροφορία που χρειάζεται για να εισέλθει ένα νέο μέλος στο διαμοιραζόμενο χώρο. Ένα κρυπτογραφικό περιεχόμενο, στο οποίο ορίζονται μεταξύ άλλων και η Diffie-Hellman παράμετροι που θα χρησιμοποιήσει η Alice για να παράγει το DIffie-Hellman ζεύγος κλειδιών. Το λογισμικό του Bob είναι υπεύθυνο να μεταδώσει το δημόσιο Diffie-Hellman κλειδί της Alice στους υπόλοιπους χρήστες. Και τα δημόσια κλειδιά για τη ταυτοποίηση και αυθεντικοποίηση του Bob.

Από τη πλευρά της η Alice όταν λάβει τη πρόσκληση μπορεί να επιβεβαιώσει ότι προορίζεται για εκείνη, ελέγχοντας το όνομα στη πρόσκληση, να αυθεντικοποιήσει την επαφή του Bob, χρησιμοποιώντας παράπλευρα κανάλια (μπορεί ο Bob να της έχει δώσει σε προηγούμενη συνάντησή τους μια κάρτα που να αναγράφει το δημόσιο κλειδί του), και να πληροφορηθεί σε τι είδους διαμοιραζόμενο χώρο έχει προσκληθεί. Εάν αποφασίσει να συμμετάσχει θα πρέπει να τρέξει το λογισμικό του Groove το οποίο θα στείλει στο Bob ένα μήνυμα μαζί με:

- Ένα μυστικό κλειδί συνόδου κρυπτογραφημένο με το δημόσιο κλειδί του Bob.
- Το δημόσιο Diffie-Hellman κλειδί της, υπογεγραμμένο από το ιδιωτικό κλειδί ταυτότητας της.

Τέλος ο Bob αφού λάβει το μήνυμα της Alice θα αποκρυπτογραφήσει με το ιδιωτικό του κλειδί το κλειδί συνόδου και με το κλειδί συνόδου θα αποκρυπτογραφήσει το μήνυμα της, προκειμένου να επιβεβαιώσει την αποδοχή της πρόσκλησης. Ύστερα μένει

να αυθεντικοποιηθεί η επαφή της Alice (τηλέφωνα, προσωπική συνάντηση, …) και να την αποδεκτή στο διαμοιραζόμενο χώρο, οπότε και:

- Ένα μήνυμα «*εισαγωγή νέου μέλους*» στέλνεται στην ομάδα.
- Η Alice αποκτά ένα αντίγραφο των δεδομένων του διαμοιραζόμενου, συμπεριλαμβανομένων των ομαδικών κλειδιών K_G και L_G. Το αντίγραφο κρυπτογραφείται, φυσικά, με ένα μυστικό κλειδί συνόδου, το οποίο με τη σειρά του έχει κρυπτογραφηθεί με το δημόσιο κλειδί της Alice.

Το Δέλτα Μήνυμα «*εισαγωγή νέου μέλους*» και το Πρωτόκολλο Απομάκρυνσης

Το μήνυμα αυτό χρησιμοποιείται για να πληροφορήσει τα υπάρχοντα μέλη, στην περίπτωσή μας τον Godai-san, σχετικά με την προσθήκη ενός νέου μέλους και για το δημόσιο Diffie-Hellman κλειδί αυτού. Ο Godai-san θα χρησιμοποιήσει αυτή τη πληροφορία για να εγκαθιδρύσει το μυστικό κλειδί του ζεύγους Alice/Godai-san.

Είναι σημαντικό να σημειωθεί ότι ένα μήνυμα «*εισαγωγής νέου μέλους*» δε προκαλεί την αλλαγή των υπαρχόντων ομαδικών κλειδιών. Ο λόγος είναι επειδή θέλουμε τα νέα μέλη να μπορούν να συγχρονίζουν απόλυτα το διαμοιραζόμενο χώρο τους με αυτόν της υπόλοιπης ομάδας και όχι να έχουν γνώση μόνο των δεδομένων που ανταλλάχτηκαν μετά την είσοδό τους.

Ένα κάποιο μέλος απομακρύνει κάποιο άλλο, το λογισμικό Groove του στέλνει ένα δέλτα μήνυμα «*αλλαγής κλειδιού*» (rekey ονομάζεται η αλλαγή της τιμής ενός κρυπτογραφικού κλειδιού [47]), που περιλαμβάνει τη πληροφορία αλλαγής της σύνθεσης της ομάδας και πάνω από το δέλτα (piggy-backing) στέλνει τη πληροφορία αλλαγής κλειδιού κρυπτογραφημένη με τα αντίστοιχα κλειδιά ζεύγους χρηστών. Με το να περιέχεται η πληροφορία αλλαγής κλειδιού μέσα στο δέλτα το Groove εισάγει διαφάνεια ως προς τις εφαρμογές υψηλότερου επιπέδου την σχετικά με τη διαχείριση των μεταδεδομένων ασφάλειας. Και επειδή με αυτό τον τρόπο το νέο κλειδί που στέλνεται πάνω από το δέλτα μπορεί να χρησιμοποιηθεί για να αποκρυπτογραφήσει το ίδιο το δέλτα.

Ταξινόμηση Κλειδιών …

Ένα συνθηματικό ανά λογαριασμό:

Το συνθηματικό είναι σε Unicode και δεν έχει περιορισμό μήκους ή μορφής. Μπορεί να αλλαχτεί κατά βούληση, αλλά είναι ενιαίος για όλες τις συσκευές του χρήστη.

Ένα ζεύγος δημοσίου-ιδιωτικού κλειδιών ταυτότητας για υπογραφή:

Ο προεπιλεγμένος ασύμμετρος αλγόριθμος είναι ο El Gamal. Το ζεύγος χρησιμοποιείται για την αυθεντικοποίηση προσκλήσεων, άμεσα μηνύματα και στο ζεύγος

Diffie-Hellman. Αποθηκεύεται στο λογαριασμό και δεν μπορεί να αλλαχτεί, παρά μόνο δημιουργώντας νέα ταυτότητα. Το δημόσιο κλειδί βρίσκεται και στη επαφή της ταυτότητας, η οποία μοιράζεται στους υπόλοιπους χρήστες.

Ένα συμμετρικό κλειδί ανά λογαριασμό:

Προστατεύει το κλειδί αποθήκευσης του λογαριασμού (βλ. παρακάτω). Δημιουργείται με ασφαλείς διαδικασίες (αλγόριθμος PBKDF2 μαζί με salt των 20-bit και αύξοντα αρίθμηση) από το συνθηματικό και ο προεπιλεγμένος αλγόριθμος είναι ο MARC4 με μέγεθος κλειδιού 256-bit.

Ένα ζεύγος δημοσίου-ιδιωτικού κλειδιών ανά ταυτότητα για την (απο)κρυπτογράφηση συμμετρικών κλειδιών:

Η προεπιλογή είναι ο El Gamal. Αυτό το ζεύγος χρησιμοποιείται για την (από)κρυπτογράφηση των συμμετρικών MARC4 κλειδιών που προστατεύουν το πρωτόκολλο πρόσκλησης και τα άμεσα μηνύματα. Αποθηκεύεται στο λογαριασμό και δεν μπορεί να αλλαχτεί, παρά μόνο δημιουργώντας νέα ταυτότητα. Το δημόσιο κλειδί βρίσκεται και στη επαφή της ταυτότητας, η οποία μοιράζεται στους υπόλοιπους χρήστες.

Ένα ψηφιακό αποτύπωμα ανά ταυτότητα:

Πρόκειται για τη σύνοψη των δύο δημοσίων κλειδιών που προανέφερα. Η σύνοψη αυτή παράγεται και επιβεβαιώνεται από το ίδιο το Groove για κάθε χρήση. Χρησιμοποιείται για την αυθεντικοποίηση των χρηστών.

Ένα ζεύγος δημοσίου-ιδιωτικού κλειδιών Diffie-Helman ανά μέλος, ανά διαμοιραζόμενο χώρο:

Παράγεται αιτιοκρατικά από τα δύο ιδιωτικά κλειδιά που προανέφερα και έναν 24-byte τυχαίο μοναδικό ταυτοποιητή (Global Unique ID – GUID) του διαμοιραζόμενου χώρου. Το κλειδί αυτό είναι μόνιμο για όλη τη διάρκεια της συμμετοχής ενός χρήστη σε έναν διαμοιραζόμενο χώρο και χρησιμοποιείται για τη παραγωγή κλειδιών ζεύγους χρηστών.

Ένα κλειδί ζεύγους χρηστών ΚΑποστολέας/Παραλήπτης ανά ζεύγος χρηστών, ανά διαμοιραζόμενο χώρο για τη διανομή κλειδιών:

Ο προεπιλεγμένος αλγόριθμος είναι ο MARC4. Το κλειδί παράγεται για κάθε ζεύγος χρηστών από το ζεύγος κλειδιών Diffie-Hellman, με τον Diffie-Hellman key agreement αλγόριθμο. Κρατείται στο διαμοιραζόμενο χώρο κάθε χρήστη και χρησιμοποιείται για τη διανομή των ομαδικών κλειδιών όταν αυτά αλλάξουν.

Ένα κλειδί ζεύγους χρηστών LΑποστολέας/Παραλήπτης ανά ζεύγος χρηστών, ανά διαμοιραζόμενο χώρο για την αυθεντικοπόιηση μηνυμάτων:

Ο προεπιλεγμένος αλγόριθμος είναι ο HMAC-SHA1. Το κλειδί παράγεται για κάθε ζεύγος χρηστών από το ζεύγος κλειδιών Diffie-Hellman, με τον Diffie-Hellman key agreement αλγόριθμο. Κρατείται στο διαμοιραζόμενο χώρο κάθε χρήστη και

χρησιμοποιείται για την αυθεντικοποίηση των μηνυμάτων και την επιβεβαίωση της ακεραιότητας αυτών σε κατάσταση αμοιβαίας δυσπιστίας.

Ένα ομαδικό κλειδί Kg ανά διαμοιραζόμενο χώρο για εμπιστευτικότητα:

Από προεπιλογή είναι ένα κλειδί MARC4 που παρέχει εμπιστευτικότητα στο διαμοιραζόμενο χώρο. Αποθηκεύεται στο διαμοιραζόμενο χώρο και αλλάζει όταν απομακρυνθεί ένα μέλος.

Ένα ομαδικό κλειδί Lg ανά διαμοιραζόμενο χώρο για ακεραιότητα μηνυμάτων:

Από προεπιλογή είναι ένα κλειδί HMAC-SHA1 που παρέχει ακεραιότητα στο διαμοιραζόμενο χώρο σε κατάσταση αμοιβαίας εμπιστοσύνης. Αποθηκεύεται στο διαμοιραζόμενο χώρο και αλλάζει όταν απομακρυνθεί ένα μέλος.

Κύριο μυστικό κλειδί:

Ένα MARC4 κλειδί ανά λογαριασμό, αποθηκεύεται στο λογαριασμό και προστατεύει τα κλειδιά αποθήκευσης.

Κλειδιά αποθήκευσης:

Ένα ανά διαμοιραζόμενο χώρο, συμπεριλαμβανομένου του λογαριασμού. Χρησιμοποιούνται για την (απο)κρυπτογράφηση των δεδομένων στον δίσκο. Το κλειδί αποθήκευσης του λογαριασμού προστατεύεται από το συμμετρικό κλειδί που παράγεται από το συνθηματικό. Τα κλειδιά αποθήκευσης των διαμοιραζόμενων χώρων από το κύτιο μυστικό κλειδί.

Εμπιστευτικότητα

Το Groove παρέχει πάντα εμπιστευτικότητα των δεδομένων, τόσο κατά τη μεταφορά τους πάνω από το δίκτυο, όσο και στην αποθήκευσή τους στις συσκευές του χρήστη. Η μόνη περίπτωση που δε παρέχεται εμπιστευτικότητα στην αποθήκευση είναι όταν ο χρήστης επιτρέψει στο λογισμικό να θυμάται το συνθηματικό του και να το χρησιμοποιεί αυτόματα κάθε φορά που συνδέεται, σε μια τέτοια περίπτωση παρέχεται μόνο η φυσική ασφάλεια της ίδιας της συσκευής. Το Groove χρησιμοποιεί το συμμετρικό κρυπτογραφικό αλγόριθμο Modified-Alleged-RC4 – MARC4 με κλειδί 192-bit ως προεπιλογή, αλλά οι χρήστες ενός διαμοιραζόμενου χώρου μπορούν να συμφωνήσουν σε διαφορετικό αλγόριθμο και μέγεθος κλειδιών.

Ακεραιότητα

Χρησιμοποιείται ο αλγόριθμος HMAC-SHA1, ως προεπιλογή, για την προστασία της ακεραιότητας των μηνυμάτων. Εάν έχει επιλεγεί η κατάσταση αμοιβαίας εμπιστοσύνης για όλες τις MAC χρησιμοποιείται ένα κοινό μυστικό κλειδί L_G. Ειδάλλως ο αποστολέας κάθε μηνύματος δημιουργεί τις MAC με το αντίστοιχο μυστικό κλειδί ζεύγους

χρηστών $L_{Αποστολέας/Παραλήπτης}$. Οι χρήστες ενός διαμοιραζόμενου χώρου μπορούν να συμφωνήσουν σε διαφορετικό αλγόριθμο και μέγεθος κλειδιών. Το σύστημα στη παρούσα φάση δε χρησιμοποιεί ψηφιακές υπογραφές για την ακεραιότητα των μηνυμάτων.

Αυθεντικοποίηση

Η αυθεντικοποίηση ενός χρήστη βασίζεται στην επαφή (contact) του. Η επαφή αποτελείται από το όνομα του χρήστη, σε ελεύθερη μορφή μαζί με τα δημόσια κλειδιά των δύο ζευγών δημοσίου-ιδιωτικού κλειδιού που αναθέτει αυτόματα το Groove στο χρήστη (τα ζεύγη παράγονται στον υπολογιστή του χρήστη, αλλά μελλοντικά θα παρέχεται η δυνατότητα να εισάγει ο χρήστης ζεύγη από τρίτες οντότητες (Παρόχου Πιστοποιητικών, Έμπιστες Τρίτες Οντότητες, …). Προκειμένου να ταυτοποιηθεί και να αυθεντικοποιηθεί ένας χρήστης θα πρέπει να χρησιμοποιηθεί ένα έμπιστο παράπλευρο κανάλι (πχ. συνάντηση πρόσωπο με πρόσωπο και ανταλλαγή δημοσίων κλειδιών). Ώστε να αντιστοιχηθεί ένα φυσικό πρόσωπο στην επαφή.

Όσον αφορά τα δεδομένα υπάρχουν δύο καταστάσεις λειτουργίας. Στη κατάσταση αμοιβαίας εμπιστοσύνης τα δεδομένα αυθεντικοποιούνται ότι προήλθαν από κάποιο μέλος του διαμοιραζόμενου χώρου, λόγω του ότι η MAC δημιουργείται με ένα κοινό μυστικό κλειδί L_G. Ενώ σε κατάσταση αμοιβαίας δυσπιστίας χρησιμοποιούνται πολύ-αυθεντικοποιητές, ήτοι για κάθε ζευγάρι χρηστών δημιουργούνται μια MAC με το αντίστοιχο μυστικό κλειδί ζεύγους χρηστών $L_{Αποστολέας/Παραλήπτης}$. Έτσι αυθεντικοποιείται η πηγή του μηνύματος

Διαθεσιμότητα

Το σύστημα εγγυάται ότι τα μηνύματα θα φτάσουν σε όλα τα μέλη ακόμα και αν κάποια από αυτά, συμπεριλαμβανομένου και του αποστολέα, βρίσκονται εκείνη τη στιγμή εκτός δικτύου ή πίσω από αναχώματα ασφαλείας ή NATs (χρήση εξυπηρετών αναμετάδοσης). Χρησιμοποιούνται αριθμοί ακολουθίας για την ανακάλυψη της απώλειας ενός μηνύματος και το σύστημα μεριμνάει ώστε να φυλάσσονται οι MAC, για την επιβεβαίωση της εγκυρότητας των καθυστερημένων μηνυμάτων.

Ανωνυμία

Σε κατάσταση αμοιβαίας εμπιστοσύνης παρέχεται ένα είδος ανωνυμίας στο θέμα της διάδοσης των μηνυμάτων, επειδή χρησιμοποιείται ένα κοινό κλειδί L_G για τη δημιουργία της MAC του μηνύματος, όμως εάν χρησιμοποιηθεί άμεση σύνδεση για τη μεταφορά του μηνύματος αποκαλύπτεται η IP διεύθυνση του αποστολέα, …. Επίσης είναι δυνατό να δημιουργηθεί ένα διαμοιραζόμενος χώρος όπου όλοι οι χρήστες

χρησιμοποιούν ψευδώνυμα στη επαφή τους, βέβαια μένει το πρόβλημα της πρόσκλησης (πώς προσκαλείς κάποιον που δε ξέρεις;). Αλλά ούτως ή άλλως η ανωνυμία δεν είναι συνήθης απαίτηση στα συνεργατικά συστήματα.

Μη αποποίηση Αποστολής και Απονομή Ευθυνών

Παρέχεται μόνο στη κατάσταση αμοιβαίας δυσπιστίας.

2.9 OceanStore

Υβριδικό Ομότιμο δίκτυο Κατανεμημένης Αποθήκευσης

Σύντομη Περιγραφή

Το OceanStore [75], [76], [77] είναι μια παγκόσμια επίμονη (persistent) αποθήκη δεδομένων σχεδιασμένη για να μπορέσει να εξυπηρετήσει δισεκατομμύρια χρηστών (κατ' εκτίμηση των σχεδιαστών του θα εξυπηρετούνται 10^{10} χρήστες, ο καθένας με 10^3 αρχεία). Είναι σχεδιασμένη ώστε να παρέχει συνεχή πρόσβαση στο σύστημα και υψηλή διαθεσιμότητα για τα αποθηκευμένα δεδομένα. Επειδή η υποδομή του βασίζεται σε ανασφαλείς και μη έμπιστους εξυπηρέτες, κάνει χρήση πλεονασμού και κρυπτογραφικών τεχνικών για να επιτύχει ρωμαλεότητα και προστασία του περιεχομένου. Η κινητήρια δύναμη πίσω από την δημιουργία του είναι η παροχή ασφαλούς, ρωμαλέας και επίμονης πληροφορίας αποσπασμένης από την έννοια του τόπου (νομαδικά δεδομένα – nomadic data), και η δυνατότητα κατασκευής του συστήματος από μια μη δομημένη υποδομή, αυτές οι ιδιότητες θα δώσουν την απαιτούμενη αδιαφάνεια που χρειάζεται η πανταχού παρούσα υπολογιστική (ubiquitous computing) για να αναπτυχθεί.

Ιστορία

Ήδη κυκλοφορεί ένα πρωτότυπο σύστημα, που υλοποιεί το Ocean Store, υπό την ονομασία Pond. Το Pond είναι λογισμικό ανοικτού κώδικα και χρησιμοποιεί τη τελευταία υλοποίηση του δομημένου ομότιμου δίκτυου Tapestry για την αποθήκευση και αναζήτηση αρχείων.

Περιγραφή Συστήματος

Κάθε υπολογιστής μπορεί να συμμετάσχει στο σύστημα, προσφέροντας αποθηκευτικό χώρο ή παρέχοντας τοπική πρόσβαση προς τους χρήστες έναντι οικονομικής ανταμοιβής. Οι χρήστες αρκεί να εγγραφούν σε ένα πάροχο υπηρεσιών OceanStore (OceanStore service provider), αν και μπορούν να καταναλώνουν αποθηκευτικό χώρο και εύρος ζώνης από πολλούς παρόχους. Οι πάροχοι αυτόματα αγοράσουν και πωλούν αποθηκευτικό χώρο μεταξύ τους, αδιαφανώς ως προς τους χρήστες.

Το OceanStore κρατάει αντίγραφα (caches) των δεδομένων αδιακρίτως, με τυχαίες επιλογές (promiscuously); οποιοσδήποτε εξυπηρέτης μπορεί να δημιουργήσει

ένα τοπικό αντίγραφο (replica) οποιουδήποτε αντικειμένου δεδομένων (data object). Αυτά τα τοπικά αντίγραφα προσφέρουν ταχύτερη πρόσβαση και ρωμαλεότητα και επίσης μειώνουν τη συμφόρηση του δικτύου (τοπική κίνηση).

Υποθέτετε ότι κάθε εξυπηρέτης στην υποδομή μπορεί να αποτύχει (crash), διαρρεύσει πληροφορία ή να υποστεί ρήγμα ασφάλειας. Η αδιάκριτη και τυχαία αποθήκευση (promiscuous caching) επιβάλει τη χρήση πλεονασμού και κρυπτογραφικών τεχνικών για να προστατευτούν τα δεδομένα από τους εξυπηρέτες στους οποίους βρίσκονται.

Το OceanStore εφαρμόζει ένα ανθεκτικό στα βυζαντινά σφάλματα πρωτόκολλο για να επιτύχει συνέπεια στα αντίγραφα.

Εάν αποθηκευτικό σύστημα αρχειοθέτησης βασισμένος στις εκδόσεις (ενν. των περιεχομένων) (version-based archival storage system) προσφέρει υψηλή ανθεκτικότητα. Το OceanStore αποθηκεύει κάθε έκδοση ενός αντικειμένου δεδομένων σε μια μόνιμη, για ανάγνωση-μόνο μορφή, η οποία κωδικοποιείται με έναν κωδικό διαγραφής και διαδίδεται σε χιλιάδες εξυπηρέτες. Ένα μικρό μόνο σύνολο των κωδικοποιημένων κομματιών είναι αρκετό για να ανακτηθεί το αρχειοθετημένο αντικείμενο.

Ακόμα το σύστημα χρησιμοποιεί ένα επίπεδο ενδοσκόπησης (introspection layer) που προσαρμόζει και βελτιώνει την απόδοση και την ανοχή σε σφάλματα του δικτύου. Επειδή η διαχείριση ενός τόσο εκτεταμένου συστήματος είναι αδύνατη ή τουλάχιστον οικονομικώς μη επιθυμητή, το σύστημα αυτόματα παρακολουθεί, συλλέγει και αναλύει, εσωτερικά, πληροφορία, όπως μοτίβα χρήσης (και εδώ υπάρχουν κάποιες υποχωρήσει όσον αφορά την ιδιωτικότητα των χρηστών), δικτυακή δραστηριότητα και διαθεσιμότητα πόρων και προσαρμόζεται ώστε να αντιμετωπίσει την τοπική παλαίωση και επιθέσεις άρνησης υπηρεσίας ή και προφυλακτικά μπορεί να μετοικίσει δεδομένα κοντύτερα σε περιοχές χρήσης και να διατηρήσει υψηλά επίπεδα πλεονασμού δεδομένων.

Ανακεφαλαιώνοντας για να μπορέσει το OceanStore, να λειτουργήσει με το βέλτιστο δυνατό τρόπο χρειάζεται ένα σύστημα τοποθέτησης αντικειμένων και δρομολόγησης που δε ταξιδεύει έξω από την τοπική περιοχή εκτός και αν αυτό είναι απαραίτητο, χρειάζεται δηλαδή τοπικότητα (locality). Η τοπικότητα προσφέρει μικρότερες καθυστερήσεις (latency), καλύτερη αξιοπιστία και καλύτερη αξιοποίηση του διαθέσιμου εύρους ζώνης. Τα περισσότερα συστήματα τοποθέτησης αντικειμένων, όπως το CAN και το Chord δεν έχουν αυτές τις ιδιότητες. Για αυτό στην τρέχουσα υλοποίηση του OceanStore χρησιμοποιείται το Tapestry.

ΚΕΦΑΛΑΙΟ 3 - ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ.

3.1 Διαμοιρασμός Αρχείων

Η εκτεταμένη χρήση των ομότιμων δικτύων διαμοιρασμού αρχείων (file-sharing P2P networks) έχει τρέψει τους τελικούς χρήστες (end users) από καταναλωτές περιεχομένου (content) σε εν δυνάμει εκδότες (publishers) και παρόχους περιεχομένου. Οι P2P τεχνολογίες έδωσαν σε άγνωστους μεταξύ τους χρήστες την δυνατότητα να προμηθεύουν και να προμηθεύονται αρχεία ή και συλλογές αυτών.

Σε αυτό το σενάριο η χρήστης Alice επιθυμεί να μοιραστεί περιεχόμενο που βρίσκεται αποθηκευμένο στον υπολογιστή της με τον υπόλοιπο κόσμο (τουλάχιστον με το κομμάτι αυτό, του κόσμου, που χρησιμοποιεί το ίδιο ομότιμο δίκτυο με εκείνη) και ως συνέπεια αυτού (στηριζόμενη προφανώς σε κάποια αρχή ανταπόδοσης και fair play) θέλει να κατεβάσει περιεχόμενο που βρίσκεται αποθηκευμένο σε άλλους ομότιμους κόμβους. Υποθέτω τις ακόλουθες απαιτήσεις ασφάλειας:

Ακεραιότητα

Η πρωταρχική ασφαλώς απαίτηση είναι να παρέχεται ακεραιότητα των διαμοιραζόμενων δεδομένων, η οποία ικανοποιείται με τη δημιουργία και διανομή συνόψεων του αρχείου. Υπάρχει πληθώρα αλγορίθμων σύνοψης που παράγουν ικανοποιητικές συνόψεις. Επίσης με την δημιουργία συνόψεων υποσυνόλων του αρχικού αρχείου γίνεται εφικτό το ταυτόχρονο κατέβασμα ενός αρχείου από πολλούς χρήστες.

Αυθεντικοποίηση Δεδομένων

Εάν η χρήστης Alice έχει αμφιβολίες ως προς τα περιεχόμενα του αρχείου που κατεβάζει (επιθέσεις δηλητηρίασης/νόθευσης των δεδομένων, …), μπορεί να προτιμήσει τη χρήση παράπλευρων καναλιών για να αναζητήσει για περιεχόμενο. Τα πρόβλημα αυθεντικοποίησης τρέπεται πρόβλημα εμπιστοσύνης της πηγής που πρόσφερε το σύνδεσμο προς το αρχείο. Επιπλέον πολλά δίκτυα ενσωματώνουν λειτουργίες αξιολόγησης και σχολιασμού των διαμοιραζόμενων αρχείων.

Αδύναμη Ανωνυμία Vs Δυνατή Ανωνυμία

Αφού η Υποδομή Δημοσίου Κλειδιού (PKI) δε χρησιμοποιείται παντού και δεν υπάρχει ένας κοινά αποδεκτός Πάροχος Πιστοποίησης, δε φαίνεται να υπάρχει κάποιος

τρόπος αυθεντικοποίησης και ταυτοποίησης των χρηστών, ειδικά όταν σκεφτεί κανείς ότι τα ομότιμα δίκτυα δεν περιορίζονται από σύνορα και ένα δίκτυο εμπιστοσύνης ανάμεσα στους ομότιμους χρήστες θα ήταν στην καλύτερη περίπτωση αδύναμο. Υπάρχουν ομότιμα δίκτυα που χρησιμοποιούν διάφορους τρόπους ταυτοποίησης κόμβων με χρήση μοναδικών ταυτοποιητών που βασίζονται στο υλικό του κόμβου ή/και την IP διεύθυνση. Αλλά από τη στιγμή που ο χρήστης έχει πρόσβαση στο περιεχόμενο της RAM του, μπορεί και τρέχει εικονικές μηχανές και δεδομένου της δυναμικής ανάθεσης IP διευθύνσεων, αυτή η μορφή ταυτοποίησης θεωρείται τετριμμένη. Άλλωστε από τη στιγμή που τα περισσότερα δίκτυα χρησιμοποιούν απευθείας συνδέσεις μεταξύ των ομότιμων χρηστών για το κατέβασμα περιεχομένου δε είναι δυνατό να αποφευχθεί η αποκάλυψη της IP διευθύνσεως του χρήστη. Υπάρχουν ασφαλώς, συστήματα που υπόσχονται πλήρη ανωνυμία, αλλά είτε το κόστος χρήσης τους είναι πολύ υψηλό (κατανάλωση πόρων, δυσχρηστία) είτε δεν υπάρχει κάποια ρωμαλέα υλοποίησή τους έτοιμη προς χρήση (περίπτωση Free Haven).

3.2 Υπολογιστική Κοινών Πόρων

3.2.1 Απλό Σενάριο Υπολογιστικής Κοινών Πόρων

Ο Bob δουλεύει σε ένα ακαδημαϊκό πρόγραμμα, που έχει σαν στόχο την ανακάλυψη της Πανάκειας. Αλλά το πανεπιστήμιο, αναλογιζόμενο την απιθανότητα του εγχειρήματος, δε φαίνεται πρόθυμο να χρηματοδοτήσει την αγορά ενός ASCI WHITE υπερυπολογιστή της IBM για να πραγματοποιήσει τους απαιτούμενους υπολογισμούς. Μια λύση θα ήταν η κατανεμημένη επεξεργασία ανάμεσα σε πολλές μηχανές με χρήση ενός πλαισίου (framework), όπως το Parallel Virtual Machine (PVM) (βλ. Parallel Virtual Machine Project. http://www.epm.ornl.gov/pvm/pvm_home.html) ή το Beowulf (βλ. The Beowulf Project. http://cesdis.gsfc.nasa.gov/linux-web/beowulf.html). Το πρόβλημα με αυτές τις λύσεις είναι ότι ο Bob μπορεί να χρησιμοποιήσει μόνο υπολογιστές που του ανήκουν ή στους οποίους έχει πρόσβαση με υψηλά δικαιώματα, και σίγουρα αυτοί είναι περιορισμένοι στο πλήθος. Η έννοια των ομότιμων δικτύων παρέχει στον Bob μια νέα λύση στο πρόβλημά του. Χρησιμοποιώντας κάποιο υπάρχων πλαίσιο, όπως το BOINC, στο οποίο βασίζονται γνωστά προγράμματα κατανεμημένης επεξεργασίας σαν το Seti@home project, ο Bob μπορεί να χτίσει τη δική του ομότιμη εφαρμογή κατανεμημένης επεξεργασίας, έστω ότι αυτή ονομάζεται Panacea@home. Το μόνο που απομένει στον Bob είναι να πείσει αρκετούς χρήστες να κατεβάσουν και να τρέξουν το Panacea@home, συνεισφέροντας επεξεργαστική ισχύ, εύρος σύνδεσης, και αποθηκευτικό χώρο για το σκοπό του. Από ‘κει κα πέρα μένει μόνο η διαχείριση του Κεντρικού Συστήματος, το οποίο είναι υπεύθυνο για το ανέβασμα των μονάδων εργασίας και τη συλλογή των αποτελεσμάτων.

Θα θέσω τώρα τις απαιτήσεις ασφάλειας που πρέπει να ληφθούν υπόψη κατά την ανάπτυξη του συστήματος. Πρώτα θα σας παρουσιάσω την αντίπαλο του Bob την Pharma. Η κακόβουλος αυτή χρήστης έχει ισχυρά κίνητρα να προκαλέσει χάος στο σύστημα, για τους δικούς της σκοτεινούς λόγους. Ένας τρόπος για να το καταφέρει αυτό είναι με το να ξεγελάσει ένα ικανοποιητικό πλήθος χρηστών, ούτως ώστε αυτοί να υποβάλλουν εσφαλμένα αποτελέσματα. Για να το πετύχει αυτό αρκεί να κατασκευάσει και διαδώσει μια τροποποιημένη έκδοση του Panacea@home που θα πραγματοποιούσε εσφαλμένους υπολογισμούς (πχ. πράξεις με μικρότερη ακρίβεια δεκαδικών ψηφίων) και ταυτόχρονα θα επέτρεπε στους χρήστες της να ανεβούν στην κατάταξη γρηγορότερα (πχ. υποβολή μεγάλου πλήθους εσφαλμένων αποτελεσμάτων, που δεν έχουν υπολογιστεί).

Γίνεται εμφανές ότι ο Bob χρειάζεται ένα τρόπο να πιστοποιεί την εγκυρότητα των λαμβανόμενων αποτελεσμάτων. Συγκεκριμένα χρειάζεται ακεραιότητα των

υποβαλλόμενων δεδομένων (κανείς δε πρέπει να μπορεί να αλλάζει τα καθ' οδόν δεδομένα) και αυθεντικοποίησή τους (οι μονάδες εργασίας πρέπει να επεξεργάστηκαν μέσω του αυθεντικού Panacea@home).

Ο Bob χρειάζεται διαθεσιμότητα δεδομένων, από τη στιγμή που οι κόμβοι δεν μπορούν να εγγυηθούν για τη διαρκή σύνδεσή τους ή για την επεξεργασία κάθε μονάδας εργασίας που έχουν λάβει, θα πρέπει να υπάρχει πλεονασμός. Ο πλεονασμός, ήτοι περισσότεροι του ενός κόμβοι επεξεργάζονται την ίδια μονάδα εργασίας, επιτρέπει και τη διάκριση ανάμεσα σε εσφαλμένους (είτε λόγω βυζαντινών σφαλμάτων, είτε λόγω κακόβουλης παρεμβολής) και ορθούς κόμβους.

Σχετικά με την αυθεντικοποίηση, εξουσιοδότηση και ανωνυμία χρήστη, ο Bob θέλει να μπορεί να διακρίνει τους χρήστες του, ώστε να μπορεί να τιμωρεί αυτούς που τρέχουν εσφαλμένους κόμβους, ταυτόχρονα, όμως, δεν ενδιαφέρεται για τη πραγματική ταυτότητα του καθενός. Του αρκεί να επιτρέπει μόνο σε εξουσιοδοτημένους χρήστες να κατεβάζουν μονάδες εργασίας και να υποβάλλουν πίσω αποτελέσματα. Δια τούτο κάθε χρήστης θα πρέπει να εγγράφεται χρησιμοποιώντας ένα ψευδώνυμο και πιθανών το email του, δεν είναι απαραίτητη η χρήση x.509 και PKI, να δημιουργεί έναν λογαριασμό χρήστη που προστατεύεται με κάποιο συνθηματικό, και μετά να χρησιμοποιεί το ζεύγος ψευδώνυμο/συνθηματικό για να αποκτά εξουσιοδότηση να συνδεθεί στο σύστημα και να ξεκινήσει την επεξεργασία. Ο κυριότερος λόγος που δε ζητείται η χρήση πιστοποιητικών και υποδομής δημοσίου κλειδιού είναι ότι κάτι τέτοιο θα αύξανε τη πολυπλοκότητα του συστήματος και το κόστος χρήσης του και ως εκ τούτου θα μείωνε το πλήθος των χρηστών. Το σύστημα αυτό προσφέρει μια αδύναμη ανωνυμία, καθώς ενώ IP διεύθυνση του χρήστη γίνεται γνωστή κάθε φορά που συνδέεται στο Κεντρικό Σύστημα, ο Bob μπορεί να διαχειρίζεται τα δεδομένα, που απέκτησε από τους χρήστες κατά την εγγραφή στο σύστημα, κατά τέτοιο τρόπο ώστε να μην είναι διαθέσιμα σε τρίτους. Αυτού του βαθμού η προστασία της ιδιωτικότητας είναι αρκετή, καθώς το κόστος προσφοράς δυνατής ανωνυμίας (δρομολόγηση μέσα από δίκτυα τύπου mix ή crowd, etc.) είναι δυσανάλογο.

Τέλος όσον αφορά την εμπιστευτικότητα, αυτή δε κρίνεται ούτε απαραίτητη, αλλά ούτε και εφικτή. Δε υπάρχει κάποιος οικονομικός και πρακτικός τρόπος να επιτευχθεί η κρυπτογράφηση των δεδομένων. Αυτό κυρίως οφείλεται στο ότι ο Bob δε γνωρίζει στην πραγματικότητα του χρήστες που συμμετέχουν και άρα δεν έχει νόημα η διανομή κλειδιών για τη προστασία των δεδομένων κατά τη μεταφορά, αφού ανάμεσα στους χρήστες μπορεί να βρίσκεται και η Pharma, μάλιστα η Phrama μπορεί να συμμετέχει πολλές φορές κάτω από διαφορετικά ψευδώνυμα/email (βλ. επίθεση Sybil). Επίσης η λύση της ενσωμάτωσης συμμετρικών κλειδιών μέσα στο λογισμικό δεν είναι ασφαλείς αφού μπορεί να γίνει reverse engineering και αυτά να βρεθούν.

3.2.2 Σενάριο Υπολογιστικής Κοινών Πόρων με Μυστικότητα

Η Eve εργάζεται στο τμήμα R&D μιας πολυεθνικής εταιρίας. Προκειμένου να πάρει την πολυπόθητη αύξηση θα πρέπει να αναπτύξει ένα νέο προϊόν που θα κάνει θραύση, όμως αυτό απαιτεί το τρέξιμο προσομοιώσεων που έχουν μεγάλο υπολογιστικό φόρτο. Επειδή ο υπερυπολογιστής της εταιρίας βρίσκεται ήδη κάτω από εκτεταμένη χρηση, θα πρέπει να βρει μια εναλλακτική λύση. Και πάλι θα μπορούσε να χρησιμοποιήσει το Beowulf ή το PVM, αλλά η ομότιμη υπολογιστική φαίνεται πιο ελκυστική. Αυτό που σκέφτεται η Eve είναι αντί να εκμεταλλευτεί τους κόμβους που βρίσκονται στις παρυφές του δικτύου, άλλωστε ποιος θα χάριζε τους υπολογιστικούς του πόρους σε μια πολυεθνική και πώς θα κατάφερνε η Eve να προστατεύσει την μυστικότητα του προγράμματος, αντίθετα να χρησιμοποιήσει τις υπολογιστικές συσκευές της εταιρίας ανά τον κόσμο, οι οποίες ούτως ή άλλως παραμένουν ανενεργές για μεγάλα χρονικά διαστήματα. Αντίθετα με το προηγούμενο σενάριο, η εφαρμογή της Eve τρέχει σε ένα σαφώς πιο καθορισμένο περιβάλλον όπου υπάρχει μεγαλύτερη εμπιστοσύνη, το οποίο είναι πιο σταθερό και που μπορεί να ανταποκριθεί σε ορισμένες απαιτήσεις που θα θέσει (πχ. η ύπαρξη ψηφιακών πιστοποιητικών, ζεύγη δημοσίων-ιδιωτικών κλειδιών). Επιπλέον η Eve δε χρειάζεται να ανησυχεί για τη διάδοση και τις μετέπειτα αναβαθμίσεις του λογισμικού πελάτη (όλοι οι διαχειριστές δουλεύουν για την ίδια εταιρία). Και τέλος δεν υπάρχουν ταπεινά κίνητρα για τους χρήστες, όπως λίστες κατάταξης, που θα τους έβαζαν σε πειρασμό να υποβάλλουν εσφαλμένα αποτελέσματα ή να παραβιάσουν το σύστημα, εάν κάποιος θελήσει να καταστρέψει το σύστημα αυτό σημαίνει ότι θα έχει ένα ισχυρό κίνητρο και άρα πρόκειται για μεμονωμένα άτομα. Ακόμα αν θεωρηθεί ότι βιομηχανική κατασκοπία αντιμετωπίζεται αποτελεσματικά στο σύνολό της, η Eve έχει μόνο να ανησυχεί για τη διαρροή πληροφορίας κατά τη μεταφορά των δεδομένων, οπότε και μπορεί να εφαρμόσει συμμετρική κρυπτογράφηση στα ανταλλασσόμενα δεδομένα και διανομή κλειδιών με χρήση ζευγών δημοσίου-ιδιωτικού κλειδιού (ψηφιακός φάκελος) για να παράσχει εμπιστευτικότητα. Κατά τα άλλα ισχύει ότι και στο προηγούμενο σενάριο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Rüdiger Schollmeier, Peer-to-Peer Networking. Applications for and Impacts on Future IP-Based Networks

[2] distributed.net web site, www.distributed.net

[3] RSA Labs web site, http://www.rsasecurity.com/rsalabs/

[4] Napster web site, http://www.napster.com

[5] wikipedia, http://en.wikipedia.org

[6] SETI@home web site, http://setiathome.berkeley.edu/

[7] Oram, Andy (Ed.), Peer to Peer – Harnessing the Power of Disruptive Technologies, 1st ed., O'Reilly Press, March 2001

[8] ntrg.cs.tcd.ie

[9] Gnutella Development Forum, http://rfc-gnutella.sf.net/

[10] Limewire web site,http://www.limewire.com

[11] eDonkey2000 web site, http://www.edonkey2000.com

[12] Petar Maymounkov, David Mazières, Kademlia: A Peer-to-peer Information System Based on the XOR Metric

[13] eMule Project web site, http://www.emule-project.net/

[14] giFT, a FastTrack plugin http://developer.berlios.de/projects/gift-fasttrack/

[15] Oliver Heckmann, Axel Bock, The eDonkey 2000 Protocol, KOM Technical Report, August 2002

[16] Bram Cohen, Incentives Build Robustness in BitTorrent, May 2003

[17] Bittorrent web site, http://www.bittorrent.com

[18] Roger R. Dingledinem, M. J. Freedman, D. Molnar, The Free Haven Project: Distributed anonymous storage service, In Workshop on Design Issues in Anonymity and Unobservability, July 2000

[19] Roger R. Dingledinem, The Free Haven Project: Design and Deployment of an Anonymous Secure Data Haven, May 2000

[20] Roger Dingledine, Nick Mathewson and Paul Syverson, Reputation in P2P Anonymity Systems

[21] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In Workshop on Design Issues in Anonymity and Unobservability, pages 46–66, 2000

[22] Skype Guide for Network Administrators, http://www.skype.com/

[23] Salman A. Baset, Henning Schulzrinne, An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol, 2004

[24] Groove web site, http://www.groove.net/

[25] D. Plonka. University of Wisconsin-Madison, Napster traffic measurement, March 2000, http://net.doit.wisc.edu/data/Napster

[26] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of internet content delivery systems. In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA, December 2002

[27] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan, Measurement, Modelling, and Analysis of a Peer-to-Peer FileSharing Workload, SOPS'03 ACM, 2003

[28] Meta Computing, Peer-to-Peer Architecture Proposal Legion - An Integrated Architecture for Secure Resource Sharing, white paper, January 2001

[29] S. Wray; T. Glauert; A. Hopper, "The Medusa applications environment", Multimedia Computing and Systems, 1994, in Proceedings of the International Conference, pp. 265-273, 1994

[30] L. Thomas, S. Suchter, A. Rifkin, Developing Peer-to-Peer Applications on the Internet: the Distributed Editor, Simulated, Dr. Dobb's Journal #281, pp. 76-81, January 1998

[31] M. P. Singh, Peering at Peer-to-Peer Computing, IEEE Internet Computing, Vol. 5, No. 1, January/February 2001

[32] O'Reilly openp2p web site, http://openp2p.com/

[33] Rüdiger Schollmeier, A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In Proceedings of the First International Conference on Peer-to-Peer Computing (P2P'01), 2001

[34] John Risson and Tim Moors, Survey of Research towards Robust Peer-to-Peer Networks: Search Methods, 2004

[35] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp,and Scott Shenker. A scalable content-addressable network. In Proceedings ACM SIGCOMM'01, San Diego, California, August 2001

[36] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In Proceedings of SIGCOMM Annual Conference on Data Communication, San Diego, California, August 2001

[37] Antony Rowstron and Peter Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, in Proceedings of IFIP/ACM Middleware 2001, Heidelberg, Germany, November 2001

[38] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph, Tapestry: An infrastructure for fault-resilient wide-area location and routing. Technical Report UCB//CSD-01-1141, U. C. Berkeley, April 2001

[39] Sean Rhea, Brighten Godfrey, Brad Karp, John Kubiatowicz, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Harlan Yu, OpenDHT: A Public DHT Service and Its Uses, SIGCOMM'05, August 2005,

[40] Emil Sit and Robert Morris, Security Considerations for Peer-to-Peer Distributed Hash Tables, in Proceedings of IPTPS, 2002

[41] Mudhakar Srivatsa and Ling Liu, Vulnerabilities and Security Threats in Structured Peer-to-Peer Systems: A Quantitative Analysis

[42] BitComet web site, http://wiki.bitcomet.com

[43] The Bamboo Distributed Hash Table, http://bamboo-dht.org

[44] Image from An Atlas of CyberSpace, Topology maps (page 2), http://www.geog.ucl.ac.uk/casa/martin/atlas/more_topology.html

[45] Στέφανος Γκρίτζαλης, Σωκράτης Κ. Κατσικας, Δημήτρης Γκρίτζαλης, Ασφάλεια δικτύων υπολογιστών : τεχνολογίες και υπηρεσίες σε περιβάλλοντα ηλεκτρονικού επιχειρείν και ηλεκτρονικής διακυβέρνησης, Παπασωτηρίου, Αθήνα 2003

[46] Matt Bishop, Introduction to Computer Security, Addison-Wesley, 2004

[47] Internet Security Glossary. R. Shirey. May 2000, Status: INFORMATIONAL

[48] Bruce Schneier, Applied Cryptography, 2nd Edition, John Wiley & Sons, 1996

[49] Bruce Schneier, Security Pitfalls In Cryptography, Counterpane Systems, 1998

[50] Miguel Castro, Practical Byzantine Fault Tolerance, 2001

[51] Leslie Lamport, Robert Shostak, Marshall Pease, The Byzantine Generals Problem, in ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, Pages 382-401

[52] Robert W. Shirey. Security Architecture for Internet Protocols. A Guide for Protocol Designs and Standards. Internet Draft: draft-irtf-psrg-secarch-sect100.txt, November 1994

[53] Ernesto Damiani, De Capitani di Vimercati, Stefano Paraboschi, Pierangela Samati and Fabio VIolante, A Reputation-Based Approach for Choosing Reliable Resources in PeertoPeer Networks, CCS'02 ACM, 2002

[54] Lidong Zhou, Lintao Zhang, Frank McSherry, Nicole Immorlica, Manuel Costa and Steve Chien , A First Look at Peer-to-Peer Worms: Threats and Defenses, in Proceedings of 4th International Workshop on Peer-to-Peer Systems (IPTPS'05), 2005

[55] Dan S. Wallach, A Survey of Peer-to-Peer Security Issues, in Proceedings of International Symposium on Software Security 2002

[56] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron and Dan S. Wallach, Secure routing for structured peer-to-peer overlay networks, in Proceedings of the 5th Usenix Symposium on Operating Systems Design and Implementation, Boston, MA, December 2002

[57] Neil Daswani and Hector GarciaMolina, QueryFlood DoS Attacks in Gnutella

[58] Eytan Adar and Bernando A. Huberman, Free Riding on Gnutella

[59] J. R. Douceur. The Sybil Attack. In Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), Cambridge, Massachusetts, Mar. 2002

[60] Atul Singh, Miguel Castro, Peter Druschel and Antony Rowstron, Defending against Eclipse attacks on overlay networks

[61] S. Bellovin, Security aspects of Napster and Gnutella, In Proc. Of USENIX 2001, Boston, June 2001

[62] GnuMap Project, http://home.attbi.com/~gregory.bray

[63] Gnutella2 web site, http://www.gnutella2.com

[64] Michael O. Rabin, Efficient dispersal of information for security, load balancing, and fault tolerance, April 1989

[65] David Hopwood, Definition of recipient-hiding cryptosystem, sci.crypt, usenet post

[66] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudo-nyms. *Communications of the ACM*, 4(2), February 1981

[67] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type III anonymous remailer protocol, in proceedings 2003 IEEE Symposium on Security and Privacy, pages 2–15. IEEE CS, May 2003

[68] Roger Dingledine, Nick Mathewson, Paul Syverson, Tor: The Second-Generation Onion Router

[69] Berkeley Open Infrastructure for Network Computing web site, www.boinc.edu

[70] David P. Anderson, BOINC: A System for Public-Resource Computing and Storage

[71] http://boinc-doc.net/boinc-wiki

[72] David P. Anderson, Public Computing: Reconnecting People to Science

[73] FIPS 180-1

[74] RFC 22104

[75] OceanStore web site, http://oceanstore.cs.berkeley.edu

[76] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Zhao, OceanStore: An Architecture for Global-Scale Persistent Storage, Technical Report UCB/CSD-00-1102, May 1999

[77] David Bindel, Yan Chen, Patrick Eaton, Dennis Geels,Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon,Westley Weimer, Christopher Wells, Ben Zhao, and John Kubiatowicz, OceanStore: An Extremely Wide{Area Storage System