ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ ΞΑΝΘΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αλγοριθμοι Αξιοποιησης Προσωπικων Δεδομενων Θεσης με Ταυτοχρονη Προστασια της Ιδιωτικοτητας: Εφαρμογη στην Παροχη Ιατρικων Υπηρεσιων

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ
του ΔΡΟΣΑΤΟΥ ΓΕΩΡΓΙΟΥ

Επιβλέπων: Αν. Καθ. Αλέξανδρος Καράκος

ΞΑΝΘΗ, ΜΑΡΤΙΟΣ 2010

Αφιερώνεται στον καθηγητή και
επιβλέποντα αυτής της μεταπτυχιακής εργασίας,
στον Επικ. Καθ. Παύλο Εφραιμίδη,
στην οικογένειά μου,
και σε όσους με βοήθησαν σε
αυτή μου την προσπάθεια.

ΠΕΡΙΛΗΨΗ

Στην παρούσα μεταπτυχιακή διατριβή, ορίζεται το πρόβλημα του κοντινότερου γιατρού (NDP = Nearest Doctor Problem) για την εύρεση του κοντινότερου γιατρού σε περιπτώσεις εκτάκτου ανάγκης και παρουσιάζεται ένας ασφαλής υπολογισμός (secure multi-party computation) για την επίλυση του. Η λύση του NDP βασίζεται σε ένα κρυπτογραφικό πρωτόκολλο διασφάλισης της ιδιωτικότητας και χρησιμοποιεί τη τρέχουσα θέση του κάθε γιατρού για το σκοπό αυτό. Το πρωτόκολλο αυτό είναι αποδοτικό και προστατεύει την ιδιωτικότητα θέσης όλων των γιατρών. Τέλος, γίνεται υλοποίηση ενός πρωτοτύπου της προτεινόμενης λύσης για μια κοινότητα γιατρών που χρησιμοποιεί φορητές συσκευές για την εύρεσης της τρέχουσα θέση τους.

ABSTRACT

In this thesis, the Nearest Doctor Problem (NDP) for finding the closest doctor in case of an emergency is defined and a secure multi-party computation for solving it is presented. The solution is based on a privacy-preserving cryptographic protocol and makes use of the current location of each participating doctor. The protocol is efficient and protects the privacy of the location of all doctors. A prototype implementing the proposed solution for a community of doctors that use mobile devices to obtain their current location is presented.

Περιεχόμενα

Πίνακας Σχημάτων

Πίνακας Πινάκων

Κεφάλαιο 1

Εισαγωγή

Η ανάπτυξη τεχνολογιών πληροφορικής και επικοινωνιών (ICT = Information and Communication Technologies) και η ευρεία αποδοχή των ηλεκτρονικών συναλλαγών στις καθημερινές δραστηριότητες των ατόμων έχει σημαντική επίδραση τόσο στη χρήση όσο και στην προστασία των προσωπικών δεδομένων. Η περαιτέρω εξέλιξη των υπολογιστών γραφείου, των φορητών συσκευών και αισθητήρων, καθώς και η πρόοδος που παρουσίασαν οι βάσεις δεδομένων και οι τεχνολογίες αποθήκευσης έχουν οδηγήσει στην αύξηση της ποσότητας των προσωπικών πληροφοριών που παράγονται, ενδεχομένως (μόνιμα) αποθηκεύονται και επεξεργάζονται. Οποιοδήποτε είδος προσωπική πληροφορία προέρχεται άμεσα ή έμμεσα από οποιαδήποτε ηλεκτρονική δραστηριότητα των ατόμων, είτε προσωπική είτε επαγγελματική, ανήκει στην κατηγορία των προσωπικών δεδομένων και είναι αναγκαίο να προστατευθεί προκειμένου να διασφαλιστεί η ιδιωτικότητα του ατόμου.

Η τεχνολογική πρόοδος, εκτός από την αύξηση της παραγωγής προσωπικών δεδομένων, παρέχει και τη δυνατότητα ανάπτυξης καινοτόμων εφαρμογών που μπορούν να χρησιμοποιούν προσωπικά δεδομένα προς όφελος του ίδιου του ατόμου. Όπως για παράδειγμα, εξατομικευμένες διαδικτυακές υπηρεσίες που προσαρμόζονται αυτόματα ανάλογα με τις προτιμήσεις του κάθε χρήστη και υπηρεσίες που βασίζονται στην εκάστοτε θέση των ατόμων για να παρέχουν μια δεδομένη λειτουργία. Το άτομο μεμονωμένα, καθώς και η ολόκληρη η κοινωνία, μπορεί να αποκομίσει σημαντικά οφέλη εάν τα προσωπικά δεδομένα μπορούν να χρησιμοποιηθούν νόμιμα και ταυτόχρονα να διασφαλίζεται η προστασία τους. Κάθε άτομο έχει το δικαίωμα να προστατεύσει την ιδιωτικότητα του με τη διατήρηση του απόλυτου ελέγχου των προσωπικών του δεδομένων και τη γνώση για το ποιος, το πότε και το γιατί έχει πρόσβαση σε αυτά.

Επιπρόσθετα, κάθε άτομο πρέπει να αποκαλύπτει κάθε φορά μόνο τα ελάχιστα εκείνα προσωπικά δεδομένα που απαιτούνται για την πραγματοποίηση μιας συναλλαγής. Δηλαδή, η αποκάλυψη των προσωπικών δεδομένων πρέπει να γίνεται με τέτοιο τρόπο έτσι ώστε μόνο τα απολύτως απαραίτητα στοιχεία να αποκαλύπτονται και μόνο όταν αυτά απαιτούνται πραγματικά.

Μια πολύ ενδιαφέρουσα κατηγορία προσωπικών δεδομένων είναι και τα δυναμικά προσωπικά δεδομένα, όπως η τρέχουσα θέση (location) ενός ατόμου. Η τρέχουσα πρόοδος της τεχνολογίας των φορητών συσκευών και γενικά των τεχνολογιών απανταχού υπολογίζειν (Ubiquitous Computing) επιτρέπει στα άτομα να συλλέγουν τα δυναμικά προσωπικά δεδομένα προκειμένου να δίνεται η δυνατότητα πραγματοποίησης χρήσιμων υπολογισμών και η εξαγωγή χρήσιμων συμπερασμάτων.

Σε αυτή την εργασία, εστιάζεται η προσοχή μας στα δυναμικά προσωπικά δεδομένα και εξετάζεται η δυνατότητα ανάπτυξης καινοτόμων εφαρμογών που θα εκμεταλλεύονται αυτού του είδους τα δεδομένα, ενώ ταυτόχρονα θα εξασφαλίζεται η προστασία της ιδιωτικότητας των ατόμων. Για το λόγο αυτό, προτείνεται το πρόβλημα εύρεσης του κοντινότερου γιατρού (NDP = Nearest Doctor Problem), το οποίο βρίσκει τον κοντινότερο γιατρό σε μια περίπτωση εκτάκτου ανάγκης. Σε ένα υποθετικό αλλά ταυτόχρονα εφικτό σενάριο, κάθε γιατρός έχει στη διάθεση του έναν προσωπικό agent όπου ανά πάσα στιγμή γνωρίζει τη τρέχουσα γεωγραφική του θέση. Οπότε σε μια περίπτωση εκτάκτου ανάγκης, οι agents όλων των γιατρών μπορούν να υπολογίσουν και να βρουν (ανάμεσα τους) ποιος γιατρός τυγχάνει να βρίσκεται κοντινότερα στη θέση που συνέβη το επείγον περιστατικό. Επιπλέον, υποθέτουμε ότι οι γιατροί μπορεί να είναι εκτός υπηρεσίας και έτσι η τρέχουσα θέση του κάθε γιατρού να αποτελεί εξαιρετικά ευαίσθητο προσωπικό δεδομένο που δεν πρέπει να αποκαλυφθεί σε κανέναν (συμπεριλαμβανομένων και των άλλων γιατρών) και σε καμία περίπτωση.

Κεφάλαιο 2

Εισαγωγικές έννοιες - Υπόβαθρο

2.1 Ιδιωτικότητα

Η λέξη ιδιωτικότητα αποτελεί την απόδοση στην ελληνική, του αγγλικού όρου privacy. Η έννοια της λέξεως με την μετάφραση της, αποδίδεται ικανοποιητικά στην ελληνική, σε αντίθεση με ορισμένες γλώσσες που χρησιμοποιούν τον όρο στα αγγλικά. Αν και η ρίζα της λέξεως στα ελληνικά είναι αρχαία, η έννοια που περιγράφει είναι σχετικά νέα. Πρέπει να τονιστεί πως το "ιδιωτεύειν", εθεωρείτο ως κοινωνική αναπηρία στην αρχαία Ελλάδα, και "ιδιώτης" χαρακτηριζόταν ο μη μετέχων στα κοινά πολίτης. Ωστόσο, ο σημερινός όρος της ιδιωτικότητας, δεν εμπεριέχει αυτή την αρνητική έννοια.

Ένας τομέας της ιστορικής μελέτης (γενικά, αλλά και της ελληνικής εν προκειμένω), αποτελεί ο "Δημόσιος και ιδιωτικός βίος", το δεύτερο σκέλος του οποίου περιγράφει τον ιδιωτικό τομέα της ζωής των μελετώμενων πολιτισμών. Μέσω αυτού μπορεί να καταστεί σαφές, ότι η αρχαία έννοιας της λέξης ήταν διαφορετική με αυτή της σημερινής, καθώς δεν ήταν το δικαίωμα στην προσωπική ζωή που καυτηρίαζε ο όρος, αλλά η συμμετοχή ή όχι (πέραν της δεδομένης οικογενειακής - προσωπικής ζωής) στο δημόσιο βίο και στο πολιτικό γίγνεσθαι.

Η γέννηση της έννοιας της ιδιωτικότητας (privacy) συναντάται για πρώτη φορά στην εργασία των Samuel D. Warren και Louis D. Brandeis [1]. Στην εργασία αυτή, οι δύο Αμερικανοί δικηγόροι καθόρισαν την ιδιωτικότητα ως "το δικαίωμα να είσαι μόνος". Ο λόγος για αυτή τη δημοσίευση ήταν η ανάπτυξη νέων μορφών τεχνολογίας που συνδέθηκαν με ορισμένες εξελίξεις. Οι φωτογραφίες, για παράδειγμα, που χρησιμοποιούνταν από τον Κίτρινο Τύπο ήταν, σύμφωνα με την άποψη των συντακτών, μια επίθεση στην προσωπική ιδιωτικότητα σύμφωνα με την άποψη του δικαιώματος να είσαι

μόνος.

Ο πιο συνήθης ορισμός της ιδιωτικότητας που χρησιμοποιείται σήμερα, είναι αυτός του Alan Westin: "Η ιδιωτικότητα είναι η αξίωση των ατόμων, των ομάδων και των ιδρυμάτων, να αποφασίζουν από μόνοι τους για το πότε, πώς και μέχρι ποιο σημείο οι πληροφορίες που τους αφορούν, θα διαβιβάζονται σε άλλους" [2]. Ένας άλλος επίσης γνωστός όρος που χρησιμοποιείται για την προστασία των προσωπικών δεδομένων, εκτός του όρου της ιδιωτικότητας, και είναι επίσης σύμφωνος με τον ορισμό του Westin, είναι αυτός του πληροφοριακού αυτο-προσδιορισμού [3].

2.2 Προσωπικά δεδομένα

Τα προσωπικά δεδομένα [4] είναι κάθε πληροφορία που αναφέρεται και περιγράφει ένα άτομο, όπως: στοιχεία αναγνώρισης (ονοματεπώνυμο, ηλικία, κατοικία, επάγγελμα, οικογενειακή κατάσταση κλπ.), φυσικά χαρακτηριστικά, εκπαίδευση, εργασία (προϋπηρεσία, εργασιακή συμπεριφορά κλπ.), οικονομική κατάσταση (έσοδα, περιουσιακά στοιχεία, οικονομική συμπεριφορά), ενδιαφέροντα, δραστηριότητες και συνήθειες. Το άτομο (φυσικό πρόσωπο) στο οποίο αναφέρονται τα δεδομένα ονομάζεται υποκείμενο των δεδομένων.

Πέραν του γενικού χαρακτηρισμού ορισμένων δεδομένων ως προσωπικά, υπάρχει διάκριση μιας υποκατηγορίας αυτών που περιγράφονται με τον όρο "Ευαίσθητα προσωπικά δεδομένα". Ευαίσθητα χαρακτηρίζονται τα προσωπικά δεδομένα ενός ατόμου που αναφέρονται στη φυλετική ή εθνική του προέλευση, στα πολιτικά του φρονήματα, στις θρησκευτικές ή φιλοσοφικές του πεποιθήσεις, στη συμμετοχή του σε συνδικαλιστική οργάνωση, στην υγεία του, στην κοινωνική του πρόνοια, στην ερωτική του ζωή, στις ποινικές διώξεις και καταδίκες του, καθώς και στη συμμετοχή του σε συναφείς με τα ανωτέρω ενώσεις προσώπων. Τα ευαίσθητα προσωπικά δεδομένα προστατεύονται από το νόμο με αυστηρότερες ρυθμίσεις από ότι τα απλά προσωπικά δεδομένα.

Εκτός από τις δύο προαναφερθείσες κατηγορίες δεδομένων, οι οποίες περιγράφονται από το νομοθέτη [4], και θεσμικά υπάγονται υπό την εποπτεία της αρμόδιας ανεξάρτητης αρχής (στην Ελλάδα αυτή είναι η "Αρχή προστασίας δεδομένων προσωπικού χαρακτήρα"), υπάρχει και μια διαφορετική κατηγορία δεδομένων, τα λεγόμενα "Προσωπικά αναγνωρίσιμα δεδομένα" (PII = Personal Identifiable Information). Τα δεδομένα αυτά μπορεί να είναι μεμονωμένα, όπως π.χ. ο αριθμός ταυτότητας, ή συνδυασμοί, όπως π.χ. η ημερομηνία γέννησης και ο ταχυδρομικός κώδικας, και μπορούν να προσδιορίσουν με ακρίβεια ένα άτομο.

Πρέπει να τονιστεί ότι τα προσωπικά αναγνωρίσιμα δεδομένα δεν περιορίζονται μόνο στα προσωπικά δεδομένα. Ένα συχνά αναφερόμενο σχετικό παράδειγμα είναι ότι, το 87% του αμερικανικού πληθυσμού μπορεί να ταυτοποιηθεί από το συνδυασμό του φύλου, της ημερομηνίας γέννησης και του ταχυδρομικού κώδικα της κατοικίας του.

Από τα παραπάνω προκύπτει ένα κρίσιμο συμπέρασμα. Η γνώση μιας ομάδας πληροφοριών, όπως για παράδειγμα το ιστορικό των αναζητήσεων ενός άγνωστου χρήστη μιας μηχανής αναζήτησης, μπορεί να μας οδηγήσει στην εξαγωγή σημαντικών στοιχείων, ακόμη και ευαίσθητων προσωπικών δεδομένων του ατόμου. Δεδομένου ότι μπορεί να γίνει ταυτοποίηση του αρχικά άγνωστου χρήστη από μια ομάδα προσωπικά αναγνωρίσιμων δεδομένων που ίσως εισάγει στις αναζητήσεις του, μπορεί ακολούθως, από τη μορφή και το περιεχόμενο των αναζητήσεων του, να εξαχθούν πληροφορίες για τις πράξεις του αλλά ακόμη και για τις σκέψεις του. Από ένα ανώνυμο σύνολο ερωτημάτων δηλαδή, μπορεί σε πρώτο επίπεδο να υπάρξει αναγνώριση του δημιουργού των ερωτημάτων, και ακολούθως, σε δεύτερο επίπεδο εξόρυξη ευαίσθητων δεδομένων που τον αφορούν.

Τέλος, χάρη στις τελευταίες εξελίξεις στην ανάπτυξη φορητών συσκευών και γενικότερα τεχνολογιών Ubiquitous Computing, έχει δοθεί επιπλέον η δυνατότητα συλλογής "Δυναμικών προσωπικών δεδομένων". Σε αυτή την ειδική κατηγορία προσωπικών δεδομένων βρίσκονται δεδομένα όπως η γεωγραφική θέση (Location) και πληροφορίες που αφορούν τη σωματική κατάσταση και την υγεία ενός ατόμου, όπως η αρτηριακή πίεση, οι χτύποι της καρδίας και η διάθεση του. Αυτού του είδους τα δεδομένα, και συγκεκριμένα η τρέχουσα γεωγραφική θέση, είναι αυτά που θα μας απασχολήσουν κατά κύριο λόγο σε αυτή την εργασία.

2.3 Βασικές έννοιες της κρυπτογραφίας

2.3.1 Χρήση της κρυπτογραφίας στην ανταλλαγή δεδομένων

Η ανάγκη για την αποστολή ενός μηνύματος από κάποιον αποστολέα σε έναν παραλήπτη, χωρίς να υπάρχει κίνδυνος να διαβαστεί από κάποιον άλλον, δηλαδή αυτό το μήνυμα να αποσταλεί με ασφάλεια, αποτελεί ένα βασικό αντικείμενο της κρυπτογραφίας [5].

2.3.2 Βασική ιδέα της κρυπτογραφίας

Το αρχικό αυτό μήνυμα είναι το plaintext (ή cleartext, δηλ. καθαρό κείμενο). Η διαδικασία με την οποία στο μήνυμα αυτό αποκρύπτεται η πληροφορία του περιεχόμενου του ονομάζεται κρυπτογράφηση (encryption). Το κρυπτογραφημένο μήνυμα είναι το ciphertext. Η διαδικασία αλλαγής του ciphertext πίσω πάλι στο plaintext ονομάζεται αποκρυπτογράφηση (decryption). (Σημείωση: Σύμφωνα με το πρότυπο ISO 7498-2, χρησιμοποιούνται οι όροι "encipher" και "decipher" αντί για τους όρους "encrypt" και "decrypt", αντίστοιχα.) Όλα αυτά παρουσιάζονται στο Σχήμα 2.1.

Σχήμα 2.1 Κρυπτογράφηση και αποκρυπτογράφηση.

Η επιστήμη που έχει ως αντικείμενο την ασφάλεια των μηνυμάτων είναι η κρυπτογραφία (cryptography), και πραγματοποιείται από τους cryptographers (κρυπτογράφους). Οι Cryptanalysts ασχολούνται με την κρυπτανάλυση (cryptanalysis), το σπάσιμο του ciphertext, δηλαδή να μπορέσουν να δουν τι κρύβεται πίσω από τα κρυπτογραφημένα δεδομένα. Ο κλάδος της επιστήμης που ασχολείται με την κρυπτογραφία και την κρυπτανάλυση, είναι η κρυπτολογία (cryptology) και αυτοί που εξασκούν το επάγγελμα αυτό είναι οι cryptologists. Οι σύγχρονοι cryptologists εκπαιδεύονται γενικά σε θεωρητικό μαθηματικό επίπεδο. Στην πράξη με τον όρο κρυπτογραφία συνήθως αναφερόμαστε συνολικά στην κρυπτολογία [5].

Το plaintext δηλώνεται με το γράμμα M, από το μήνυμα, ή με το γράμμα P, από το plaintext. Το plaintext μπορεί να είναι ένα ρεύμα (stream) από bits, ένα αρχείο κειμένου, ένα αρχείο εικόνας, ένα ρεύμα ψηφιακής φωνής, μια ψηφιακή βίντεο εικόνα κλπ.. Όσον αφορά σε έναν υπολογιστή, το M είναι απλά δυαδικά δεδομένα. Το plaintext μπορεί να αποτελεί είτε τη μεταφορά είτε την αποθήκευση δεδομένων. Ουσιαστικά δηλαδή το M είναι το μήνυμα που κρυπτογραφείται.

Ας υποθέσουμε τώρα ότι το ciphertext δηλώνεται με το γράμμα C. Είναι επίσης δυαδικά δεδομένα, μερικές φορές του ίδιου μεγέθους με το M και κάποιες άλλες μπορεί και μεγαλύτερου. (Η κρυπτογράφηση όμως σε συνδυασμό με τη συμπίεση, μπορεί να οδηγήσει ώστε το C να είναι μικρότερο από το M, χωρίς όμως η κρυπτογράφηση να μπορεί να εκτελεστεί απευθείας.) Η συνάρτηση κρυπτογράφησης E, επιδρά πάνω στο

M για να παράγει το C. Η μαθηματική αυτή σχέση είναι:

$$E(M) = C$$

Στην αντίστροφη διαδικασία, η συνάρτηση αποκρυπτογράφησης D επιδρά πάνω στο C για να παραγάγει το M:

$$D(C) = M$$

Δεδομένου ότι η ουσία της κρυπτογράφησης και έπειτα της αποκρυπτογράφησης ενός μηνύματος είναι να ανακτηθεί το αρχικό plaintext, η ακόλουθη σχέση αποδεικνύει αυτό:

$$D(E(M)) = M$$

2.3.3 Οι έννοιες Authentication, Integrity και Nonrepudiation

Εκτός από την παροχή της εμπιστευτικότητας, το σύστημα της κρυπτογραφίας καλείται συχνά να παρέχει και άλλες υπηρεσίες όπως:

- **Authentication (Αυθεντικοποίηση).** Πρέπει να είναι σε θέση ο δέκτης να εξακριβώνει εάν το μήνυμα ανήκει πράγματι στον αποστολέα ή ότι ο αποστολέας είναι αυτός που ισχυρίζεται ότι είναι.

- **Integrity (Ακεραιότητα).** Πρέπει να είναι σε θέση ο δέκτης να ελέγχει εάν το μήνυμα δεν έχει τροποποιηθεί κατά τη μεταφορά, ώστε να μην μπορεί κάποιος εισβολέας να αντικαταστήσει το μήνυμα με ένα ψεύτικο και αυτό να φαίνεται ως αληθινό.

- **Nonrepudiation.** Ένας αποστολέας δεν μπορεί αργότερα να αρνηθεί ψευδώς ότι έστειλε το μήνυμα.

Αυτές οι έννοιες είναι ζωτικής σημασίας για την κοινωνική αλληλεπίδραση με τη χρήση υπολογιστών, και είναι ανάλογες με τις διαπροσωπικές αλληλεπιδράσεις. Ορισμένα παραδείγματα όπου μπορούν να εφαρμοστούν οι παραπάνω υπηρεσίες είναι:

- Ότι κάποιος είναι αυτός που λέει ότι είναι (Αυθεντικοποίηση).

- Ότι η άδεια ενός οδηγού, το ιατρικό δίπλωμα, και το διαβατήριο είναι έγκυρα (Υπογραφή).

- Ότι ένα έγγραφο είναι απόλυτα βέβαιο ότι έχει προέλθει από κάποιο πρόσωπο (Nonrepudiation).

- Ότι ένα έγγραφο δεν έχει αλλοιωθεί ή τροποποιηθεί (Ακεραιότητα).

2.3.4 Αλγόριθμοι και κλειδιά της κρυπτογραφίας

Ένας κρυπτογραφικός αλγόριθμος (αποκαλείται και ως cipher), είναι η μαθηματική συνάρτηση που χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση [5]. Γενικά, υπάρχουν δύο σχετικές συναρτήσεις: μια για την κρυπτογράφηση και μια άλλη για την αποκρυπτογράφηση.

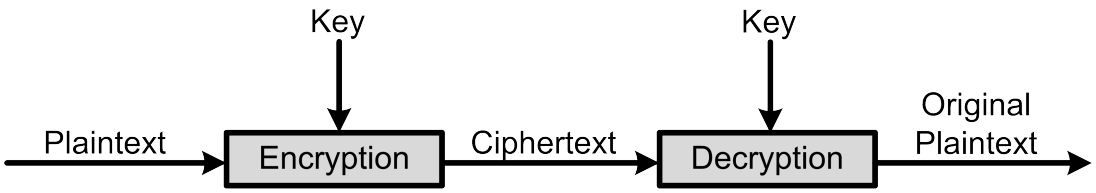
Εάν η ασφάλεια ενός αλγορίθμου είναι βασισμένη στο μυστικό τρόπο λειτουργίας του αλγόριθμου, τότε ένας τέτοιος αλγόριθμος είναι περιορισμένος (restricted). Τέτοιοι περιορισμένοι αλγόριθμοι έχουν μόνο ιστορικό ενδιαφέρον και θεωρούνται ανεπαρκείς για τα σημερινά πρότυπα. Μια μεγάλη ή μεταβαλλόμενη ομάδα χρηστών δεν μπορεί να τους χρησιμοποιήσει, επειδή κάθε φορά που ένας χρήστης αλλάζει ομάδα θα πρέπει να χρησιμοποιήσει και έναν διαφορετικό αλγόριθμο. Επίσης εάν κάποιος αποκαλύψει τυχαία το μυστικό, τότε ο καθένας θα πρέπει να αλλάξει τον αλγόριθμό.

Επιπλέον, οι περιορισμένοι αυτοί αλγόριθμοι δεν επιτρέπουν κανέναν ποιοτικό έλεγχο ή τυποποίηση. Κάθε ομάδα χρηστών πρέπει να έχει έναν μοναδικό αλγόριθμό. Μια τέτοια ομάδα δεν μπορεί να χρησιμοποιήσει έτοιμα προϊόντα υλικού ή λογισμικού, επειδή οποιοσδήποτε θα μπορούσε να αγοράσει το ίδιο προϊόν και να μάθει τον αλγόριθμο, οπότε θα πρέπει να γράψουν οι ίδιοι τους αλγορίθμους και τις εφαρμογές τους. Εάν κανένας στην ομάδα δεν είναι καλός cryptographer, δεν θα μπορούν να ξέρουν εάν ο αλγόριθμος τους είναι ασφαλής.

Παρόλα αυτά τα σημαντικά μειονεκτήματα, οι περιορισμένοι αλγόριθμοι είναι πάρα πολύ δημοφιλείς για εφαρμογές χαμηλής ασφάλειας. Στις περιπτώσεις αυτές οι χρήστες είτε δεν αντιλαμβάνονται είτε δεν ενδιαφέρονται για τα προβλήματα ασφάλειας που υπάρχουν στο σύστημά τους.

Τα σύγχρονα συστήματα κρυπτογραφίας λύνουν αυτό το πρόβλημα με τη χρήση ενός κλειδιού, που δηλώνεται εάς υποθέσουμε με το γράμμα K. Αυτό το κλειδί μπορεί να είναι οποιοδήποτε από ένα μεγάλο αριθμό τιμών. Η περιοχή των πιθανών τιμών του κλειδιού καλείται keyspace. Οπότε τώρα οι διαδικασίες της κρυπτογράφησης και της αποκρυπτογράφησης χρησιμοποιούν αυτό το κλειδί (δηλαδή, εξαρτώνται από αυτό το κλειδί και αυτό το γεγονός δηλώνεται από τον δείκτη K), έτσι οι συναρτήσεις τώρα γίνονται (δείτε το Σχήμα 2.2):

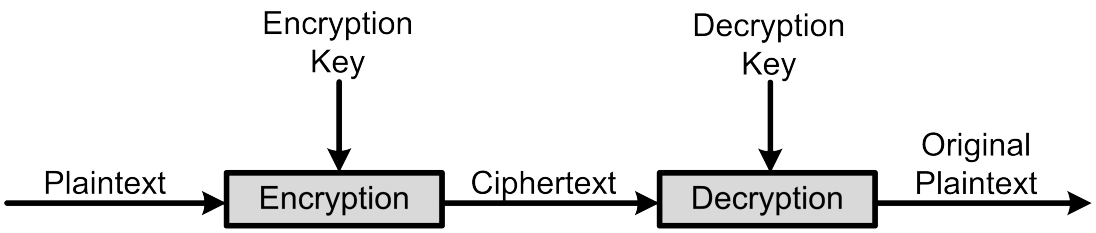
$$E_K(M) = C$$
$$D_K(C) = M$$
$$D_K(E_K(M)) = M$$



Σχήμα 2.2 Κρυπτογράφηση και αποκρυπτογράφηση με ένα κλειδί.

Μερικοί αλγόριθμοι χρησιμοποιούν διαφορετικά κλειδιά στην κρυπτογράφηση και στην αποκρυπτογράφηση (δείτε το Σχήμα 2.3). Δηλαδή το κλειδί κρυπτογράφησης, K_1, είναι διαφορετικό από το αντίστοιχο κλειδί αποκρυπτογράφησης, K_2. Σε αυτήν την περίπτωση οι συναρτήσεις γίνονται:

$$E_{K_1}(M) = C$$
$$D_{K_2}(C) = M$$
$$D_{K_2}(E_{K_1}(M)) = M$$



Σχήμα 2.3 Κρυπτογράφηση και αποκρυπτογράφηση με δύο διαφορετικά κλειδιά.

Όλη η ασφάλεια σε αυτούς τους αλγόριθμους είναι βασισμένη στο κλειδί (ή τα κλειδιά) και όχι στις λεπτομέρειες του αλγορίθμου. Αυτό σημαίνει ότι ο αλγόριθμος μπορεί να δημοσιευθεί και να αναλυθεί, οπότε και τα προϊόντα που χρησιμοποιούν τον αλγόριθμο μπορούν να παραχθούν μαζικά. Επίσης, δεν πειράζει εάν κάποιος ξέρει τον αλγόριθμό σας, επειδή δεν ξέρει το ιδιαίτερο κλειδί σας, δεν μπορεί να διαβάσει τα μηνύματά σας. Έτσι λοιπόν, ένα σύγχρονο κρυπτογραφικό σύστημα αποτελείται πλέον από τον αλγόριθμο, όλα τα πιθανά plaintexts, τα ciphertexts και τα κλειδιά.

2.3.5 Συμμετρικοί Αλγόριθμοι

Υπάρχουν δύο γενικοί τύποι αλγορίθμων βασισμένων σε κλειδιά: οι συμμετρικοί αλγόριθμοι και οι αλγόριθμοι δημόσιου-κλειδιού (public-key). Οι συμμετρικοί αλγόριθμοι,

οι οποίοι αποκαλούνται μερικές φορές και συμβατικοί αλγόριθμοι, είναι αλγόριθμοι όπου το κλειδί κρυπτογράφησης μπορεί να υπολογιστεί από το κλειδί αποκρυπτογράφησης και αντίστροφα. Στους περισσότερους συμμετρικούς αλγορίθμους το κλειδί κρυπτογράφησης και το κλειδί αποκρυπτογράφησης είναι τα ίδια. Αυτοί οι αλγόριθμοι αποκαλούνται επίσης και αλγόριθμοι μοναδικού-κλειδιού και απαιτούν ο αποστολέας και ο δέκτης να συμφωνούν σχετικά με ένα κλειδί προτού να μπορέσουν να επικοινωνήσουν με ασφάλεια. Η ασφάλεια ενός συμμετρικού αλγορίθμου στηρίζεται στο κλειδί, χωρίς να υπάρχει λόγος απόκρυψης των βασικών σημείων κρυπτογράφησης και αποκρυπτογράφησης. Εφόσον πρέπει να παραμείνει η επικοινωνία μυστική, το κλειδί πρέπει να παραμείνει μυστικό [5].

Η κρυπτογράφηση και η αποκρυπτογράφηση με έναν συμμετρικό αλγόριθμο δείχνονται από τις σχέσεις:

$$E_K(M) = C$$
$$D_K(C) = M$$

Οι συμμετρικοί αλγόριθμοι μπορούν να διαιρεθούν σε δύο κατηγορίες:

1. Σε αυτούς που το plaintext είναι ένα bit (ή μερικές φορές ένα byte) την κάθε φορά και ονομάζονται αλγόριθμοι ρευμάτων (stream) ή ρεύματα (stream) ciphers.

2. Και σε αυτούς που το plaintext είναι ομάδες από bits. Οι ομάδες των bits καλούνται blocks, και οι αλγόριθμοι αυτοί ονομάζονται αλγόριθμοι block ή blocks ciphers. Για τους σύγχρονους αλγορίθμους υπολογιστών, ένα χαρακτηριστικό μέγεθος blocks είναι 64 bits, αρκετά μεγάλο για να αποτρέψει την ανάλυση και αρκετά μικρό για να είναι εφαρμόσιμο.

2.3.6 Αλγόριθμοι δημόσιου-κλειδιού

Οι αλγόριθμοι δημόσιου-κλειδιού (public-key) (επίσης αποκαλούνται και ασύμμετροι αλγόριθμοι) σχεδιάστηκαν έτσι ώστε το κλειδί που χρησιμοποιείται για την κρυπτογράφηση να είναι διαφορετικό από το κλειδί που χρησιμοποιείται για την αποκρυπτογράφηση. Ακόμη, το κλειδί αποκρυπτογράφησης δεν μπορεί (τουλάχιστον σε λογικό χρονικό διάστημα) να υπολογιστεί από το κλειδί κρυπτογράφησης. Οι αλγόριθμοι αυτοί καλούνται ως "public-key" επειδή το κλειδί κρυπτογράφησης μπορεί να δημοσιοποιηθεί: Ένας άγνωστος μπορεί να χρησιμοποιήσει το κλειδί κρυπτογράφησης για να κρυπτογραφήσει ένα μήνυμα, αλλά μόνο ένα συγκεκριμένο πρόσωπο με το αντίστοιχο κλειδί αποκρυπτογράφησης μπορεί να αποκρυπτογραφήσει το μήνυμα. Σε αυτά τα συστήματα,

το κλειδί κρυπτογράφησης καλείται συχνά δημόσιο κλειδί, και το κλειδί αποκρυπτογράφησης καλείται συχνά ιδιωτικό κλειδί. Το ιδιωτικό κλειδί μερικές φορές επίσης καλείται μυστικό (secret) κλειδί, αλλά για να αποφευχθεί η σύγχυση με τους συμμετρικούς αλγόριθμους, αυτή η φράση γενικά δεν χρησιμοποιείται [5]. Η κρυπτογράφηση που χρησιμοποιεί το δημόσιο κλειδί K μπορεί να περιγραφεί με την ακόλουθη συνάρτηση:

$$E_{K_\delta}(M) = C$$

Η αποκρυπτογράφηση με το ιδιωτικό κλειδί περιγράφεται με τη συνάρτηση:

$$D_{K_\iota}(C) = M$$

Μερικές φορές, τα μηνύματα κρυπτογραφούνται με το ιδιωτικό κλειδί και αποκρυπτογραφούνται με το δημόσιο κλειδί. Συγκεκριμένα, αυτό χρησιμοποιείται στις ψηφιακές υπογραφές. Παρά την πιθανή σύγχυση, αυτές οι διαδικασίες περιγράφονται με τις συναρτήσεις:

$$E_{K_\iota}(M) = C$$
$$D_{K_\delta}(C) = M$$

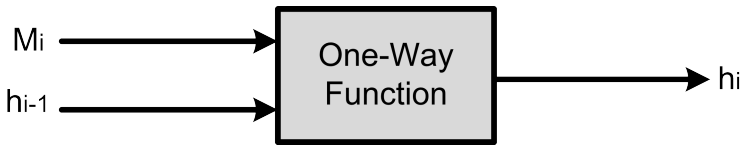
2.4 Αλγόριθμοι κρυπτογραφίας hash και SHA

2.4.1 Μονόδρομες hash συναρτήσεις

Μια hash συνάρτηση είναι μια συνάρτηση που δέχεται σαν είσοδο δεδομένα αυθαίρετου μήκους και τα μετατρέπει σε μονόδρομα δεδομένα (χωρίς να υπάρχει δυνατότητα επαναφοράς στην αρχική τους μορφή). Στην πραγματικότητα, οι μονόδρομες hash συναρτήσεις στηρίζονται στην ιδέα μιας συνάρτησης συμπίεσης. Αποτελέσματα αυτής της μονόδρομης συνάρτησης είναι μια μεταβλητή hash με μήκος n, για δεδομένη είσοδο μεγαλύτερου μήκους m. Οι είσοδοι της συνάρτησης συμπίεσης είναι ένα block μηνύματος και η έξοδος του προηγούμενου block από το κείμενο [5] (Σχήμα 2.4). Η έξοδος της συνάρτησης είναι το hash όλων των blocks μέχρι εκείνο το σημείο. Δηλαδή το hash του block M_i θα είναι:

$$h_i = f(M_i, h_{i-1})$$

Αυτή η hash μεταβλητή, μαζί με το επόμενο block μήνυμα, γίνεται η επόμενη είσοδος της συνάρτησης συμπίεσης. Το hash ολόκληρου του μηνύματος είναι το hash του τελευταίου block.



Σχήμα 2.4 Μονόδρομη hash συνάρτηση.

Η προ-εικόνα (δηλαδή, πριν ξεκινήσει ακόμη η διαδικασία) πρέπει να περιέχει κάποια δυαδική αντιπροσώπευση του μήκους ολόκληρου του μηνύματος. Αυτή η τεχνική ξεπερνά ένα πιθανό πρόβλημα ασφάλειας σε μηνύματα που έχουν ενδεχομένως διαφορετικά μήκη hashing για την ίδια μεταβλητή. Αυτή η τεχνική μερικές φορές καλείται και ως MD-strengthening.

Διάφοροι ερευνητές έχουν εξετάσει σε θεωρητικό επίπεδο ότι εάν η συνάρτηση συμπίεσης είναι ασφαλής, τότε και η μέθοδος hashing αυθαίρετου μήκους με προ-εικόνα είναι επίσης ασφαλής, αλλά τίποτα δεν έχει αποδειχθεί.

2.4.2 SHA hash συναρτήσεις

Η οικογένεια του SHA (Secure Hash Algorithm) είναι ένα σύνολο σχετικών κρυπτογραφικών hash συναρτήσεων. Η συνηθέστερα χρησιμοποιούμενη συνάρτηση της οικογένειας αυτής, ο SHA-1, υιοθετείται σε μια μεγάλη ποικιλία δημοφιλών εφαρμογών και πρωτοκόλλων ασφάλειας, συμπεριλαμβανομένου του TLS, SSL, PGP, SSH, S/MIME, και IPSec. Ο SHA-1 θεωρείται ο διάδοχος του MD5, μιας προηγούμενης, ευρέως χρησιμοποιούμενης hash συνάρτησης. Σε μερικές περιπτώσεις, με ιδιαίτερες απαιτήσεις ασφάλειας, προτείνεται η χρήση του SHA-256 ή μεγαλύτερου. Οι αλγόριθμοι SHA σχεδιάστηκαν από την NSA (National Security Agency) και δημοσιεύθηκαν ως πρότυπα της αμερικανικής κυβέρνησης.

Το πρώτο μέλος της οικογένειας, δημοσιεύτηκε το 1993 και ονομάζεται επίσημα SHA. Εντούτοις, καλείται συχνά ως SHA-0 για να αποφευχθεί η σύγχυση με τους διαδόχους του. Δύο έτη αργότερα δημοσιεύθηκε ο SHA-1, που είναι ο πρώτος διάδοχος του SHA. Από τότε, τέσσερις παραλλαγές του SHA έχουν δημοσιευτεί για να καλύψουν τις αυξανόμενες ανάγκες ασφαλείας: SHA-224, SHA-256, SHA-384, και SHA-512 (μερικές φορές συλλογικά αναφέρονται ως SHA-2). Στο Πίνακα 2.1 φαίνονται τα διάφορα είδη του SHA [6].

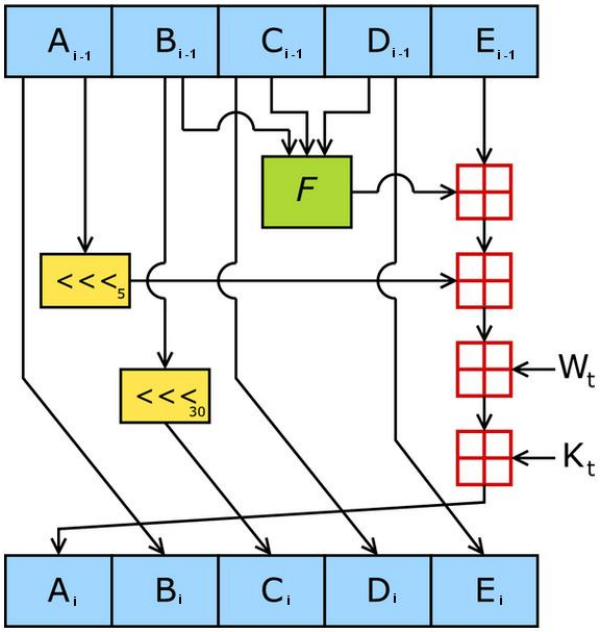
Algorithm	Output size	Internal state size	Block size	Length size	Word size	Passes	Operations	Collision
SHA-0	160	160	512	64	32	80	+,and,or,xor,rotl	Yes
SHA-1	160	160	512	64	32	80	+,and,or,xor,rotl	With flaws
SHA-256/224	256/224	256	512	64	32	64	+,and,or,xor,shr,rotr	No
SHA-512/384	512/384	512	1024	128	64	80	+,and,or,xor,shr,rotr	No

Πίνακας 2.1 Τα διάφορα μεγέθη του SHA [6].

2.4.3 SHA-1 και SHA-2

Ο SHA-1 παράγει μια συγχώνευση 160-bits από ένα μήνυμα με μέγιστο μέγεθος 2^{64} bits, και είναι βασισμένος σε αρχές παρόμοιες με εκείνες που χρησιμοποιήθηκαν από τους αλγόριθμους MD4 και MD5. Ο SHA-1 είναι ασφαλής επειδή έχει ως σκοπό να είναι υπολογιστικά ανέφικτο να ανακτηθεί ένα μήνυμα που αντιστοιχεί σε μια δεδομένη συγχώνευση μηνυμάτων, ή για να βρεθούν δύο διαφορετικά μηνύματα που να παράγουν την ίδια συγχώνευση μηνυμάτων. Οποιαδήποτε αλλαγή συμβεί σε ένα μήνυμα κατά τη μεταφορά, με μεγάλη πιθανότητα θα οδηγήσει σε μια διαφορετική συγχώνευση μηνυμάτων, και ο έλεγχος της υπογραφής θα αποτύχει. Στο Σχήμα 2.5 φαίνεται μία επανάληψη της συνάρτησης συμπίεσης του SHA-1. Εκεί τα A, B, C, D και E είναι λέξεις των 32-bits μιας κατάστασης, το F είναι μια μη γραμμική συνάρτηση, το $<<<_n$ δηλώνει μια αριστερή μετατόπιση κατά n-bits, το + δείχνει την πρόσθεση modulo 2^{32}, το i-1 είναι η παρούσα κατάσταση, ενώ το i η επόμενη κατάσταση, και το K_t είναι μια σταθερά.

Η οικογένεια του SHA-2 χρησιμοποιεί παρόμοια λογική με τον SHA-1 με την διαφορά ότι ανάλογα με την έκδοση του χρησιμοποιεί λέξεις των 32-bits (στον SHA-256 και SHA-224) και των 64-bits (στον SHA-512 και SHA-384). Επιπρόσθετα, πραγματοποιεί μετατόπιση διαφορετικού αριθμού από bits και χρησιμοποιεί διαφορετικές τιμές σταθερών (ανάλογα με την έκδοση), χωρίς όμως η βασική δομή του να διαφέρει ουσιαστικά [6].



Σχήμα 2.5 Μια επανάληψη της συνάρτησης συμπίεσης του SHA-1.

2.4.4 Ασφάλεια του SHA

Επιθέσεις έχουν βρεθεί για τον SHA-0 και τον SHA-1, όπως φαίνεται άλλωστε και στη στήλη "Collision" του Πίνακα 2.1. Καμία επίθεση δεν έχει αναφερθεί ακόμα για τις διάφορες παραλλαγές του SHA-2, αλλά δεδομένου ότι είναι παρόμοιοι με τον SHA-1 οι ερευνητές ανησυχούν. Για αυτό το λόγο αναπτύσσουν νέα καλύτερα hashing πρότυπα του SHA. Ένα νέο hashing πρότυπο, ο SHA-3, αυτή τη στιγμή βρίσκεται στο στάδιο της ανάπτυξης και αναμένεται να έχει ολοκληρωθεί στο τέλος του 2012.

2.5 Κρυπτογραφικό Σύστημα RSA

Ένα από τα πιο δημοφιλή κρυπτογραφικά συστήματα είναι το κρυπτογραφικό σύστημα RSA [7], που επινοήθηκε το 1978 από τους Ronald Rivest, Adi Shamir και Leonard Adelman και πήρε το όνομά του από τα αρχικά των επιθέτων τους. Το RSA είναι ένα κρυπτογραφικό σύστημα με δημόσιο κλειδί και μέχρι σήμερα θεωρείται αδύνατο να σπάσει με τη χρήση σύγχρονων υπολογιστών, εάν όμως κάποια στιγμή κατασκευαστούν κβαντικοί υπολογιστές αυτό ενδέχεται να αλλάξει ριζικά. Σήμερα χρησιμοποιείται ευρύτατα κυρίως στις οικονομικές και στις τραπεζικές συναλλαγές.

2.5.1 Αλγόριθμος RSA

Ο αλγόριθμος RSA αποτελείται από τρία κύρια μέρη: τη δημιουργία κλειδιών, τον αλγόριθμο κρυπτογράφησης και τον αλγόριθμο αποκρυπτογράφησης.

2.5.1.1 Δημιουργία κλειδιών

Η διαδικασία που πρέπει να ακολουθηθεί για τη δημιουργία ενός ζεύγους (δημόσιου και ιδιωτικού) κλειδιών είναι η εξής:

1. Επιλέγουμε δύο ακέραιους πρώτους αριθμούς τον p και q και υπολογίζουμε το γινόμενο τους $n = p \cdot q$.
2. Επιλέγουμε ένα τυχαίο αριθμό τον d ο οποίος είναι πρώτος ως προς τους $(p-1)$ και $(q-1)$, δηλαδή ο μέγιστος κοινός διαιρέτης των d, $(p-1)$ και $(q-1)$ είναι το ένα.
3. Υπολογίζουμε τον αριθμό e από τη σχέση: $(e \cdot d)\ mod\ (p-1)(q-1) = 1$. Δηλαδή, ο e είναι ο αντίστροφος του d, $mod\ (p-1)(q-1)$.
4. Το ζεύγος των αριθμών (e, n) είναι το δημόσιο κλειδί.
5. Το ζεύγος των αριθμών (d, n) είναι το ιδιωτικό κλειδί.

2.5.1.2 Αλγόριθμος κρυπτογράφησης

Ο αλγόριθμος που ακολουθείτε για την κρυπτογράφηση ενός μηνύματος είναι η εξής:

1. Το δημόσιο κλειδί στέλνεται στον αποστολέα του μηνύματος.
2. Ο αποστολέας κρυπτογραφεί το μήνυμα με το δημόσιο κλειδί (e, n).
3. Το κρυπτογραφημένο μήνυμα $E_k(m) = m^e\ mod\ n$ αποστέλεται στον παραλήπτη.

2.5.1.3 Αλγόριθμος αποκρυπτογράφησης

Ο αλγόριθμος που ακολουθείτε για την αποκρυπτογράφηση του αρχικού μηνύματος είναι ο εξής:

1. Το ιδιωτικό κλειδί (d, n) το κρατά ο παραλήπτης.

2. Για να αποκρυπτογραφηθεί το μήνυμα, χρειάζεται και το δημόσιο και το ιδιωτικό κλειδί.

3. Το αποτέλεσμα της αποκρυπτογράφησης θα είναι $D_k(c) = c^d \ mod \ n$.

2.5.2 Ασφάλεια του RSA

Για να καταλάβουμε καλύτερα την ασφάλεια [7] που μας παρέχει το κρυπτογραφικό σύστημα RSA θα αναφέρουμε πρώτα πως μπορούμε να το σπάσουμε, δηλαδή τι πρέπει να κάνουμε, για να αποκρυπτογραφήσουμε ένα μήνυμα που κρυπτογραφήθηκε με το σύστημα RSA. Μπορούμε πολύ εύκολα να βρούμε το δημόσιο κλειδί, δηλαδή το ζεύγος των αριθμών (e, n). Αφού τώρα γνωρίζουμε τον αριθμό n, δεν έχουμε παρά να τον αναλύσουμε σε γινόμενο δύο πρώτων αριθμών για να βρούμε τους αριθμούς p και q. Μόλις τους βρούμε, η αποκρυπτογράφηση γίνεται αμέσως, αφού η μέθοδος του συστήματος RSA είναι γνωστή.

Ενώ είναι πολύ εύκολο να πολλαπλασιάσουμε δύο πρώτους αριθμούς για να βρούμε το γινόμενό τους, είναι πάρα πολύ δύσκολο να αναλύσουμε έναν αριθμό σε γινόμενο δύο πρώτων αριθμών και είναι πρακτικά αδύνατον αν ο αριθμός έχει πολλά ψηφία. Έτσι λοιπόν, οι πρώτοι αριθμοί p και q θα πρέπει να είναι αρκετά μεγάλοι, ώστε ο καλύτερος γνωστός αλγόριθμος παραγοντοποίησης να απαιτεί χρόνο μεγαλύτερο από αυτόν με τον οποίο πρέπει να προστατευθούν τα δεδομένα. Στον Πίνακα 2.2 παρουσιάζονται ενδεικτικά μεγέθη και αντίστοιχες περιπτώσεις στις οποίες θα πρέπει να εφαρμοσθούν τα μεγέθη αυτά.

Για να αποδείξουν ότι το κρυπτογραφικό τους σύστημα δεν μπορεί να σπάσει, οι Rivest, Shamir και Adelman ζήτησαν από όποιο νομίζει ότι μπορεί, να αναλύσει σε γινόμενο δύο πρώτων αριθμών έναν ακέραιο με 129 ψηφία. Μετά από 17 χρόνια ο αριθμός αναλύθηκε από ένα δίκτυο 1.600 υπολογιστών. Έτσι λοιπόν με τα σημερινά τεχνολογικά δεδομένα, το πρόβλημα της ανάλυσης ενός αριθμού σε γινόμενο δύο πρώτων αριθμών είναι αδύνατον να λυθεί με τη χρήση σύγχρονων υπολογιστών και πόσο μάλλον όταν ο αριθμός έχει πολλά ψηφία.

2.6 Κρυπτογραφικό Σύστημα ElGamal

Το σύστημα κρυπτογράφησης ElGamal είναι ένας ασύμμετρος αλγόριθμος κρυπτογράφησης δημοσίου κλειδιού και βασίζεται στη βασική ιδέα των Diffie-Hellman. Για πρώτη φορά περιγράφτηκε από τον Taher Elgamal το 1984. Η ασφάλεια της κρυπτογράφησης

p, q	n	χρόνος προστασίας	τύπος δεδομένων
256 bits	512 bits	μερικές εβδομάδες	πληροφορίες που επηρεάζουν βραχυπρόθεσμα το χρηματιστήριο (π.χ. απόφαση συγχώνευσης δύο εταιρειών)
512 bits	1024 bits	50-100 χρόνια	προσωπικά μυστικά
1024 bits	2048 bits	> 100 χρόνια	εμπορικά μυστικά, προσωπικά δεδομένα
2048 bits	4096 bits	ηλικία του Σύμπαντος	στρατιωτικά μυστικά

Πίνακας 2.2 Μεγέθη παραμέτρων RSA και ενδεικτικοί τύποι δεδομένων προς προστασία [8].

του ElGamal βασίζεται στο πρόβλημα του Διακριτού Λογάριθμου. Στις ενότητες που ακολουθούν γίνεται αναλυτική περιγραφή του αλγορίθμου [9].

2.6.1 Αλγόριθμος ElGamal

Ο αλγόριθμος ElGamal αποτελείται από τρία κύρια μέρη: τη δημιουργία κλειδιών, τον αλγόριθμο κρυπτογράφησης, και τον αλγόριθμο αποκρυπτογράφησης.

2.6.1.1 Δημιουργία κλειδιών

Η διαδικασία που πρέπει να ακολουθηθεί για τη δημιουργία ενός ζεύγους (δημόσιου και ιδιωτικού) κλειδιών είναι η εξής:

1. Επιλέγουμε ένα τυχαίο μεγάλο και πρώτο αριθμό p και ένα πρωταρχικό στοιχείο/γεννήτορα g από το σύνολο ${Z_p}^*$, όπου ${Z_p}^*$ συμβολίζουμε το σύνολο όλων των ακέραιων $\{1, 2, \ldots, p-1\}$, δηλαδή $g^k \neq 1 \; mod \; p$ για όλα τα k μικρότερα του $p-1$.

2. Επιλέγουμε ένα τυχαίο αριθμό a στο διάστημα $1 \leq a \leq p-1$ ως το ιδιωτικό κλειδί.

3. Υπολογίζουμε το $y = g^a \; mod \; p$.

4. Το δημόσιο κλειδί είναι το (p, g, y) και το ιδιωτικό κλειδί είναι το a.

Για την εύρεση του γεννήτορα g θα πρέπει να σημειώσουμε τα εξής:

- Αν ισχύει το $g^k = 1 \ mod \ p$ για κάποιον ακέραιο αριθμό $1 \leq k \leq p-1$, τότε ο αριθμός k υποχρεωτικά διαιρεί τον $p-1$.
- Άρα, αν θέλουμε να ελέγξουμε αν ένας αριθμός g είναι γεννήτορας $mod \ p$, δεν χρειάζεται να τον υψώσουμε σε όλες τις δυνάμεις $\{1, 2, \ldots, p-1\}$, αρκεί να τον υψώσουμε στους διαιρέτες του $p-1$.

2.6.1.2 Αλγόριθμος κρυπτογράφησης

Ο αλγόριθμος που ακολουθείτε για την κρυπτογράφηση ενός μηνύματος είναι η εξής:

1. Επιλέγουμε έναν τυχαίο αριθμό r από το σύνολο $\{1, 2, \ldots, p-1\}$.
2. Εκφράζουμε το μήνυμα με έναν ακέραιο αριθμό m από τους $\{1, 2, \ldots, p-1\}$.
3. Υπολογίζουμε το $\gamma = g^r \ mod \ p$ και το $\delta = my^r \ mod \ p$.
4. Οπότε το ciphertext είναι το $E_k(m, r) = (\gamma, \delta)$.

2.6.1.3 Αλγόριθμος αποκρυπτογράφησης

Ο αλγόριθμος που ακολουθείτε για την αποκρυπτογράφηση του αρχικού μηνύματος είναι ο εξής:

1. Υπολογίζουμε το γ^{-a}, αφού γνωρίζουμε το ιδιωτικό κλειδί.
2. Το αρχικό μήνυμα θα είναι το $m = (\gamma^{-a})\delta \ mod \ p$.
3. Με άλλα λόγια η ανάκτηση του αρχικού μηνύματος γίνεται με την πράξη $\frac{\delta}{\gamma^a}$.
4. Οπότε το αρχικό plaintext είναι το $D_k(\gamma, \delta) = m \ (mod \ p)$.

2.6.2 Ασφάλεια του ElGamal

Ο αντίπαλος που θα επιχειρήσει επίθεση στο κρυπτοσύστημα [8], θα πρέπει να ανακτήσει το ιδιωτικό κλειδί a, από τη σχέση:

$$y = g^a \ mod \ p$$

γνωρίζοντας τα p, g, y. Θα πρέπει δηλαδή να λύσει το διακριτό λογάριθμο με βάση g. Ωστόσο, θεωρούμε ότι η ασφάλεια του κρυπτοσυστήματος ElGamal βασίζεται στο διακριτό λογάριθμο, διότι η λύση του διακριτού λογάριθμου μπορεί να καθιστά το κρυπτοσύστημα ανασφαλές, αλλά δεν έχει αποδειχθεί το αντίστροφο, ότι δηλαδή η ασφάλεια του κρυπτοσυστήματος στηρίζεται αποκλειστικά στο πρόβλημα του διακριτού λογάριθμου.

Η ύπαρξη του τυχαίου αριθμού r, έχει ως αποτέλεσμα τη δυνατότητα αντιστοίχισης του απλού κειμένου σε $p-1$ κρυπτοκείμενα. Η διαδικασία όπου το απλό κείμενο αναμειγνύεται με μια τυχαία μεταβλητή, ονομάζεται διαδικασία δημιουργίας συνθηκών τυχαιότητας (randomization process). Το βήμα αυτό το οποίο δεν υπάρχει στο RSA, καθιστά το κρυπτοσύστημα ElGamal ανθεκτικότερο σε επιθέσεις παρόμοιες με αυτές που παρουσιάζονται στο RSA. Βέβαια, η χρήση του τυχαίου αριθμού εισάγει έναν επιπλέον κίνδυνο που οδηγεί σε μια πρόσθετη απαίτηση. Για κάθε μήνυμα που κρυπτογραφείται, θα πρέπει να επιλέγεται διαφορετικός τυχαίος r. Στην περίπτωση που δύο μηνύματα m και m' κρυπτογραφηθούν με τον ίδιο r, τότε για τα αντίστοιχα κρυπτοκείμενα που θα προκύψουν (γ, δ) και (γ', δ'), η γνώση του ενός μηνύματος επιτρέπει την ανάκτηση του άλλου από τον λόγο:

$$\frac{\delta}{\delta'} = \frac{m \cdot y^r}{m' \cdot y^r} = \frac{m}{m'}$$

Τέλος, όσον αφορά το μέγεθος του p, το κατώτατο όριο που προτείνεται είναι 1024 bits. Γενικά, κατά την κρυπτογράφηση με το κρυπτοσύστημα ElGamal, το μέγεθος των παραμέτρων αποτελεί σημαντικό κριτήριο υλοποίησης, λόγω του αυξημένου χρόνου που απαιτείται για την κρυπτογράφηση (δύο πράξεις ύψωσης σε δύναμη έναντι της μιας στην περίπτωση του RSA), και λόγω της διαστολής του κρυπτοκειμένου. Τα μειονεκτήματα αυτά έχουν σαν αποτέλεσμα να προτιμάται μειωμένο μέγεθος του modulus.

2.7 Ομομορφική κρυπτογράφηση (Homomorphic encryption)

Η Ομομορφική κρυπτογράφηση (Homomorphic encryption) [10] είναι μια μορφή κρυπτογράφησης που μπορεί να εκτελέσει μια συγκεκριμένη αλγεβρική πράξη στο plaintext με την εκτέλεση μιας (ενδεχομένως διαφορετικής) αλγεβρικής πράξης στο ciphertext. Αυτή η ιδιότητα μπορεί να έχει τόσο θετικές όσο και αρνητικές επιπτώσεις σε ένα κρυπτογραφικό σύστημα. Έτσι λοιπόν, το μοντέλο της ομομορφικής κρυπτογράφησης

είναι ευάλωτο σε κακόβουλες επιθέσεις από το σχεδιασμό της, με αποτέλεσμα να είναι ακατάλληλο για την ασφαλή μετάδοση δεδομένων. Όμως, η ομομορφική ιδιότητα των διάφορων κρυπτογραφικών συστημάτων μπορεί να χρησιμοποιηθεί για τη δημιουργία ασφαλών εκλογικών συστημάτων, ανθεκτικών hash συναρτήσεων και μοντέλων ιδιωτικής ανάκτησης πληροφοριών (private information retrieval).

Παρακάτω παρουσιάζονται διάφορα αποδοτικά ομομορφικά κρυπτογραφικά συστήματα μαζί με την αντίστοιχη ομομορφική ιδιότητα που παρουσιάζουν:

- **Unpadded RSA:** Εάν το δημόσιο κλειδί είναι (e, n) τότε η κρυπτογράφηση ενός μηνύματος x θα δίνεται από $\mathcal{E}(x) = x^e \ mod \ m$. Τότε η ομομορφική ιδιότητα που παρουσιάζει είναι:

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = x_1^e x_2^e \ mod \ m = (x_1 x_2)^e \ mod \ m = \mathcal{E}(x_1 \cdot x_2)$$

- **ElGamal:** Εάν το δημόσιο κλειδί είναι (p, g, y) και το ιδιωτικό κλειδί είναι το a, όπου $y = g^a \ mod \ p$, τότε η κρυπτογράφηση ενός μηνύματος x θα δίνεται από $\mathcal{E}(x) = (g^r, my^r)$, όπου r ένας τυχαίος αριθμός στο σύνολο $\{1, 2, \ldots, p-1\}$. Τότε η ομομορφική ιδιότητα που παρουσιάζει είναι:

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = (g^{r_1}, x_1 \cdot y^{r_1})(g^{r_2}, x_2 \cdot y^{r_2}) = (g^{r_1+r_2}, (x_1 \cdot x_2)y^{r_1+r_2}) = \mathcal{E}(x_1 \cdot x_2)$$

- **Goldwasser-Micali:** Εάν το δημόσιο κλειδί είναι modulus m και ένα τετραγωνικό μη-υπόλοιπο (quadratic non-residue) x, τότε η κρυπτογράφηση ενός bit b θα δίνεται από $\mathcal{E}(b) = r^2 x^b \ mod \ m$. Τότε η ομομορφική ιδιότητα που παρουσιάζει είναι:

$$\mathcal{E}(b_1) \cdot \mathcal{E}(b_2) = r_1^2 x^{b_1} r_2^2 x^{b_2} = (r_1 r_2)^2 x^{b_1+b_2} = \mathcal{E}(b_1 \oplus b_2)$$

  όπου $\oplus$ δηλώνει ένα επιπλέον modulo 2 (π.χ. αποκλειστικό ή (XOR)).

- **Benaloh:** Εάν το δημόσιο κλειδί είναι modulus m και μια βάση g με ένα blocksize r, τότε η κρυπτογράφηση ενός μηνύματος x θα δίνεται από $\mathcal{E}(b) = g^x u^r \ mod \ m$. Τότε η ομομορφική ιδιότητα που παρουσιάζει είναι:

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = (g^{x_1} u_1^r)(g^{x_2} u_2^r) = g^{x_1+x_2}(u_1 u_2)^r = \mathcal{E}(x_1 + x_2 \ mod \ r)$$

- **Paillier:** Εάν το δημόσιο κλειδί είναι modulus m και μια βάση g, τότε η κρυπτογράφηση ενός μηνύματος x θα δίνεται από $\mathcal{E}(x) = g^x r^m \ mod \ m^2$. Τότε η ομομορφική ιδιότητα που παρουσιάζει είναι:

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = (g^{x_1} r_1^m)(g^{x_2} r_2^m) = g^{x_1+x_2}(r_1 r_2)^m = \mathcal{E}(x_1 + x_2 \ mod \ m)$$

Τέλος, αξίζει να αναφερθεί ότι υπάρχει μια πρόσφατη εργασία του Craig Gentry [11] που παρουσιάζει μια πλήρης ομομορφική κρυπτογράφηση (fully homomorphic encryption). Αυτή η πλήρης ομομορφική κρυπτογράφηση, παρουσιάζει ταυτόχρονα τόσο την ιδιότητα της πρόσθεσης όσο και του πολλαπλασιασμού με αποτέλεσμα να είναι σε θέση να μπορεί να πραγματοποιήσει σύνθετες πράξεις πάνω στα ciphertexts.

2.8 Πιστοποιητικό δημοσίου κλειδιού (Public key certificate)

Στην κρυπτογραφία, το πιστοποιητικό δημοσίου κλειδιού (ή πιστοποιητικό ταυτότητας) [12] είναι ένα πιστοποιητικό που χρησιμοποιεί μια ψηφιακή υπογραφή για να δεσμεύσει μαζί ένα δημόσιο κλειδί με μια ταυτότητα (δηλ., πληροφορίες όπως το όνομα ενός προσώπου ή ενός οργανισμού, τη διεύθυνσή τους, κλπ.). Το πιστοποιητικό αυτό μπορεί να χρησιμοποιηθεί για να ελέγξει εάν ένα δημόσιο κλειδί ανήκει σε ένα άτομο.

Σε ένα τυπικό σχήμα υποδομής δημοσίου κλειδιού (PKI = Public Key Infrastructure), η υπογραφή θα είναι μιας αρχής πιστοποίησης (CA = Certificate Authority). Σε έναν σχήμα εμπιστοσύνης WEB, η υπογραφή είναι είτε του χρήστη (ένα self-signed certificate), είτε άλλων χρηστών ("επικυρώσεις"). Σε καθεμία περίπτωση, οι υπογραφές σε ένα πιστοποιητικό είναι επιβεβαιώσεις από τον υπογράφοντα πιστοποιητικών ότι οι πληροφορίες ταυτότητας και του δημοσίου κλειδιού ανήκουν στον ίδιο. Το πιο κοινό πρότυπο πιστοποιητικών είναι το ITU-T X.509.

Ένα πιστοποιητικό μπορεί να ανακληθεί εάν ανακαλυφθεί ότι το σχετικό ιδιωτικό κλειδί του έχει παραχωρηθεί, ή εάν η συγγένεια (μεταξύ μιας οντότητας και ενός δημόσιου κλειδιού) που ενσωματώνεται στο πιστοποιητικό ανακαλύπτεται ότι είναι ανακριβής ή έχει αλλάξει (π.χ., εάν ένα πρόσωπο αλλάζει τη δουλειά ή το όνομα). Εάν και η ανάκληση είναι ένα σπάνιο περιστατικό, ο χρήστης πρέπει πάντα να ελέγχει την ισχύ του πιστοποιητικού. Αυτό μπορεί να γίνει με τη σύγκριση του πιστοποιητικού με έναν κατάλογο ανάκλησης πιστοποιητικών (CRL = Certificate Revocation List), δηλαδή με ένα κατάλογο ανακλημένων ή ακυρωμένων πιστοποιητικών. Ένας άλλος τρόπος να ελεγχθεί η ισχύς των πιστοποιητικών είναι να ερωτηθεί η αρχή των πιστοποιητικών (CA) που χρησιμοποιεί το Online Certificate Status Protocol (OCSP).

Τα βασικά χαρακτηριστικά από τα οποία αποτελείται ένα πιστοποιητικό είναι τα εξής:

- Το δημόσιο κλειδί που υπογράφεται.

- Ένα όνομα, το οποίο μπορεί να αναφέρεται σε ένα πρόσωπο, έναν υπολογιστή ή ενός οργανισμού.
- Μια περίοδος ισχύος.
- Τη διεύθυνση (URL) ενός κέντρου ανάκλησης (revocation center).

2.9 Πρωτόκολλο ασφαλής επικοινωνίας TLS/SSL

Το SSL (Secure Sockets Layer) και ο διάδοχός του TLS (Transport Layer Security) [13], είναι κρυπτογραφικά πρωτόκολλα που παρέχουν ασφαλή επικοινωνία στο διαδίκτυο για ειδικές περιπτώσεις όπως το ηλεκτρονικό ταχυδρομείο (e-mail), το internet fax, και για άλλες μεταφορές δεδομένων. Υπάρχουν μικρές διαφορές μεταξύ του SSL 3.0 και του TLS 1.0, αλλά το πρωτόκολλο παραμένει ουσιαστικά το ίδιο. Ο όρος "SSL" όπως χρησιμοποιείται εδώ ισχύει και για τα δύο πρωτόκολλα.

Το SSL παρέχει πιστοποίηση και απομονωμένη επικοινωνία μέσω του διαδικτύου χρησιμοποιώντας κρυπτογραφία. Σε μια τυπική χρήση του, μόνο ο server πιστοποιείται (δηλ. η ταυτότητά του κατοχυρώνεται) ενώ ο client παραμένει ανώνυμος. Σε μια αμοιβαία πιστοποίηση απαιτείται επιπλέον και σχήμα υποδομής δημοσίου κλειδιού (PKI) στους clients. Τα πρωτόκολλα αυτά επιτρέπουν εφαρμογές επικοινωνίας client/server με τέτοιο τρόπο ώστε να αποτρέπεται η παρακολούθηση και η παραποίηση των μηνυμάτων.

Το SSL αποτελείται από τρεις βασικές φάσεις:

1. Διαπραγμάτευση για την υποστήριξη του αλγόριθμου.
2. Δημόσιο κλειδί κρυπτογράφησης (βασισμένο σε κλειδί ανταλλαγής) και πιστοποιητικό (βασισμένο σε κάποια πιστοποίηση).
3. Συμμετρική κρυπτογραφία (βασισμένη σε κρυπτογράφηση επικοινωνίας).

Κατά τη διάρκεια της πρώτης φάσης, η διαπραγμάτευση client και server μπορεί να γίνει για διάφορους κρυπτογραφικούς αλγορίθμους. Οι αλγόριθμοι που υποστηρίζονται για την κάθε φάση είναι:

- Για δημόσιου κλειδιού κρυπτογράφηση: RSA, Diffie-Hellman, DSA ή Fortezza.
- Για συμμετρική κρυπτογραφία: RC2, RC4, IDEA, DES, Triple DES ή AES.
- Για μονόδρομες hash συναρτήσεις: MD5 ή SHA.

2.10 Ασφαλής εκτέλεση υπολογισμών (Secure multi-party computation)

Στην κρυπτογραφία, η ασφαλής εκτέλεση υπολογισμών (secure multi-party computation) [14] είναι ένα πρόβλημα που προτάθηκε για πρώτη φορά από τον Andrew C. Yao το 1982 [15]. Το παράδειγμα που χρησιμοποίησε για την περιγραφή της ασφαλής εκτέλεσης υπολογισμών είναι το πρόβλημα του εκατομμυριούχου: Η Alice και ο Bob είναι δύο εκατομμυριούχοι που θέλουν να βρουν ποιος είναι πλουσιότερος χωρίς όμως να αποκαλυφθεί το ακριβές ποσό της περιουσίας τους. Ο Yao σε αυτό το πρόβλημα πρότεινε μια λύση που επιτρέπει στην Alice και στον Bob να ικανοποιήσουν την περιέργειά τους ενώ ταυτόχρονα σέβονται τον παραπάνω περιορισμό.

Αυτό το πρόβλημα και τα αποτελέσματα του άνοιξαν τον δρόμο σε μια γενίκευση αποκαλούμενη ως secure multi-party computation (MPC) πρωτόκολλα. Σε έναν MPC, θεωρούμε ένα δεδομένο αριθμό συμμετεχόντων $P_1, P_2, \ldots, P_N$, όπου ο καθένας έχει αντίστοιχα μια ιδιωτική πληροφορία $D_1, D_2, \ldots, D_N$ και έστω ότι θέλουν να υπολογίσουν την τιμή μιας δημόσιας συνάρτησης F για τις N αυτές ιδιωτικές πληροφορίες. Ένα MPC πρωτόκολλο θεωρείτε ασφαλές εάν κανένας συμμετέχων δεν μπορεί να μάθει περισσότερα από την περιγραφή της δημόσιας συνάρτησης και το αποτέλεσμα αυτού του υπολογισμού.

Όπως και πολλά άλλα κρυπτογραφικά πρωτόκολλα, η ασφάλεια ενός MPC πρωτοκόλλου μπορεί να στηριχθεί σε διάφορες υποθέσεις:

- Μπορεί να είναι computational (δηλ. βασισμένο σε κάποιο μαθηματικό πρόβλημα, όπως π.χ. η παραγοντοποίηση) ή unconditional (συνήθως με κάποια πιθανότητα λάθους που μπορεί να είναι αυθαίρετα μικρή).

- Στο μοντέλο που χρησιμοποιούμε, υποθέτουμε ότι οι συμμετέχοντες χρησιμοποιούν ένα συγχρονισμένο δίκτυο (δηλ. ένα μήνυμα που στέλνεται σε κάποιον, φθάνει στον παραλήπτη του στο επόμενο βήμα), ότι υπάρχει ένα ασφαλές και αξιόπιστο broadcast κανάλι, ότι υπάρχει ένα ασφαλές κανάλι επικοινωνίας μεταξύ όλων των συμμετεχόντων (δηλ. ότι ένας αντίπαλος (adversary) δεν μπορεί να διαβάσει, να τροποποιήσει ή να παραγάγει τα μηνύματα στο κανάλι), κλπ.

- Ο κεντρικά ελεγχόμενος αντίπαλος μπορεί να είναι παθητικός (passive) (δηλ. μπορεί να διαβάσει μόνο τα στοιχεία ορισμένων συμμετεχόντων) ή ενεργητικός

(active) (δηλ. μπορεί να αλλοιώσει το πρωτόκολλο εκτέλεσης ή ένα συγκεκριμένο αριθμό συμμετεχόντων).

- Ένας αντίπαλος μπορεί να είναι στατικός (δηλ. επιλέγει τα θύματά του πριν από την έναρξη του υπολογισμού) ή δυναμικός (δηλ. μπορεί να επιλέξει τα θύματά του κατά τη διάρκεια της εκτέλεσης του υπολογισμού). Η επίτευξη της ασφάλειας ενάντια σε ένα δυναμικό αντίπαλο είναι συχνά πολύ πιο δύσκολη από ότι σε ένα στατικό αντίπαλο.

- Ένας αντίπαλος μπορεί να οριστεί ως μια κατώτατου ορίου δομή (που σημαίνει ότι μπορεί να αλλοιώσει ή να διαβάσει απλά τις τιμές των διάφορων συμμετεχόντων μέχρι κάποιο κατώτατο όριο), ή να οριστεί ως μια πιο σύνθετη δομή (μπορεί να έχει επιπτώσεις σε ένα ορισμένο προκαθορισμένο υποσύνολο των συμμετεχόντων). Αυτές οι δομές αναφέρονται συνήθως ως adversary structures. Το υπόλοιπο σύνολο που αποτελείται από τίμια συμβαλλόμενα μέλη που μπορούν ακόμη να εκτελέσουν μια υπολογιστική πράξη, αναφέρονται ως access structures.

Η ασφαλής εκτέλεση υπολογισμών παρέχει λύσεις σε διάφορα πραγματικά προβλήματα όπως η κατανεμημένη ψηφοφορία, η ιδιωτική προσφορά και δημοπρασίες, ο διαμοιρασμός συναρτήσεων υπογραφής ή αποκρυπτογράφησης και η ιδιωτική ανάκτηση πληροφοριών. Η πρώτη μεγάλης κλίμακας και πρακτική εφαρμογή της ασφαλούς εκτέλεσης υπολογισμών πραγματοποιήθηκε στη Δανία τον Ιανουαρίου 2008, όπως περιγράφεται από τον Bogetoft [16], και στόχο είχε να πραγματοποιήσει μια δημοπρασία μεταξύ των αγροτών όπου με την οποία θα καθοριζόταν η τιμή πώλησης των ζαχαρότευτλων στο εργοστάσιο ζάχαρης.

2.11 Polis Project

2.11.1 Περιγραφή του Polis

Το Polis [17] είναι ένα framework για τη διαχείριση προσωπικών δεδομένων που έχει σαν στόχο την προστασία των προσωπικών δεδομένων που χρησιμοποιούνται στις ηλεκτρονικές συναλλαγές. Η βασική αρχή πάνω στην οποία στηρίζεται η λειτουργία του, είναι ότι το κάθε άτομο έχει τον απόλυτο έλεγχο πάνω στα προσωπικά του δεδομένα, τα οποία παραμένουν στην πλευρά του ιδιοκτήτη τους και αποτελούν προσωπική του περιουσία.

Η χρήση του Polis προϋποθέτει ότι τα προσωπικά δεδομένα δεν αποθηκεύονται ποτέ στην πλευρά της εκάστοτε υπηρεσίας για την ολοκλήρωση μιας συναλλαγής. Αντιθέτως, κάθε φορά που η υπηρεσία χρειάζεται κάποια δεδομένα θα πρέπει να τα πληροφορείται εκείνη τη στιγμή που τα χρειάζεται από τον Polis agent του ατόμου/πελάτη. Με αυτό το τρόπο, οι χρήστες αποκτούν τον απόλυτο έλεγχο πάνω στη χρήση των προσωπικών τους δεδομένων.

2.11.2 Απαιτήσεις

Το Polis σχεδιάστηκε έτσι ώστε να προστατεύει την ιδιωτικότητα των χρηστών του διαδικτύου, οι οποίοι θα πρέπει να διαθέτουν τα εξής:

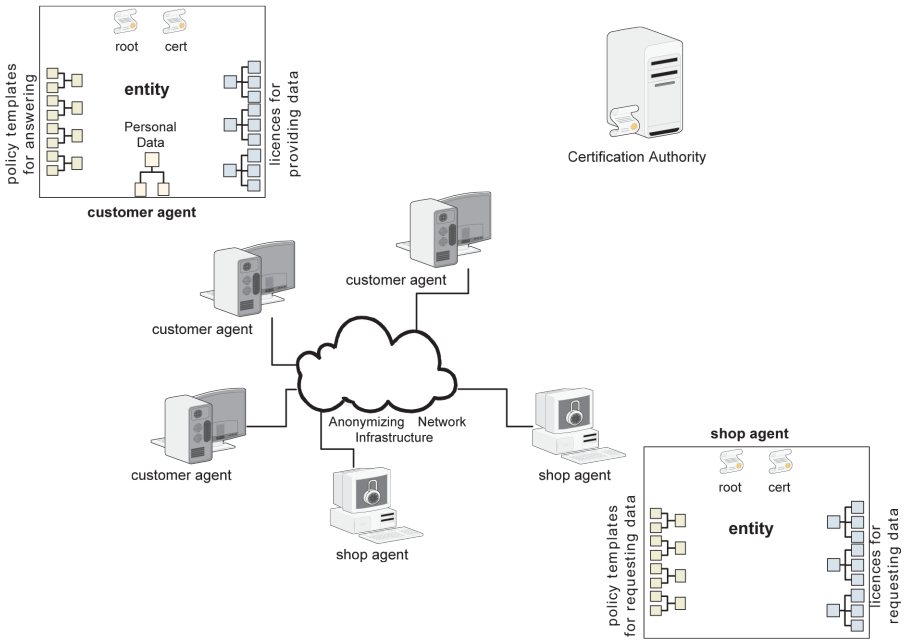
- Έναν αξιόπιστο και συνεχώς προσβάσιμο μέσω του διαδικτύου agent.
- Ένα πιστοποιητικό από μια εγκεκριμένη Αρχή Πιστοποίησης.

2.11.3 Αρχιτεκτονική του Polis

Κάθε χρήστης του Polis αναπαρίσταται ως μια μοναδική οντότητα που αντιπροσωπεύεται από έναν Polis agent. Οι Polis agents αποτελούν το βασικό στοιχείο της αρχιτεκτονικής του Polis, χρησιμοποιούνται για να διαχειρίζονται τα προσωπικά δεδομένα της οντότητας και παρέχουν ελεγχόμενη πρόσβαση σε αυτά. Οι παροχείς υπηρεσιών χρησιμοποιούν τον agent για να πληροφορηθούν για τα προσωπικά δεδομένα των χρηστών τους, που προηγουμένως έχουν συμφωνηθεί με την χρήση αδειών (policies). Στο Σχήμα 2.6 παρουσιάζεται η γενική αρχιτεκτονική του Polis καθώς και τα βασικά μέρη ενός agent πελάτη και ενός agent του καταστήματος.

Η αρχιτεκτονική του Polis διαθέτει τα εξής βασικά χαρακτηριστικά:

- Από την πλευρά του παροχέα υπηρεσιών, το Polis παρέχει μια αποκεντρωμένη προσέγγιση για την αποθήκευση και διαχείριση των προσωπικών δεδομένων.
- Αντιθέτως, από την πλευρά των πελατών, το Polis είναι ένα πλήρως συγκεντρωτικό σύστημα υπό την έννοια ότι τα προσωπικά δεδομένα βρίσκονται και διαχειρίζονται τοπικά από τον agent του πελάτη.



Σχήμα 2.6 Γενική αρχιτεκτονική του Polis.

2.11.4 Δομή προσωπικών δεδομένων και αδειών

Η δομή των προσωπικών δεδομένων του Polis περιέχει οχτώ γενικές κατηγορίες προσωπικών δεδομένων οι οποίες είναι Name, BDate, Cert, Skill, Characteristic, Home-Info, Business-Info και CreditCard. Κάθε μια από αυτές περιέχει μία ή περισσότερες υποκατηγορίες. Η ορολογία που χρησιμοποιείται βασίζεται στο P3P (Platform for Privacy Preferences Project) και στο CPExchange (Global Standards for Privacy-Enabled Customer Data Exchange). Κάθε οντότητα αποθηκεύει τα προσωπικά της δεδομένα σε ένα τοπικό XML αρχείο. Κάθε άδεια (policy) περιέχει τα εξής στοιχεία:

- **Principals:** Οι συμμετέχοντες Polis οντότητες.
- **Data:** Κάθε συγκεκριμένη πληροφορία των προσωπικών δεδομένων του χρήστη.
- **Purposes:** Ένα σύνολο από σκοπούς για τους οποίους επιτρέπεται η χρησιμοποίηση/πληροφόρηση των δεδομένων.
- **Usage Restrictions:** Περιορισμοί που περιορίζουν των αριθμό των προσβάσεων ή το χρονικό διάστημα χρήσης ή και τα δύο.

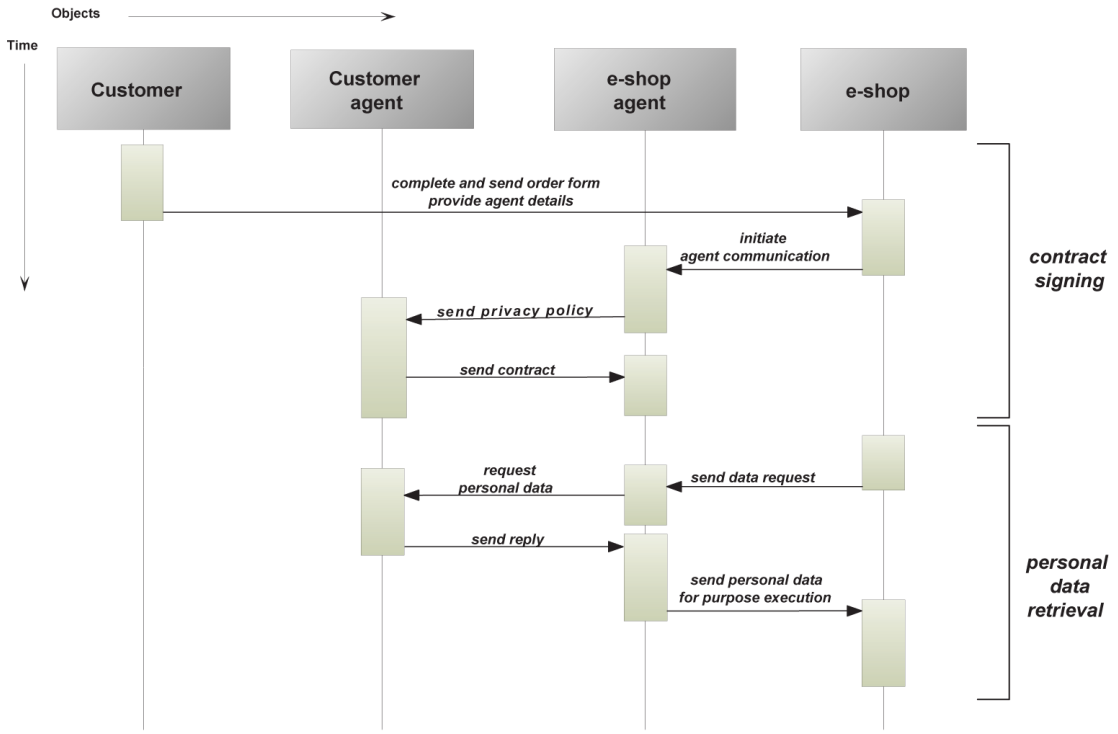
Ανάμεσα στις δύο πλευρές (πελάτη και καταστήματος/υπηρεσίας) συνάπτεται ένα συμβόλαιο το οποίο περιέχει ένα αυθαίρετο σύνολο αδειών. Κάθε agent μπορεί να

υπογράψει έναν οποιοδήποτε αριθμό συμβολαίων με έναν αυθαίρετο αριθμό οντοτήτων.

Στην παρούσα εργασία, οι Polis agents, εκτός από τα παραπάνω στατικά προσωπικά δεδομένα, θα μπορούσαν να χειριστούν και δυναμικά προσωπικά δεδομένα που μπορεί να προκύπτουν από διάφορες φορητές συσκευές, όπως GPS δέκτες. Δεδομένα από τέτοιου είδους τεχνολογίες Ubiquitous Computing θα μπορούσαν να συλλέγονται στον προσωπικό Polis agent των πολιτών και να αξιοποιούνται για την εκτέλεση χρήσιμων υπολογισμών ενισχυμένης ιδιωτικότητας.

2.11.5 Παράδειγμα συναλλαγών με το Polis

Το Polis μπορεί να χρησιμοποιηθεί στις καθημερινές on-line συναλλαγές, όπως για παράδειγμα στις on-line αγορές. Με τη χρήση του μπορούν να πραγματοποιηθούν ηλεκτρονικές αγορές οι οποίες όμως προστατεύουν την ιδιωτικότητα των αγοραστών. Πιο συγκεκριμένα, όταν κάποιος χρήστης χρειάζεται να συμπληρώσει τα προσωπικά του



Σχήμα 2.7 Παράδειγμα συναλλαγής στο Polis με κάποιο ηλεκτρονικό κατάστημα.

δεδομένα σε μια παραγγελία, μπορεί να παρέχει εναλλακτικά τα στοιχεία του προσωπικού του agent. Οι agent του καταστήματος και του πελάτη πραγματοποιούν αρχικά

μια συμφωνία/συμβόλαιο. Εάν η συμφωνία αυτή είναι επιτυχής, ο agent του καταστήματος αποκτά πρόσβαση σε συγκεκριμένα δεδομένα και για συγκεκριμένο χρονικό διάστημα που περιγράφεται στη συμφωνία/συμβόλαιο. Η διαδικασία αυτή απεικονίζεται στο Σχήμα 2.7.

2.11.6 Υποστήριξη εκτέλεσης πρωτοκόλλων

Μια επιπλέον δυνατότητα που προστέθηκε στους Polis agents, εκτός από τη διαχείριση των προσωπικών δεδομένων μιας οντότητας, είναι η εκτέλεση πρωτοκόλλων μεταξύ των agents. Από την στιγμή που οι agents έχουν στη διάθεση τους τα προσωπικά δεδομένα των ιδιόκτητών τους και επιπλέον διαθέτουν μια λίστα με άλλους προσωπικούς agents με τους οποίους συνεργάζονται, μπορούν με τη χρήση κατάλληλων πρωτοκόλλων να πραγματοποιήσουν διάφορους υπολογισμούς που να σέβονται την ιδιωτικότητα των συμμετεχόντων αλλά και όχι μόνο. Δηλαδή, οι Polis agents εκτός από την ανταλλαγή προσωπικών δεδομένων τους δίνεται η δυνατότητα να χρησιμοποιήσουν τα δεδομένα αυτά για την εκτέλεση ασφαλών υπολογισμών με ενισχυμένη ιδιωτικότητα, όπως για παράδειγμα το κλασικό MPC πρωτόκολλο του Yao (πρόβλημα των εκατομμυριούχων).

Κεφάλαιο 3

Συστήματα εύρεσης θέσης (LBS)

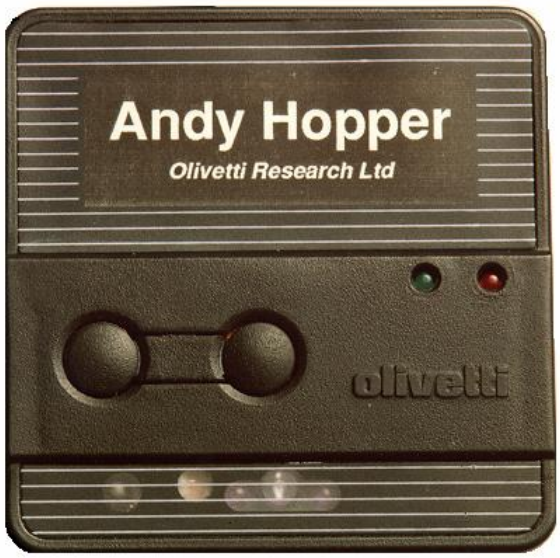
3.1 Εύρεση θέσης σε εσωτερικούς χώρους

3.1.1 Active Badge

Το Active Badge [18] είναι το πρώτο σύστημα που παρουσιάστηκε για την εύρεση θέσης σε εσωτερικούς χώρους (indoor location system). Το συγκεκριμένο σύστημα παρακολουθεί τα διάφορα αντικείμενα/χρήστες που βρίσκονται σε ένα περιβάλλον/χώρο και αποθηκεύει την εκάστοτε θέση τους σε μια κεντρική βάση δεδομένων, με αποτέλεσμα αυτές οι πληροφορίες να είναι ορατές σε όλους μέσα στο κτήριο. Το σύστημα, όπως είχε αρχικά αναπτυχθεί υποθέτει ότι όλοι στο κτίριο είναι αξιόπιστοι και δεν προβλέπει μηχανισμούς για τον περιορισμό διάδοσης της πληροφορίας θέσης των χρηστών με αποτέλεσμα να δημιουργούνται θέματα παραβίασης της ιδιωτικότητας.

Πιο αναλυτικά, για την επίτευξη της εύρεσης θέσης, τα αντικείμενα/χρήστες παρακολουθούνται χρησιμοποιώντας κάποιες ετικέτες (badges) που πρέπει να φέρουν πάντα μαζί τους. Οι ετικέτες (badges) αυτές εκπέμπουν στο υπέρυθρο φάσμα συχνοτήτων ένα μοναδικό κωδικό (ID) που αντιπροσωπεύει το κάτοχο τους, για περίπου ένα δέκατο του δευτερολέπτου και κάθε 15 δευτερόλεπτα (περιοδικά). Η ετικέτα αυτή έχει διαστάσεις περίπου 55 x 55 x 7 χιλιοστά και ζυγίζει μόλις 40 γραμμάρια (Σχήμα 3.1). Αντίστοιχα, υπάρχουν και σταθεροί υπέρυθροι δέκτες που συλλέγουν αυτές τις πληροφορίες και τις μεταδίδουν μέσα από ένα ενσύρματο δίκτυο στη κεντρική βάση δεδομένων. Οι τοίχοι των δωματίων ενεργούν ως φυσικά όρια για τα υπέρυθρα σήματα, επιτρέποντας κατά συνέπεια σε έναν δέκτη να προσδιορίζει τις ετικέτες (badges) που βρίσκονται μέσα σε ένα δωμάτιο. Έτσι λοιπόν, η κάθε ετικέτα (badge) συνδέεται με τη σταθερή θέση του δέκτη που βρίσκεται μέσα σε ένα δωμάτιο και με αυτό τον τρόπο η κεντρική διαχείριση

του συστήματος γνωρίζει που βρίσκεται το κάθε αντικείμενο/χρήστης.



Σχήμα 3.1 Ετικέτα του Active Badge.

Το πρωτότυπο αυτό σύστημα εγκαταστάθηκε για πρώτη φορά το Φεβρουάριο του 1990 και το αποτελούσαν 100 (badges) χρήστες, 200 αισθητήρες και 5 δίκτυα από αισθητήρες σε τέσσερις περιοχές του Cambridge της Αγγλία.

3.1.2 Active Bat

Στο σύστημα Active Bat [19], τα διάφορα αντικείμενα/χρήστες μέσα στο περιβάλλον/χώρο αντιπροσωπεύονται από κάποιες μικρές ασύρματες συσκευές αποστολής σημάτων. Η θέση αυτών των συσκευών αποστολής σημάτων παρακολουθείται από το κεντρικό σύστημα για τη δημιουργία μιας βάση δεδομένων με τις θέσεις των διαφόρων αντικειμένων/χρηστών.

Το σύστημα αυτό αποτελείται από ένα σύνολο φορητών ή σταθερών ασύρματων συσκευών αποστολής σημάτων (transmitters), μια συστοιχία με τους αντίστοιχους δέκτες σημάτων και έναν κεντρικό σταθμό βάσης RF. Η ασύρματη συσκευή αποστολής σημάτων (Σχήμα 3.2) αποτελείται από έναν πομποδέκτη RF, διάφορες συσκευές αποστολής υπέρηχων σημάτων, ένα FPGA, και έναν μικροεπεξεργαστή, και η οποία διαθέτει ένα μοναδικό κωδικό (ID). Οι αντίστοιχοι δέκτες αποτελούνται από έναν δέκτη RF, και μια διεπαφή για ένα σειριακό δίκτυο δεδομένων. Οι δέκτες αυτοί τοποθετούνται στις οροφές του κτηρίου και είναι συνδεδεμένοι σε ένα σειριακό δίκτυο καλωδίων

που δημιουργεί μια συστοιχία δεκτών. Επίσης, αυτό το δίκτυο συνδέεται με έναν υπολογιστή, ο οποίος κάνει όλη την ανάλυση των δεδομένων για την παρακολούθηση των ασύρματων συσκευών αποστολής σημάτων.

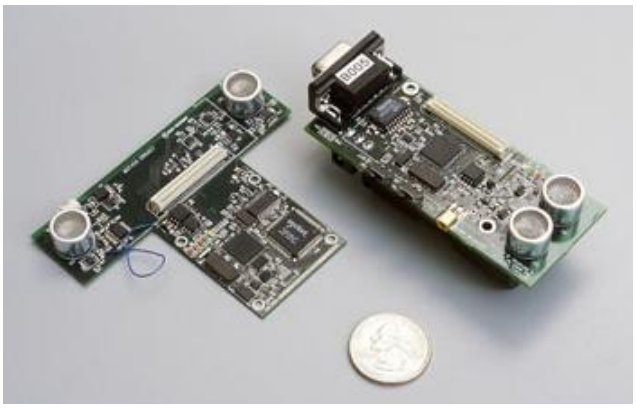
Σχήμα 3.2 Ασύρματη συσκευή αποστολής σημάτων στο Active Bat.

Ο σταθμός βάσης RF διοργανώνει τη δραστηριότητα των συσκευών αποστολής σημάτων με broadcast περιοδικά μηνύματα που απευθύνονται σε έναν κάθε φορά. Μια συσκευή αποστολής σημάτων, όταν δεχτεί ένα μήνυμα που απευθύνεται για αυτή, στέλνει έναν υπέρηχο παλμό. Τα στοιχεία των δεκτών, τα οποία επίσης λαμβάνουν το αρχικό σήμα RF από το σταθμό βάσης, υπολογίζουν το χρονικό διάστημα μεταξύ της λήψης του σήματος RF και της αντίστοιχης λήψης του υπέρηχου σήματος, το οποίο βοηθάει στον υπολογισμό της απόστασης που βρίσκεται η συσκευή αποστολής σημάτων. Αυτές οι αποστάσεις στη συνέχεια αποστέλλονται στον υπολογιστή που εκτελεί την ανάλυση των δεδομένων. Με τη συλλογή αρκετών τέτοιων αποστάσεων από τον υπολογιστή, είναι δυνατό να καθοριστεί η θέση της συσκευής αποστολής σημάτων με ακρίβεια μερικών μάλιστα εκατοστών (μέχρι 3 cm), η οποία με βάση το μοναδικό κωδικό (ID) της συσκευής αποστολής σημάτων αποθηκεύεται στη κεντρική βάση δεδομένων θέσης. Η ακρίβεια όμως που προσφέρει το σύστημα του Active Bat προκύπτει από μια στενά ελεγχόμενη κεντρική (centralized) αρχιτεκτονική που παρακολουθεί και παραβιάζει την ιδιωτικότητα των αντικειμένων/χρηστών σε κάθε κίνηση τους.

Το Active Badge που είδαμε στη προηγούμενη ενότητα εάν και είναι μια σχετικά φθηνή τεχνολογία για τον εντοπισμό της θέσης σε εσωτερικούς χώρους, δεν διαθέτει ακρίβεια στον εντοπισμό της θέσης και πιο συγκεκριμένα μπορούμε μόνο να βρούμε σε ποιο δωμάτιο του κτηρίου βρισκόμαστε. Για το λόγο αυτό, το Active Bat είναι μια πιο ολοκληρωμένη λύση για εφαρμογές που απαιτούν μεγαλύτερη ακρίβεια εντοπισμού της θέσης.

3.1.3 Cricket Location-Support System

Το Cricket [20] είναι ένα σύστημα που επιτρέπει σε εφαρμογές που τρέχουν σε συσκευές χρηστών και σε κόμβους υπηρεσιών να βρίσκουν τη φυσική τους θέση σε εσωτερικούς χώρους. Οι εφαρμογές αυτές των χρηστών έχουν την δυνατότητα, εάν δεν το επιθυμούν, να μην ανακαλύπτουν τη θέση τους στην υπηρεσία ή σε τρίτους. Επιπλέον, παρέχει τη δυνατότητα στο χρήστη να μαθαίνει για τις υπηρεσίες που βρίσκονται στην ακτίνα δράσης του μέσω ενός ενεργού χάρτη (active map) που στέλνεται από μια κεντρική υπηρεσία χαρτών και αλληλεπιδρώντας με τις εκάστοτε υπηρεσίες χρησιμοποιώντας τη τρέχουσα θέση του. Το Cricket διαχωρίζεται από τις υπηρεσίες tracking και αποτελεί μια υπηρεσία πληροφόρησης θέσης, πράγμα το οποίο δίνει τη δυνατότητα ενίσχυσης της ιδιωτικότητας του χρήστη σε αντίθεση με τα συστήματα Active Badge και Active Bat. Στόχος του Cricket είναι ένα σύστημα παροχής/υποστήριξης θέσης, παρά ένα συμβατικό σύστημα παρακολούθησης θέσης που παρακολουθεί και αποθηκεύει τις πληροφορίες θέσης για τις υπηρεσίες και τους χρήστες σε μια κεντρική βάση δεδομένων.



Σχήμα 3.3 Τα αναγνωριστικά (beacons) που χρησιμοποιούνται στο Cricket.

Το Cricket χρησιμοποιεί ένα συνδυασμό RF σημάτων και υπερήχων για την παροχή υπηρεσιών υποστήριξης θέσης σε χρήστες και εφαρμογές. Επιτοίχια και οροφής αναγνωριστικά (beacons) (Σχήμα 3.3) βρίσκονται τοποθετημένα μέσα στους χώρους ενός κτηρίου, δημοσιεύοντας πληροφορίες θέσης με σήματα RF. Με κάθε RF σήμα, το αναγνωριστικό (beacons) μεταδίδει ταυτόχρονα και έναν υπέρηχο παλμό. Οι δέκτες (receivers) λαμβάνουν αυτό το RF και το υπέρηχο σήμα, τα συσχετίζουν μεταξύ τους, εκτιμούν τις αποστάσεις ανάμεσα στα διαφορετικά αναγνωριστικά χρησιμοποιώντας τη διαφορά στο χρόνο διάδοσης του RF και υπέρηχου σήματος, και επομένως

συμπεραίνουν ποιο είναι το διάστημα μέσα στο οποίο βρίσκονται. Τα αναγνωριστικά (beacons) χρησιμοποιούν έναν αποκεντρωμένο τυχαίας μετάδοσης αλγόριθμο για να ελαχιστοποιούνται οι συγκρούσεις και οι παρεμβολές μεταξύ των σημάτων. Κλείνοντας, οι δέκτες εφαρμόζουν έναν αλγόριθμο αποκωδικοποίησης για να αντιμετωπίσουν τις επιδράσεις τέτοιων παρεμβολών μεταξύ των σημάτων.

Στον Πίνακα 3.1 παρουσιάζονται συγκριτικά τα τρία συστήματα (Active Bat, Active Badge και Cricket) εύρεσης θέσης σε εσωτερικούς χώρους.

Σύστημα	**Active Bat**	**Active Badge**	**Cricket**
Ιδιωτικότητα χρήστη	Όχι	Όχι	Ναί
Αποκεντροποιημένο	Όχι	Όχι	Ναί
Ετερογένεια των δικτύων	Ναί	Ναί	Ναί
Κόστος	Υψηλό	Υψηλό	Χαμηλό (10 $)
Ευκολία κατασκευής	Δύσκολο, απαιτεί πίνακα σενσόρων	Δύσκολο, απαιτεί πίνακα σενσόρων	Εύκολο

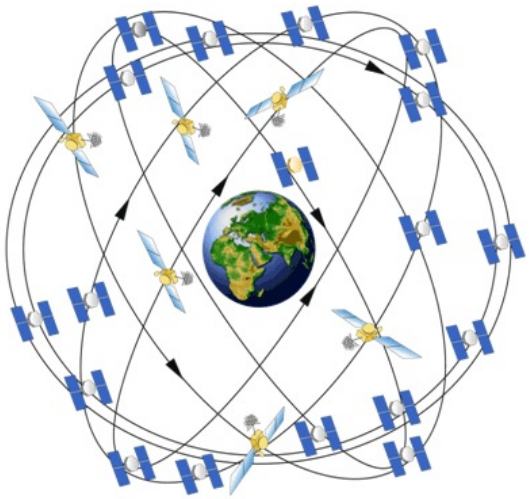
Πίνακας 3.1 Σύγκριση των άλλων συστημάτων εύρεσης θέσης εσωτερικών χώρων με το Cricket [20].

3.2 Εύρεση θέσης σε εξωτερικούς χώρους

3.2.1 Global Positioning System (GPS)

Το GPS (Global Positioning System) [21] είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης, το οποίο βασίζεται σε ένα "πλέγμα" εικοσιτεσσάρων δορυφόρων που περιστρέφονται γύρο από τη Γη (Σχήμα 3.4). Ο καθένας που έχει στη διάθεση του ένα δέκτη GPS μπορεί να βρει ακριβείς πληροφορίες για τη θέση ενός σημείου, το υψόμετρό του, τη ταχύτητα και την κατεύθυνση της κίνησης του. Επίσης, σε συνδυασμό με ειδικό λογισμικό χαρτογράφησης οι πληροφορίες αυτές μπορούν να απεικονιστούν και γραφικά.

Το σύστημα εντοπισμού θέσης GPS σχηματίζει ένα παγκόσμιο δίκτυο, με εμβέλεια που καλύπτει ξηρά, θάλασσα και αέρα. Εξαιτίας αυτής της έκτασής του είναι απαραίτητος ο διαχωρισμός του σε επιμέρους τμήματα όπου πραγματοποιούνται όλες οι λειτουργίες του αλλά και ο συντονισμός του. Αναλυτικά, τα τμήματα αυτά είναι:



Σχήμα 3.4 Απεικόνιση GPS δορυφόρων πάνω από τη Γη.

- **Διαστημικό τμήμα:** Αποτελείται από το δίκτυο 24 δορυφόρων που ήδη αναφέραμε. Οι δορυφόροι αυτοί "σκεπάζουν" ομοιόμορφα με το σήμα τους ολόκληρο τον πλανήτη, γεγονός που αποδεικνύει τη φιλοσοφία που κρύβεται πίσω από τη λειτουργία του συστήματος GPS, δηλαδή τη διαθεσιμότητά του σε κάθε σημείο της Γης, ώστε να μην υπάρχει περίπτωση να αποπροσανατολιστεί κανείς, ποτέ και πουθενά. Όλοι οι δορυφόροι βρίσκονται σε ύψος περίπου 12.700 μιλίων πάνω από την επιφάνεια της θάλασσας και εκτελούν δύο περιστροφές γύρω από τη Γη κάθε 24ωρο. Η κατασκευάστρια εταιρεία είναι η Rockwell International, η εκτόξευσή τους πραγματοποιήθηκε από το ακρωτήριο Canaveral, ενώ η τροφοδοσία τους με ηλεκτρική ενέργεια πραγματοποιείται μέσω των ηλιακών στοιχείων που διαθέτουν.

- **Επίγειο τμήμα ελέγχου:** Οι δορυφόροι, όπως είναι αναμενόμενο, είναι πολύ πιθανό να αντιμετωπίσουν ανά πάσα στιγμή προβλήματα στη σωστή λειτουργία τους. Οι έλεγχοι που πραγματοποιούνται σε αυτούς αφορούν τη σωστή τους ταχύτητα, το υψόμετρο και την κατάσταση της επάρκειάς τους σε ηλεκτρική ενέργεια. Παράλληλα, εφαρμόζονται όλες οι διορθωτικές ενέργειες που αφορούν στο σύστημα χρονομέτρησης των δορυφόρων, ώστε να αποτρέπεται η παροχή λανθασμένων πληροφοριών στους χρήστες του συστήματος. Το τμήμα επίγειου ελέγχου αποτελείται από ένα επανδρωμένο και τέσσερα μη επανδρωμένα κέντρα, εγκατεστημένα σε ισάριθμες περιοχές του πλανήτη.

- **Το τμήμα τελικού χρήστη:** Απαρτίζεται από τους χιλιάδες χρήστες δεκτών GPS ανά την υφήλιο. Οι δέκτες αυτοί μπορούν να χρησιμοποιηθούν τόσο κατά τη διάρκεια μιας απλής πεζοπορίας, όσο και σε οχήματα ή θαλάσσια σκάφη και κατά κανόνα διαθέτουν αρκετά μικρές διαστάσεις. Για να προσφέρουν όσο το δυνατόν περισσότερες πληροφορίες, οι δέκτες συνδυάζονται με ειδικό λογισμικό, που προβάλλει ένα χάρτη στην οθόνη της συσκευής GPS. Πρόκειται, δηλαδή, για λογισμικό που λαμβάνει από τους δορυφόρους τις πληροφορίες για το στίγμα του σημείου στο οποίο βρίσκεται ο δέκτης και τις μετατρέπει σε κατανοητή "ανθρώπινη" μορφή, πληροφορώντας το χρήστη για την ακριβή γεωγραφική του θέση.

Κεφάλαιο 4

Peer-To-Peer (P2P) δίκτυα

4.1 Εισαγωγή

Ενά P2P δίκτυο μπορεί να θεωρηθεί οποιοδήποτε δίκτυο το οποίο λειτουργεί με πρωτόκολλο που εφαρμόζει μια αρχιτεκτονική κατανεμημένου δικτύου το οποίο αποτελείται από συμμετέχοντες κόμβους με συγκεκριμένα χαρακτηριστικά. Αφού το δίκτυο είναι κατανεμημένο, όλοι οι κόμβοι δρουν ως servers και ως clients και είναι όλοι ισάξιοι. Τα δύο αυτά χαρακτηριστικά είναι αλληλένδετα, αφού η διάκριση σε server και client στα συμβατικά δίκτυα αυτομάτως καθιστά κάποιους κόμβους περισσότερο σημαντικούς. Το κύριο χαρακτηριστικό των δικτύων P2P είναι το γεγονός ότι οι κόμβοι παραχωρούν στο δίκτυο ένα μέρος των πόρων τους όπως υπολογιστική ισχύ, αποθηκευτικό χώρο ή εύρος ζώνης (bandwidth) με σκοπό την αύξηση των δυνατοτήτων του κάθε κόμβου χωριστά, αλλά και του δικτύου ως συνόλου, χρησιμοποιώντας τους αναξιοποίητους πόρους των συμμετεχόντων κόμβων [22, 23].

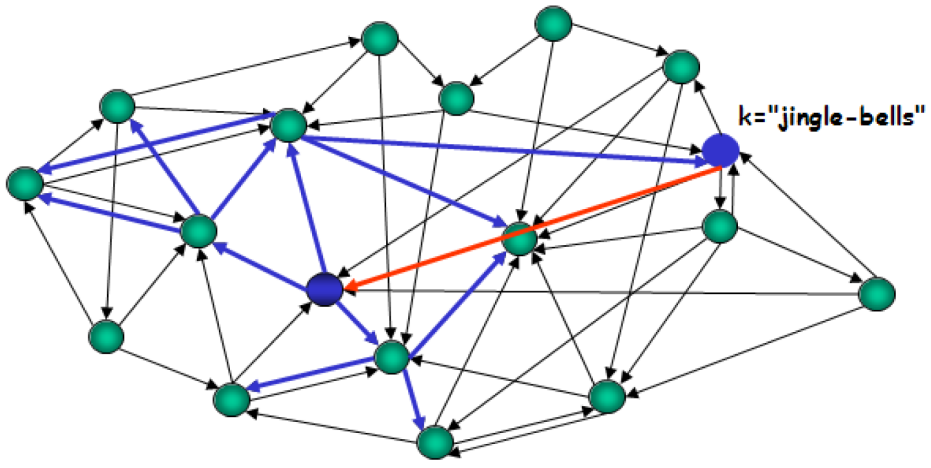
Ένα από τα χαρακτηριστικά των δικτύων P2P που τα καθιστούν τόσο δημοφιλή και θελκτικά είναι το γεγονός ότι πρόκειται για πλήρως κατανεμημένα συστήματα χωρίς την ανάγκη για κεντρική οργάνωση και διαχείριση. Επιπρόσθετα η επεκτασιμότητά τους, καθώς και η απουσία του μοναδικού σημείου αποτυχίας (single point of failure) είναι παράγοντες που παίζουν σημαντικό ρόλο στο ευρύ ενδιαφέρον που υπάρχει γύρω από τα δίκτυα P2P.

4.2 Διαθέσιμες δομές P2P δικτύων

Ένα πρόβλημα που πρέπει να λύσουμε για να επιτύχουμε την διαχείριση των δεδομένων μας σε ένα κατανεμημένο P2P περιβάλλον είναι το εξής: σε ένα δίκτυο P2P στο οποίο συμμετέχουν ισάξιοι κόμβοι, κάθε αντικείμενο d έχει μια διεύθυνση p (ο κόμβος στον οποίο είναι αποθηκευμένο), και κάποιο ή κάποια κλειδιά k_d τα οποία αντιστοιχούν σε αυτό και με τα οποία μπορούν οι κόμβοι να το αναζητήσουν. Επιθυμούμε χρησιμοποιώντας το κλειδί k_d να μπορεί το δίκτυο, χωρίς κεντρικό έλεγχο, να βρίσκει τον κόμβο που κατέχει το d. Για την αντιμετώπιση αυτού του προβλήματος, έχουν προταθεί δομές P2P δικτύων που μπορούμε να τις διακρίνουμε σε: αδόμητα, ιεραρχικά και δομημένα P2P δίκτυα.

4.2.1 Αδόμητα P2P δίκτυα

Ο πιο απλός τρόπος λύσης του προβλήματος που ορίστηκε στην προηγούμενη παράγραφο εφαρμόζεται από τα αδόμητα P2P δίκτυα (Σχήμα 4.1). Δεν χρησιμοποιείται κάποιο είδος δεικτοδότησης, και η πληροφορία (k_d, p) βρίσκεται μόνο στον κόμβο p. Η αναζήτηση σε ένα αδόμητο P2P δίκτυο γίνεται με πλημύρα (flooding). Ο κόμβος που εκκινεί την αναζήτηση στέλνει ένα ερώτημα σε c γείτονες με συγκεκριμένο Time-To-Live (TTL), που ορίζει το βάθος στο οποίο θα επεκταθεί η αναζήτηση. Οι τιμές που θα πάρουν τα c και το TTL εξαρτώνται από την εφαρμογή που χρησιμοποιεί το αδόμητο δίκτυο [24, 25].



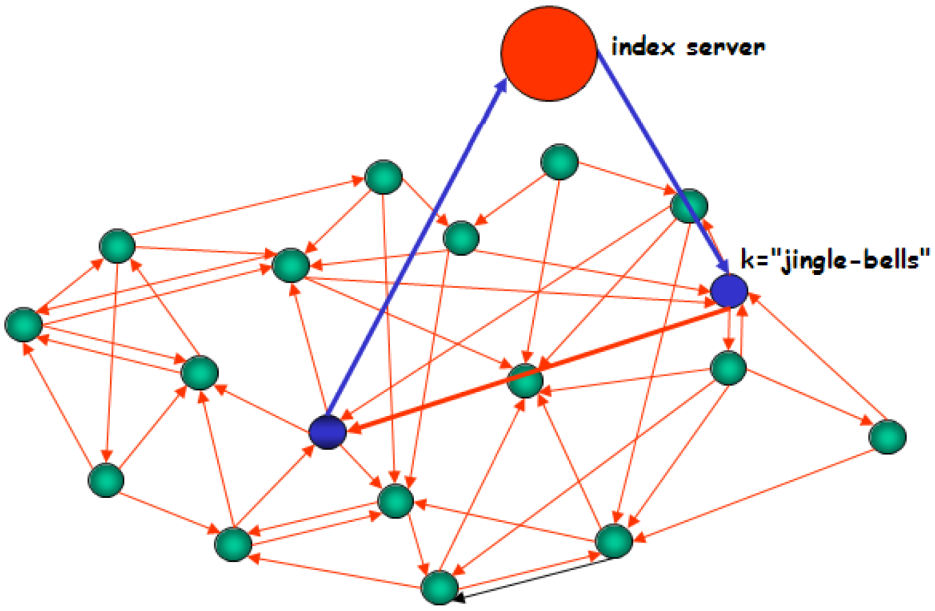
Σχήμα 4.1 Αδόμητο P2P δίκτυο.

Είναι αποδεδειγμένο ότι τα αδόμητα P2P δίκτυα, όπως το Gnutella, είναι ικανά

με σχετικά μικρό c και TTL να προσφέρουν ικανοποιητική ικανότητα αναζήτησης με μικρή καθυστέρηση. Το αρνητικό είναι ότι ένα απλό ερώτημα προκαλεί τεράστιο φόρτο στο δίκτυο, και τον καθιστά ιδιαίτερα αναποτελεσματικό σαν τρόπο αναζήτησης κατανεμημένης πληροφορίας.

4.2.2 Ιεραρχικά P2P δίκτυα

Στα ιεραρχικά P2P δίκτυα ένας ή περισσότεροι κόμβοι (index server, super-peers) κρατάνε αποθηκευμένη την πληροφορία για το ποια δεδομένα βρίσκονται στο δίκτυο και που. Έτσι όταν ένας κόμβος εκκινήσει μια αναζήτηση αυτή αποστέλλεται στον super-peer (Σχήμα 4.2) που είναι αντιστοιχισμένος με αυτόν και μετά από αναζήτηση στους δείκτες που βρίσκονται αποθηκευμένοι σε αυτόν και άλλους πιθανόν υπαρκτούς super-peers, του επιστρέφεται η διεύθυνση του κόμβου που έχει τα δεδομένα που αναζητά.



Σχήμα 4.2 Ιεραρχικό P2P δίκτυο.

Με αυτόν το τρόπο λειτουργούσε και η πρώτη ευρέως διαδεδομένη P2P εφαρμογή Napster. Η χρήση super-peers μας λύνει το πρόβλημα της αποδοτικής αναζήτησης, αλλά δεν εξαλείφει το bottleneck που υπάρχει και στο συμβατικό μοντέλο server-client. Το γεγονός ότι υπάρχει ένα είδος κεντρικής οργάνωσης καθιστά τα ιεραρχικά P2P δίκτυα ουσιαστικά υβριδικά δίκτυα.

4.2.3 Δομημένα P2P δίκτυα

Είναι προφανές ότι η δεικτοδότηση είναι απαραίτητη προϋπόθεση για την επίτευξη γρήγορης και αποδοτικής αναζήτησης σε ένα P2P δίκτυο χωρίς δημιουργία μεγάλου φόρτου σε αυτό. Η συγκέντρωσή τους σε έναν ή λίγους κόμβους δεν είναι σωστή τακτική. Η κατανομή της πληροφορίας των δεικτών όμως σε όλους τους κόμβους δεν είναι κάτι δύσκολο και θα εξάλειφε το bottleneck που εισάγει ο index server. Το μόνο πρόβλημα που έχουμε τώρα να λύσουμε είναι η αποδοτική κατανεμημένη αναζήτηση των δεικτών.

Για να επιτευχθεί η αποδοτική αναζήτηση στην κατανεμημένη πληροφορία που κατέχουν οι δείκτες, αναπτύχθηκε η ιδέα των δομημένων P2P δικτύων. Σε αυτά οι συνδέσεις μεταξύ των κόμβων δεν είναι τυχαίες, όπως ήταν στα προηγούμενα παραδείγματα. Αντίθετα για τη δημιουργία τους εφαρμόζουν ένα πρωτόκολλο για να διασφαλιστεί ότι κάθε κόμβος μπορεί να δρομολογήσει αποδοτικά ένα αίτημα προς έναν άλλο κόμβο που κατέχει τα εν λόγω δεδομένα, όσο σπάνια και να είναι. Ο κύριος τρόπος πρακτικής εφαρμογής επιτυγχάνεται με χρήση DHT (Distributed Hash Table).

Η χρήση DHT μας επιτρέπει να αντιστοιχίζουμε δεδομένα με κόμβους σε κατανεμημένα συστήματα. Χρησιμοποιώντας μια παραλλαγή του συνεχούς (consistent) hashing περιορίζουμε τις αλλαγές που πρέπει να γίνουν στις αντιστοιχίσεις στην περίπτωση που καινούριοι κόμβοι εισέλθουν στο δίκτυο ή υπάρχοντες κόμβοι πάψουν να λειτουργούν, μειώνοντας τον πιθανό φόρτο που θα δημιουργόταν στο δίκτυο σε αντίθετη περίπτωση. Κάποια από τα πιο γνωστά πρωτόκολλα δομημένων δικτύων που χρησιμοποιούν DHT είναι το Chord [26], το Pastry [27], το CAN [28],το Freenet [29] και το Kademlia [30], ενώ το P-Grid [31] χρησιμοποιεί ένα κατανεμημένο δυαδικό δένδρο αναζήτησης.

4.3 Το P2P δίκτυο Chord

Το Chord [26] αποτελεί ένα πρωτόκολλο ομότιμου δικτύου που στοχεύει στην γρήγορη και αποτελεσματική αναζήτηση μεταξύ των κόμβων. Το Chord οργανώνει τους κόμβους του δικτύου γύρω από ένα νοητό δακτύλιο, βάση ενός μοναδικού ID που αντιστοιχεί σε κάθε κόμβο. Η γρήγορη αναζήτηση επιτυγχάνεται με τη χρήση αναφορών σε πολλαπλούς κόμβους σταθερής θέσης (Finger Tables), που αποτελούν σημεία με αποστάσεις της δύναμης του δύο, έτσι ώστε να λογαριθμείται ο χρόνος που χρειάζεται να ταξιδέψει ένα αίτημα μέσα στον δακτύλιο.

Τα βασικά χαρακτηριστικά της δικτυακής αρχιτεκτονικής του Chord είναι η ένταξη των κόμβων σε δακτύλιο, η διατήρηση fingers, η σταθεροποίηση των αναφορών κα-

θώς επίσης και η είσοδος και έξοδος κόμβων από το δίκτυο. Ενδιαφέρον παρουσιάζει ο αλγόριθμος αναζήτησης ενός κλειδιού. Ένας κόμβος που θα λάβει/εκκινήσει ένα αίτημα αναζήτησης θα προωθήσει το αίτημα στη μεγαλύτερη αναφορά του που είναι μικρότερη από το ζητούμενο κλειδί. Αυτά τα χαρακτηριστικά περιγράφονται αναλυτικά στις υπόλοιπες υποενότητες.

4.3.1 Δομή δακτυλίου

Το Chord δομείται στο χαμηλότερο επίπεδο με ένα λογικό δακτύλιο πάνω στον οποίο τοποθετούνται όλοι οι κόμβοι του δικτύου. Η σειρά με την οποία θα τοποθετηθούν καθορίζεται από το κλειδί (ID) των κόμβων. Το ID ενός κόμβου είναι ένα μοναδικό χαρακτηριστικό του και υπολογίζεται με βάση το hash (ένας ακέραιος αριθμός από m-bits) από την IP διεύθυνση και το port που χρησιμοποιεί ο εκάστοτε κόμβος. Έτσι, με διαδικασίες που περιγράφονται πιο κάτω, οι κόμβοι τείνουν να τοποθετηθούν γύρω από το νοητό αυτό δακτύλιο με αύξουσα σειρά modulo n, όπου $n = 2^m$ το μέγιστο πλήθος κόμβων που μπορεί να υποστηρίξει το δίκτυο. Σύμφωνα με αυτή την αρχιτεκτονική, κάθε κόμβος διατηρεί αναφορές στον αμέσως επόμενο του κόμβο στο δακτύλιο (successor, αναφέρεται succ()), καθώς επίσης και για τον αμέσως προηγούμενό του (predecessor, αναφέρεται pred()). Η δικτυακή αυτή οργάνωση αποτελεί μια βασική δομή πάνω στην οποία μπορεί να θεμελιωθεί ένα ομότιμο δίκτυο.

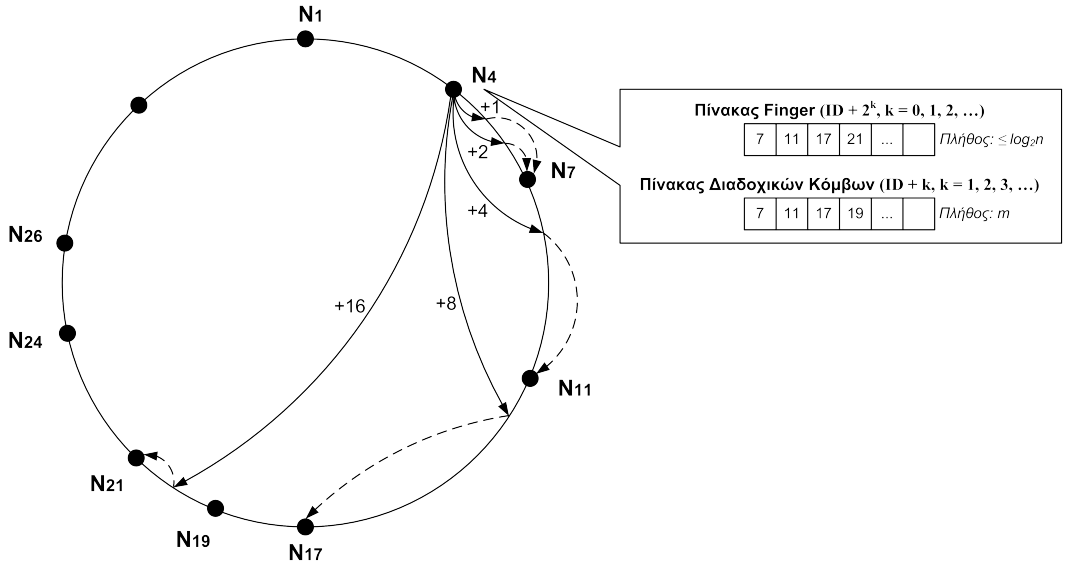
4.3.2 Διαδοχικοί κόμβοι

Πολλές φορές παρατηρούνται φαινόμενα αποχώρησης κόμβων από το δίκτυο, πράγμα το οποίο σημαίνει και διάσπαση του δακτυλίου που αναφέρθηκε πιο πάνω. Για το λόγο αυτό, και προς όφελος του αξιοπιστίας, ο κάθε κόμβος διατηρεί εκτός από τους κοντινότερους του κόμβους, αναφορές και σε ένα ορισμένο αριθμό κόμβων (m) ακριβώς μετά από αυτόν. Με τον τρόπο αυτό, όταν ένας κόμβος αντιληφθεί ότι ο successor του δεν είναι προσβάσιμος, θα ορίσει ως successor τον επόμενο κόμβο που βρίσκεται στις διαδοχικές αναφορές του. Στην παράγραφο 4.3.6 περιγράφεται η διαδικασία δημιουργίας του πίνακα που κρατάει αυτές τις αναφορές.

4.3.3 Fingers

Συχνά ένα ομότιμο δίκτυο αποτελείται από ένα πολύ μεγάλο αριθμό κόμβων. Σε μια τέτοια περίπτωση η δικτυακή αρχιτεκτονική που αναφέρθηκε πιο πάνω, παρότι αξιόπιστη,

δεν είναι επαρκής για την επεκτασιμότητα του δικτύου κάτω από τέτοιες συνθήκες και αυτό διότι ο χρόνος ταξιδιού ενός πακέτου γύρω από το δακτύλιο είναι ανάλογος με τον αριθμό των κόμβων. Τη λύση σε αυτό το πρόβλημα προσφέρει η δικτυακή οργάνωση των fingers του Chord, μέσω της οποίας επιτυγχάνουμε την επιθυμητή επεκτασιμότητα. Σύμφωνα με αυτή, κάθε κόμβος διατηρεί έναν πίνακα ο οποίος περιέχει τις διευθύνσεις των κόμβων με εκθετικά αυξανόμενο ID από τον αρχικό. Πιο συγκεκριμένα, ο κόμβος με κλειδί S θα διατηρήσει αναφορές για τους κόμβους με κλειδιά $S + 2^k$, για $k \geq 1$ και με όριο το μέγιστο αριθμό bits του κλειδιού (π.χ. στην περίπτωση του SHA-1 είναι 160). Αξίζει να σημειωθεί ότι για k=0 η αναφορά ισοδυναμεί με τον successor. Στην παράγραφο 4.3.6 περιγράφεται η διαδικασία δημιουργίας και ενημέρωσης των αναφορών αυτών.



Σχήμα 4.3 Διαδοχικοί κόμβοι και fingers του κόμβου N_4.

4.3.4 Αίτημα αναζήτησης κλειδιού

Η αναζήτηση στο δίκτυο αφορά την εύρεση ενός κόμβου που χαρακτηρίζεται από ένα συγκεκριμένο κλειδί. Εάν δεν υπάρχει κόμβος με το εν λόγω κλειδί, το δίκτυο επιστρέφει τον κόμβο με το αμέσως επόμενο διαθέσιμο κλειδί. Το αίτημα αποτελείται από τα εξής πεδία: τον κόμβο S που ξεκίνησε το αίτημα, το κλειδί K, το οποίο ζητάμε καθώς επίσης και έναν ακέραιο TTL (Time-To-Live) ο οποίος περιορίζει τη διάρκεια ζωής του αιτήματος στο δίκτυο. Για την εύρεση του κλειδιού, το αίτημα διασχίζει το δίκτυο με το τρόπο που περιγράφεται παρακάτω:

Κάθε κόμβος M που λαμβάνει το αίτημα πραγματοποιεί τα παρακάτω βήματα:

- Εάν το TTL είναι μηδέν, αγνοεί πλήρως το αίτημα και τερματίζει, αλλιώς το μειώνει κατά ένα.
- Ο κόμβος M πράττει ανάλογα με τις εξής περιπτώσεις:
  - Εάν το K είναι το κλειδί του κόμβου M, τότε αποστέλλει στον S ένα πακέτο επιβεβαίωσης εύρεσης του κλειδιού.
  - Εάν το K βρίσκεται μεταξύ του pred(M) και M, τότε αποστέλλει και πάλι πακέτο επιβεβαίωσης στον S.
  - Εάν το K βρίσκεται μεταξύ του M και succ(M), προωθεί το αίτημα στον succ(M).
  - Δημιουργεί μια ταξινομημένη κατά ID λίστα που περιέχει όλους τους κόμβους που βρίσκονται τόσο στο Finger Table όσο και στον πίνακα των διαδοχικών κόμβων. Βρίσκει τον κόμβο P της παραπάνω λίστας που έχει το αμέσως μεγαλύτερο ID από το K και προωθεί το αίτημα στον κόμβο P.

Σύμφωνα με τα παραπάνω, το μέγιστο πλήθος αναπηδήσεων που μπορεί να κάνει ένα αίτημα μέσα από το δίκτυο είναι $log_2 n$.

4.3.5 Διαδικασία εισόδου κόμβου στο δίκτυο

Στην παράγραφο αυτή παρουσιάζεται η διαδικασία που ακολουθείται ώστε να εισέλθει ένας κόμβος στο ομότιμο δίκτυο του Chord. Απαραίτητη προϋπόθεση είναι ο προς εισαγωγή κόμβος S να γνωρίζει τα στοιχεία επικοινωνίας (διεύθυνση IP και Port) ενός οποιουδήποτε άλλου κόμβου M που ανήκει στο δίκτυο. Ο S στέλνει στον M ένα πακέτο αίτησης εισχώρησης στο δίκτυο μαζί με κάποια χαρακτηριστικά του κόμβου, όπως η θύρα (port) στην οποία λαμβάνει μηνύματα. Εφόσον ο M λάβει το πακέτο, θα στείλει πίσω ένα πακέτο "δοκιμής" στη θύρα που ακούει ο S, στο οποίο ο S είναι υποχρεωμένος να απαντήσει, επιβεβαιώνοντας έτσι τη δυνατότητά του να λάβει μηνύματα. Ο S θέτει τον M ως successor του, δεδομένου ότι είναι ο μοναδικός κόμβος που γνωρίζει και τον ενημερώνει σχετικά. Εάν ο M δεν έχει predecessor ή εάν το ID του pred(M) είναι μικρότερο από το ID του S, τότε τον δέχεται ως predecessor και η διαδικασία εισαγωγής του S έχει ολοκληρωθεί. Μπορεί να σημειωθεί ότι σε αυτό το σημείο δεν έχει επέλθει πλήρης σταθεροποίηση στο δίκτυο, μιας και κανένας άλλος κόμβος πέραν

του M δε γνωρίζει την ύπαρξη του S. Στην επόμενη ενότητα αναλύεται ο τρόπος με τον οποίο το δίκτυο σταθεροποιείται.

4.3.6 Σταθεροποίηση δικτύου

Ανά τακτά χρονικά διαστήματα εκτελούνται 3 διαφορετικά ήδη σταθεροποίησης (stabilization), τα οποία περιγράφονται στη συνέχεια.

4.3.6.1 Σταθεροποίηση Successor

Η σταθεροποίηση του successor γίνεται με δύο τρόπους:

- *Σταθεροποίηση με έλεγχο predecessor.* Ο S στέλνει ένα πακέτο στον succ(S), ζητώντας να του επιστρέψει τον Predecessor του (pred(succ(S)) = K). Εάν ο K βρίσκεται μεταξύ του S και του succ(S), ο S θα θέσει ως νέο του Successor τον K και θα ενημερώσει τον K σχετικά. Ο K έχει τη δυνατότητα να δεχθεί τον S ως Predecessor του αναλόγως με τη σχετική του θέση ως προς τον τρέχων του Predecessor.

- *Σταθεροποίηση successor με αίτημα.* Ο S εκκινεί ένα αίτημα εύρεσης του κλειδιού με αριθμό $ID(S)+1$ στο δίκτυο (η διαδικασία περιγράφεται σε προηγούμενη παράγραφο). Εάν η απάντηση είναι ο M που θα δεχθεί από το δίκτυο αποτελεί έναν κόμβο πιο κοντά στον succ(S), ο successor θα αντικατασταθεί με τον M. Η μέθοδος αυτή είναι αντικειμενικά πιο γρήγορη από τη μέθοδο με έλεγχο predecessor, αλλά εκτελείται με πιο αργούς ρυθμούς δεδομένης της ενδεχόμενης καταπόνησης του δικτύου.

4.3.6.2 Σταθεροποίηση Διαδοχικών Κόμβων

Μετά την ένταξή του στο δίκτυο, ο S διαθέτει έναν πίνακα διαδοχικών κόμβων, που στο ξεκίνημα της λειτουργίας του περιέχει μόνο τον successor του. Κατά τη διαδικασία της εν λόγω σταθεροποίησης, ο S επιλέγει τυχαία έναν κόμβο M από αυτό τον πίνακα και του αποστέλλει ένα ερώτημα, με το οποίο ζητάει να τον πληροφορήσει για τον succ(M). Στη συνέχεια, ο S θα προσθέσει τον succ(M) στον πίνακα των διαδοχικών κόμβων και θα αφαιρέσει τον κόμβο με τη μεγαλύτερη απόσταση από τον εαυτό του.

4.3.6.3 Σταθεροποίηση Fingers

Η λειτουργία αυτή επιτυγχάνεται αποκλειστικά με ερωτήματα στο δίκτυο. Ο N διατηρεί έναν πίνακα (όπως και στην περίπτωση των διαδοχικών) με τους fingers του. Ο πίνακας αυτός έχει σταθερό μέγεθος και ίσο με τον αριθμό των bits των IDs του δικτύου. Κατά τη σταθεροποίηση αυτή, ο N επιλέγει τυχαία έναν ακέραιο k τέτοιο ώστε $0 \leq k < n$ και ξεκινάει ένα ερώτημα εύρεσης του $S+2^{k+1}$. Ο N προσθέτει τον κόμβο/απάντηση (έστω M) στον πίνακα των fingers εάν στο διάστημα $(S+2^k, S+2^{k+1})$ δεν εμπεριέχεται άλλος κόμβος ή αν αυτός που εμπεριέχεται έχει μεγαλύτερο ID από τον M. Στην τελευταία περίπτωση ο ήδη υπάρχον κόμβος αφαιρείται από τον πίνακα.

Κεφάλαιο 5

Πρόβλημα εύρεσης του κοντινότερου γιατρού

5.1 Εισαγωγή

Το πρόβλημα εύρεσης του κοντινότερου γιατρού (NDP) είναι ένα παράδειγμα εφαρμογής διασφάλισης της ιδιωτικότητας και βασίζεται στα δυναμικά προσωπικά δεδομένα της θέσης του κάθε γιατρού. Έτσι λοιπόν, προτείνεται μια λύση που διασφαλίζει την ιδιωτικότητα και λύνει το πρόβλημα, χωρίς να αποκαλύπτει τη θέση οποιουδήποτε γιατρού. Το άτομο που ανώνυμα προσδιορίζεται ως ο κοντινότερος γιατρός μπορεί να αποκαλύψει (εάν το επιθυμεί) την ταυτότητα του και να προσφέρει τις υπηρεσίες του στο συγκεκριμένο επείγον περιστατικό.

Η λύση που προτείνεται για το NDP, χρησιμοποιεί κρυπτογραφικές μεθόδους και τεχνολογίες κατανεμημένων υπολογισμών. Μια βασική προϋπόθεση είναι ότι όλοι οι γιατροί έχουν στην διάθεση ένας προσωπικό agent για την διαχείριση των προσωπικών τους δεδομένων, όπως η τρέχουσα θέση τους. Επίσης, κάθε agent βρίσκεται κάτω από τον έλεγχο του ιδιοκτήτη του και όλοι οι προσωπικοί agents βρίσκονται μόνιμα συνδεδεμένοι με το διαδίκτυο. Σε περίπτωση έκτακτης ανάγκης, οι agents όλων των γιατρών εκτελούν έναν κατανεμημένο υπολογισμό που μπορεί να προσδιορίσει με έναν κρυπτογραφικά ασφαλή τρόπο ποιος είναι ο κοντινότερος γιατρός στο περιστατικό. Για λόγους απόδοσης, επεκτασιμότητας (scalability), ανεκτικότητας σφαλμάτων (fault tolerance) και επιπλέον ενίσχυσης της ιδιωτικότητας, ο υπολογισμός εκτελείται με έναν πλήρως αποκεντροποιημένο (decentralized) τρόπο και οι agents/κόμβοι είναι οργανωμένοι σε μια κατανεμημένη τοπολογία. Για την επίτευξη αυτού του πλήρως αποκεντροποιημέ-

νου και κατανεμημένου υπολογισμού χρησιμοποιούνται τεχνικές από την περιοχή των Peer-To-Peer (P2P) δικτύων. Η χρήση αυτών των P2P τεχνικών εξασφαλίζει την επεκτασιμότητα του συστήματος και επιπλέον μειώνει τον κίνδυνο παραβίασης της ιδιωτικότητας. Πιο συγκεκριμένα, εφαρμόζονται τεχνικές που έχουν αναπτυχθεί στα πλαίσια του Quantum P2P δικτύου [32] και το οποίο είναι βασισμένο στην αρχιτεκτονική του Chord [26].

5.2 Πιθανές εφαρμογές του NDP

Το πρόβλημα του NDP είναι ένα παράδειγμα μιας εφαρμογής όπου τα προσωπικά δεδομένα μπορούν να χρησιμοποιηθούν για το κοινό καλό (δημόσια υγεία) ενώ συγχρόνως η ιδιωτικότητα των συμμετεχόντων διασφαλίζεται. Μάλιστα, είναι πιθανό να προκύψουν πολλές νέες εφαρμογές που μπορούν να χρησιμοποιήσουν την ιδέα που προτείνεται σε αυτή την εργασία, όπως για παράδειγμα:

- **Πρώτες βοήθειες σε περίπτωση εκτάκτου ανάγκης αυτοκινητιστικού ατυχήματος.** Η Ευρωπαϊκή Ένωση έχει προωθήσει το πρόγραμμα eCall [33] για τη διερεύνηση της δυνατότητας παροχής βοήθειας σε περιπτώσεις εκτάκτου ανάγκης σε ένα αυτοκινητιστικό ατύχημα. Στόχος αυτού του προγράμματος είναι η κατασκευή ενός μαύρου κουτιού που θα εγκαθίσταται στα οχήματα και θα έχει τη δυνατότητα αποστολής ενός αιτήματος εκτάκτου ανάγκης σε περιπτώσεις αυτοκινητιστικών ατυχημάτων. Το αίτημα θα διαβιβάζεται ασύρματα μέσω ενός GSM τηλεπικοινωνιακού δικτύου και θα περιλαμβάνει πληροφορίες όπως GPS συντεταγμένες της θέση του ατυχήματος, εάν άνοιξε ο αερόσακος ή όχι και πιθανές άλλες πληροφορίες από τους αισθητήρες του οχήματος. Μια επιπλέον λειτουργία που θα μπορούσε να προστεθεί σε αυτό το σύστημα είναι και η αναζήτηση εξειδικευμένων πρώτων βοηθειών από άτομα (π.χ. γιατρών, νοσηλευτών και νοσοκόμων) που βρίσκονται στα κοντινά αυτοκίνητα. Ωστόσο, η θέση ενός οχήματος είναι ιδιωτική πληροφορία και έτσι η αναζήτηση των κοντινών αυτοκινήτων πρέπει να γίνει με τρόπου που να διασφαλίζεται η ιδιωτικότητα. Μια πιθανή λύση που θα μπορούσε να προταθεί είναι αυτή του NDP για την ειδοποίηση του κατάλληλου ατόμου που πιθανόν να βρίσκεται στα κοντινά αυτοκίνητα. Μια άλλη διαφορετική εφαρμογή του NDP σε αυτό το πρόβλημα θα ήταν να ειδοποιεί τα επερχόμενα οχήματα να επιβραδύνουν έτσι ώστε να αποφευχθεί ένα πιθανόν νέο αυτοκινητιστικό ατύχημα στην περιοχή.

- **Έκτακτη ανάγκη για αστυνομική ή πυροσβεστική βοήθεια.** Σε περίπτωση έκτακτης ανάγκης για την παροχή αστυνομικών ή πυροσβεστικών υπηρεσιών, ο αστυνομικός ή ο πυροσβέστης που δεν είναι σε υπηρεσία και τυγχάνει να βρίσκεται κοντά σε ένα περιστατικό, θα μπορούσε να είναι σε θέση να παρέχει σημαντικές υπηρεσίες εάν ενημερωνόταν για την έκτακτη ανάγκη. Συγχρόνως, δεδομένου ότι το άτομο (αστυνομικός ή πυροσβέστης) δεν είναι σε υπηρεσία, η ακριβής του θέση είναι ευαίσθητο προσωπικό δεδομένο και κανένας δεν έχει το δικαίωμα να τη ξέρει. Μια λύση, όπως αυτή του NDP, θα μπορούσε να βρει ένα τέτοιο κατάλληλο άτομο (βέβαια με τη συγκατάθεσή του). Το άτομο θα καλούταν να παρέχει τη βοήθεια του μόνο εάν ήταν το κοντινότερο άτομο και μόνο εάν ήταν αρκετά κοντά ώστε να μπορέσει να σπεύσει σχετικά γρήγορα.

5.3 Σχετικές εργασίες

Το Active Badge (Ενότητα 3.1.1) ήταν το πρώτο σύστημα εύρεσης θέσης σε εσωτερικούς χώρους για άτομα που βρίσκονται στα γραφεία μιας εταιρίας. Το σύστημα όμως αυτό αύξησε τα ζητήματα για την προστασία της ιδιωτικότητας της θέσης των ατόμων στον εργασιακό τους χώρο. Επέκταση αυτού του συστήματος αποτελεί το Active Bat (Ενότητα 3.1.2) που προσφέρει αυξημένες δυνατότητες στους χρήστες για τον έλεγχο του τρόπου με τον οποίο κάποιος έχει τη δυνατότητα να βρει που βρίσκεται (θέση) στους χώρους μιας εταιρίας. Ωστόσο, και τα δύο αυτά συστήματα υποθέτουν ότι η κεντρική υπηρεσία (trust server) διαχείρισης των δεδομένων θέσης είναι έμπιστη. Ένα σύστημα το οποίο διαθέτει έναν αποκεντροποιημένο έλεγχο των προσωπικών δεδομένων θέσης είναι το σύστημα Cricket (Ενότητα 3.1.3). Το Cricket είναι ένα σύστημα που προσφέρει στο άτομο τη δυνατότητα να μάθει τη φυσική του θέση μέσα σε ένα κτήριο, χωρίς όμως να παρακολουθείται το ίδιο από μια κεντρική υπηρεσία (που κάνουν τα προηγούμενα δυο συστήματα). Λόγο αυτής της δυνατότητας, το άτομο μπορεί στη συνέχεια να αποφασίσει σε ποιον θα αποκαλύψει τη θέση του. Αυτή η προσέγγιση, προσφέρει έναν καλύτερο έλεγχο για το ποιος μαθαίνει τις πληροφορίες θέσης του ατόμου. Ωστόσο, εάν ο χρήστης θέλει να χρησιμοποιήσει ενεργά τις υπηρεσίες/λειτουργίες που παρέχει αυτό το σύστημα μέσα στο κτήριο θα πρέπει να αποκαλύψει τελικά τη θέση του. Η προσέγγιση του Cricket θα μπορούσε να χρησιμοποιηθεί στο NDP για την εύρεση της θέσης των ατόμων μέσα σε κτήρια και εκεί γενικά όπου δεν θα μπορούσε να χρησιμοποιηθεί το GPS (Ενότητα 3.2.1).

Τα θέματα ιδιωτικότητας για εφαρμογές όπως το NDP είναι περισσότερο κρίσιμα

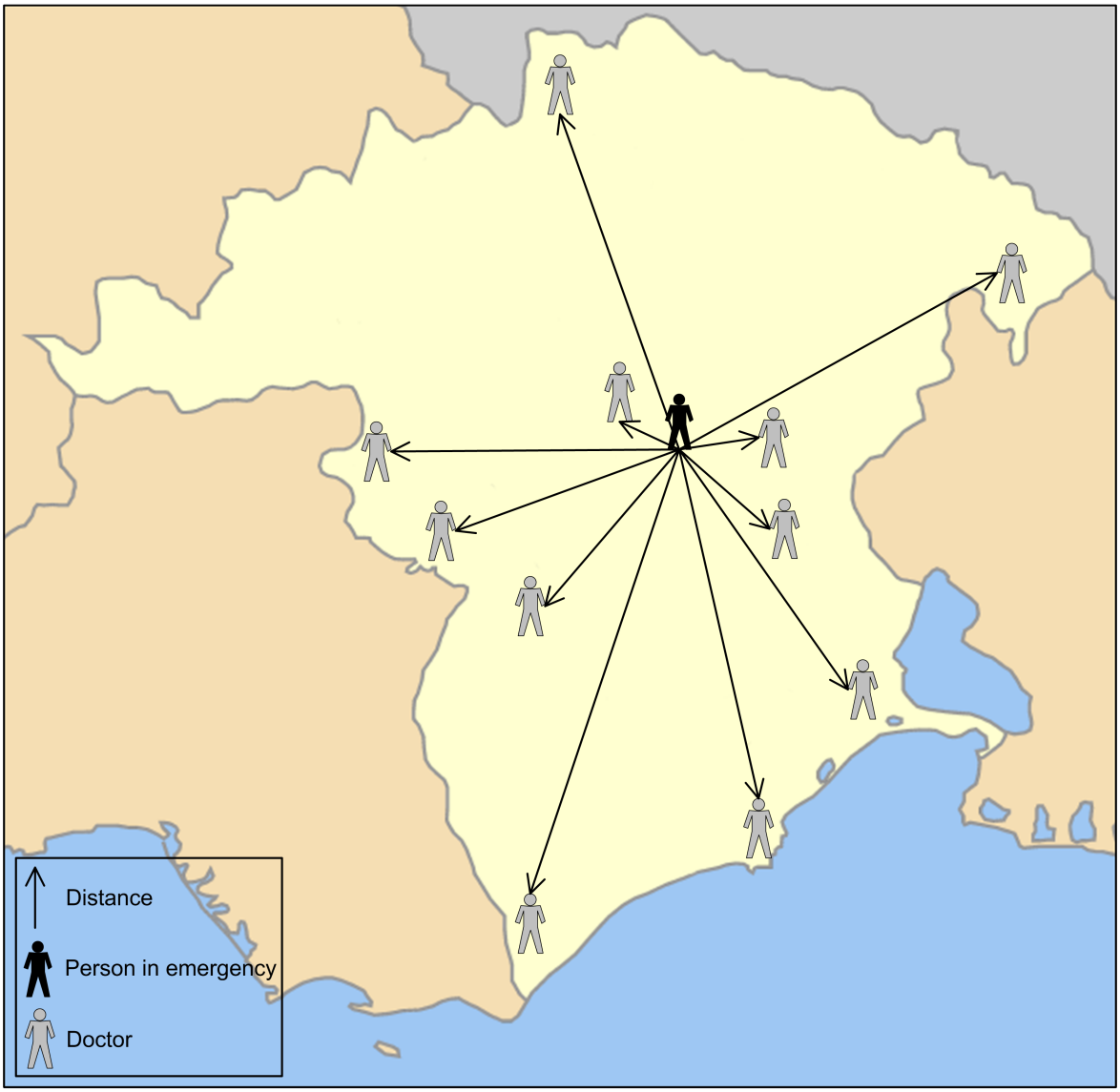
δεδομένου ότι ισχύουν για άτομα που μπορεί να είναι στον ιδιωτικό τους χρόνο και όχι μόνο στο γραφείο τους αλλά σε οποιαδήποτε θέση. Ένας τέτοιος υπολογισμός που απαιτεί την είσοδο τιμών από δύο ή περισσότερους συμμετέχοντες και υπολογίζει το αποτέλεσμα χωρίς να αποκαλύπτει τη τιμή εισόδου σε οποιονδήποτε συμμετέχοντα είναι μια ασφαλής εκτέλεση υπολογισμών (MPC = Secure multi-party computation). Ένα γενικό μοντέλο για την ασφαλή εκτέλεση υπολογισμών προτάθηκε για πρώτη φορά από τον Yao [15]. Ωστόσο, το γενικό αυτό μοντέλο είναι σχεδόν αδύνατο να εφαρμοσθεί σε πρακτικές εφαρμογές. Αποδοτικότερες προσεγγίσεις έχουν αναπτυχθεί για συγκεκριμένες εφαρμογές, όπως παράδειγμα στις εργασίες [34, 35]. Η λύση του NDP που παρουσιάζεται σε αυτή την εργασία είναι μια αποδοτικά ασφαλή εκτέλεση υπολογισμών (MPC) για το πρόβλημα του NDP.

5.4 Ορισμός του πρόβληματος εύρεσης του κοντινότερου γιατρού (NDP)

Σε αυτή την ενότητα γίνεται μια αναλυτική περιγραφή του προβλήματος εύρεσης του κοντινότερου γιατρού (NDP) σε ένα επείγον περιστατικό (στο Σχήμα 5.1 παρουσιάζεται παραστατικά που μπορεί να βρίσκονται οι γιατροί και που το επείγον περιστατικό). Ο κύριος στόχος του NDP είναι να βρεθεί ο κοντινότερος γιατρός χωρίς όμως να παραβιάζεται η ιδιωτικότητα των γιατρών. Για την πραγματοποίηση αυτού του υπολογισμού, τα μόνα προσωπικά δεδομένα που απαιτούνται είναι οι ακριβείς θέσεις όλων των γιατρών.

Ο ορισμός του προβλήματος της εύρεσης του κοντινότερου γιατρού (NDP) είναι ο εξής:

- Υπάρχουν **N** γιατροί $(D_1, D_2, \ldots, D_N)$.
- Για $i = 1, 2, \ldots, N$, ορίζουμε ως L_i να είναι η τρέχουσα θέση του γιατρού D_i. Για παράδειγμα, η θέση L_i μπορεί να είναι η ακριβής GPS γεωγραφική θέση του γιατρού, η οποία μπορεί να υπολογίζεται με τη βοήθεια ενός φορητού GPS δέκτη.
- **Η NDP συνάρτηση υπολογισμού:** Στην περίπτωση ενός επείγοντος περιστατικού, οι agents όλων των γιατρών πραγματοποιούν έναν κατανεμημένο υπολογισμό που διασφαλίζει όμως και την ιδιωτικότητα τους. Η είσοδος και η έξοδος αυτής της συνάρτησης υπολογισμού είναι:

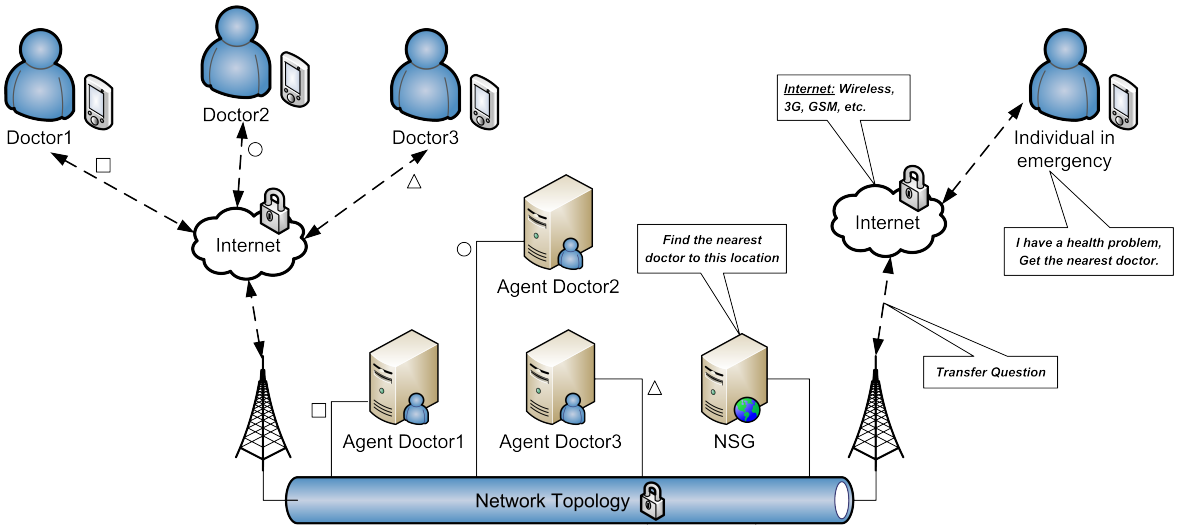


Σχήμα 5.1 Αναπαράσταση για του που μπορεί να βρίσκονται οι γιατροί και το επείγον περιστατικό στο Νομό Ξάνθης.

- **Είσοδος:** Η θέση L_{em} του επείγοντος περιστατικού.
- **Έξοδος:** Στο τέλος του υπολογισμού, ο γιατρός που είναι κοντινότερος στη θέση του επείγοντος περιστατικού γίνεται ενήμερος για αυτό το γεγονός και μπορεί, εάν το επιθυμεί και ο ίδιος, να προσφέρει τις υπηρεσίες του.

5.5 Η λύση του NDP

Σε αυτή την ενότητα περιγράφεται μια κατανεμημένη λύση με ενισχυμένη ιδιωτικότητα για την επίλυση του NDP προβλήματος. Μια γενική επισκόπηση της αρχιτεκτονικής της NDP λύσης παρουσιάζεται στο Σχήμα 5.2. Η λύση η οποία προτείνεται στηρίζεται σε ένα ασφαλές κρυπτογραφικό πρωτόκολλο για την πραγματοποίηση κατανεμημένων υπολογισμών. Επιπρόσθετα, δείχνεται ότι το πρωτόκολλο αυτό είναι ασφαλές σύμφωνα με το μοντέλο ασφάλειας του "Honest-But-Curious" (HBC) χρήστη, δηλαδή ότι οι γιατροί ακολουθούν πιστά τα βήματα του πρωτοκόλλου, αλλά μπορεί να προσπαθήσουν να εξαγάγουν πρόσθετες πληροφορίες κατά την εκτέλεση (Ενότητα 5.6, Ορισμός 1). Το μοντέλο HBC χρησιμοποιείται συνήθως σε κρυπτογραφικά πρωτόκολλα και ταιριάζει απόλυτα για το πρόβλημα του NDP, δεδομένου ότι οι συμμετέχοντες είναι πιστοποιημένοι γιατροί.



Σχήμα 5.2 Αρχιτεκτονική της λύσης του NDP.

Οι υποθέσεις που έχουν γίνει για την επίλυση του NDP προβλήματος είναι οι εξής:

- Κάθε γιατρός έχει έναν προσωπικό agent για τη διαχείριση των προσωπικών του δεδομένων και βρίσκεται πάντα διαθέσιμος στο διαδίκτυο.
- Η τρέχουσα θέση κάθε γιατρού αποθηκεύεται στον προσωπικό του agent.

5.5.1 Διαδικασία υπολογισμού του NDP

Τα βήματα που πρέπει να ακολουθηθούν για την επίλυση του NDP προβλήματος σε περιπτώσεις εκτάκτου ανάγκης είναι τα εξής:

- Το άτομο που βρίσκεται στην έκτακτη ανάγκη, υποβάλλει ένα αίτημα στην υπηρεσία NDP Service Gateway (NSG). Το αίτημα περιέχει τη τρέχουσα θέση του ατόμου (π.χ., ακριβές γεωγραφικό μήκος και πλάτος) και ενδεχομένως πρόσθετες πληροφορίες όπως η ταυτότητά του, η κατάσταση του, κλπ.

- Η υπηρεσία NSG είναι αυτή που δέχεται το αίτημα και είναι υπεύθυνη να το μεταβιβάσει σε κάποιον agent της κοινότητας των γιατρών. Ο agent που θα παραλάβει το αίτημα από την υπηρεσία NSG, παίζει το ρόλο του root-κόμβου για το συγκεκριμένο υπολογισμό.

- Ο root-κόμβος συντονίζει τον κατανεμημένο υπολογισμό, όπου με τον οποίο υπολογίζεται η απόσταση του κοντινότερου γιατρού.

- Στο τέλος του κατανεμημένου υπολογισμού, ο agent του γιατρού που είναι κοντινότερος στη θέση του επείγοντος περιστατικού ενημερώνεται για αυτό το γεγονός και έρχεται σε επαφή με την υπηρεσία NSG για να δηλώσει την ετοιμότητά του να προσφέρει βοήθεια.

5.5.2 Περιγραφή του κατανεμημένου υπολογισμού

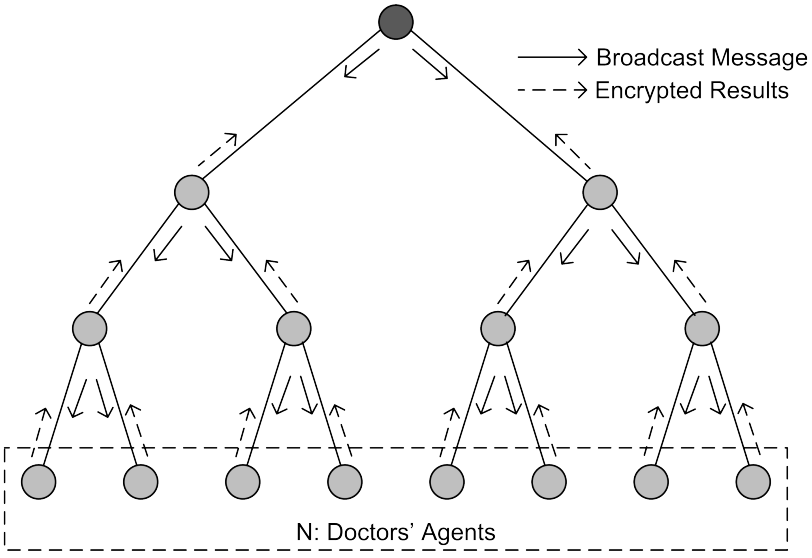
Σε αυτή την ενότητα παρουσιάζεται το πρωτόκολλο ενός ασφαλούς κατανεμημένου υπολογισμού που επιλύει το πρόβλημα NDP. Το πρωτόκολλο αυτό δεν αποκαλύπτει την ακριβή θέση οποιουδήποτε γιατρού, παρά μόνο αποκαλύπτεται ανώνυμα στην υπηρεσία NSG ένας μικρός αριθμός από αποστάσεις, που πιθανόν υπάρχουν γιατροί, κατά την διάρκεια του υπολογισμού. Η διαδικασία του υπολογισμού αποτελείται από τρεις κύριες φάσεις. Στη **Φάση 1**, υπολογίζεται το πιο κοντινό διάστημα που περιέχει τουλάχιστον έναν γιατρό. Στη **Φάση 2**, υπολογίζεται η ακριβή απόσταση του κοντινότερου γιατρού μαζί με ένα τυχαίο ID που αντιστοιχεί στον ανώνυμο γιατρό. Τέλος, στη **Φάση 3**, ο γιατρός που είναι κάτοχος αυτού του τυχαίου ID ανακαλύπτει ότι είναι ο κοντινότερος γιατρός και έρχεται σε επαφή με την υπηρεσία NSG για να προσφέρει τη βοήθειά του.

- **Φάση 1**

  - **Είσοδος:** Η θέση L_{em} του επείγοντος περιστατικού.

– **Έξοδος:** Ένα διάστημα I, με τις ελάχιστες αποστάσεις των γιατρών, που περιέχει τουλάχιστον ένα γιατρό και το πολύ K γιατρούς, όπου K είναι μια δεδομένη σταθερά (π.χ. $K = 5$).

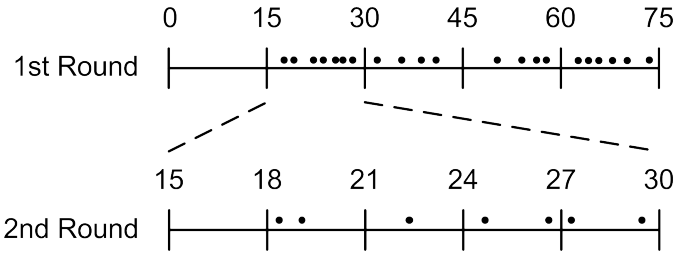
– **Περιγραφή:** Η υπηρεσία NSG επιλέγει έναν κόμβο (τυχαία) ως root-κόμβο για το συγκεκριμένο υπολογισμό και του στέλνει τη θέση L_{em} του επείγοντος περιστατικού. Ο root-κόμβος στέλνει ένα broadcast μήνυμα όπου με το οποίο αρχίζει το κατανεμημένο πρωτόκολλο και ξεκινάει τη Φάση 1. Αυτό το πρωτόκολλο εκτελείται σε μια λογική δυαδική δεντρική τοπολογία που έχει ως φύλλα της όλους τους agents (κόμβους) των γιατρών (Σχήμα 5.3). Η Φάση 1 μπορεί να επαναληφθεί αρκετές φορές μέχρι να βρεθεί το κοντινότερο διάστημα που περιέχει τουλάχιστον ένα γιατρό. Σε κάθε γύρο, ο root-κόμβος συλλέγει τα (ενδιάμεσα) αποτελέσματα του υπο-



Σχήμα 5.3 Δυαδική δεντρική τοπολογία.

λογισμού ως ένα κρυπτογραφημένο μήνυμα και αποστέλλει το μήνυμα αυτό στην υπηρεσία NSG. Αυτό το μήνυμα είναι κρυπτογραφημένο με το δημόσιο κλειδί της υπηρεσίας NSG για τον συγκεκριμένο υπολογισμό και το οποίο γίνεται γνωστό σε όλους τους κόμβους της κοινότητας των γιατρών. Υποθέτουμε ότι το $Count_D$ είναι ο αριθμός των γιατρών που βρίσκονται μέσα στο κοντινότερο διάστημα των αποστάσεων. Η τιμή που έχει το $Count_D$ υπολογίζεται από την υπηρεσία NSG με την αποκρυπτογράφηση του μηνύματος. Εάν το $Count_D > K$, τότε η διαδικασία του υπολογισμού επαναλαμβάνεται για το διάστημα που προέκυψε από τον προηγούμενο γύρο με μεγαλύτερη

όμως διακριτότητα. Αυτή η διαδικασία συνεχίζεται μέχρι για το κοντινότερο διάστημα που θα προκύψει να ισχύει $Count_D < K$, δηλαδή μέσα σε αυτό το διάστημα να υπάρχουν λιγότεροι από K γιατροί. Ένα παράδειγμα αυτής της διαδικασίας, όπου το κοντινότερο διάστημα βρέθηκε μέσα σε δύο γύρους, παρουσιάζεται στο Σχήμα 5.4. Στην επόμενη ενότητα 5.5.3 περιγράφεται αναλυτικά το πρωτόκολλο της Φάσης 1, με το οποίο εξασφαλίζεται k-anonymity (Ενότητα 5.6, Ορισμός 2), για $k = N$, όπου N είναι το πλήθος όλων των συμμετεχόντων κόμβων του δικτύου της κοινότητας των γιατρών.



Σχήμα 5.4 Παράδειγμα εκτέλσης της Φάσης 1.

- **Φάση 2**
  - **Είσοδος:** Το διάστημα I από τη Φάση 1.
  - **Έξοδος:** Οι ακριβείς αποστάσεις των $Count_D$ κοντινότερων γιατρών, με τα αντίστοιχα τυχαία ID έχουν συλλεγεί ανώνυμα.
  - **Περιγραφή:** Σε αυτή τη φάση, η υπηρεσία NSG στέλνει το διάστημα I της Φάσης 1 στον root-κόμβο, ο οποίος με την σειρά του αποστέλλει με broadcast μήνυμα το διάστημα I ανακοινώνοντας το στους agents. Κάθε agent του οποίου η απόσταση είναι μέσα στο διάστημα I αποκρίνεται στέλνοντας ανώνυμα ένα μήνυμα στην υπηρεσία NSG. Το μήνυμα αυτό είναι κρυπτογραφημένο με το δημόσιο κλειδί της NSG και περιέχει την ακριβή απόσταση του agent/γιατρού μαζί με ένα τυχαίο ID (που είναι ένας μοναδικός αριθμός για τον συγκεκριμένο υπολογισμό). Αυτή η ανώνυμη επικοινωνία επιτυγχάνεται με τεχνικές Onion Routing [36]. Περισσότερες πληροφορίες σχετικά με το Onion Routing δίνονται στην ενότητα 5.5.4. Η υπηρεσία NSG συλλέγει όλα αυτά τα ανώνυμα μηνύματα και βρίσκει την απόσταση του κοντινότερου γιατρού με το αντίστοιχο τυχαίο ID. Δεδομένου ότι, τα μηνύματα στέλνονται ανώνυμα, επιτυγχάνεται η προστασία της ιδιωτικότητας των γιατρών.

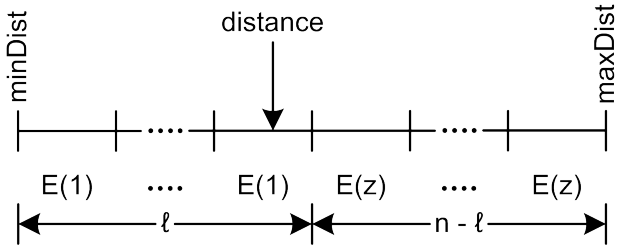
- **Φάση 3**
  - **Είσοδος:** Το τυχαίο ID που αντιπροσωπεύει την απόσταση του κοντινότερου γιατρού.
  - **Έξοδος:** Ο ιδιοκτήτης του τυχαίου ID αποκαλύπτει ότι είναι ο κοντινότερος γιατρός $D_{nearest}$ και έρχεται σε επαφή με την υπηρεσία NSG.
  - **Περιγραφή:** Η υπηρεσία NSG στέλνει ένα μήνυμα στο root-κόμβο που περιέχει το τυχαίο ID της απόστασης του κοντινότερου γιατρού. Ο root-κόμβος με broadcast μήνυμα αποστέλλει το τυχαίο ID στο δίκτυο των agents. Ο agent του γιατρού που παρήγαγε το τυχαίο ID ενημερώνεται για το γεγονός ότι είναι ο κοντινότερος γιατρός και αποκαλύπτει την ταυτότητά του στην υπηρεσία NSG.

5.5.3 Το πρωτόκολλο ενισχυμένης ιδιωτικότητας της Φάσης 1

Το κρυπτογραφικό πρωτόκολλο ενισχυμένης ιδιωτικότητας που παρουσιάστηκε στην προηγούμενη ενότητα 5.5.2 στη Φάση 1 βρίσκει το μικρότερο διάστημα απόστασης στο οποίο υπάρχει τουλάχιστον ένας γιατρός. Η βασική ιδέα αυτού του πρωτοκόλλου είναι βασισμένη σε ένα σχετικό πρωτόκολλο του Yokoo [34] για ασφαλή δυναμικό προγραμματισμό. Το κρυπτογραφικό αυτό πρωτόκολλο χρησιμοποιεί το κρυπτογραφικό σύστημα δημοσίου κλειδιού ElGamal (Ενότητα 2.6) και εκμεταλλεύεται την αντίστοιχη ομομορφική του ιδιότητα (Ενότητα 2.7).

Πιο αναλυτικά, το πρωτόκολλο δέχεται τρεις παραμέτρους: Την ελάχιστη απόσταση $minDist$, τη μέγιστη απόσταση $maxDist$ και τον αριθμό n των υποδιαστημάτων. Οπότε δηλαδή το διάστημα $(minDist, maxDist)$ είναι τεμαχισμένο σε n υποδιαστήματα. Για λόγους απλότητας χρησιμοποιούμε υποδιαστήματα του ίδιου μεγέθους, αλλά είναι σχετικά εύκολο να τεμαχιστεί αυτό το διάστημα σε γεωμετρικά αυξανόμενα υποδιαστήματα. Το αποτέλεσμα αυτού του πρωτοκόλλου είναι η εύρεση του μικρότερου υποδιαστήματος που περιέχει (την απόσταση) τουλάχιστον ένα γιατρό. Για να επιτευχτεί αυτό, κάθε υποδιάστημα αντιπροσωπεύεται με ένα ciphertext και ολόκληρο το διάστημα αντιπροσωπεύεται με μια ταξινομημένη λίστα όλων των ciphertexts. Συνολικά δηλαδή, κάθε μήνυμα έχει n κρυπτογραφημένους αριθμούς, τόσοι όσα και τα υποδιαστήματα στα οποία χωρίζεται το αρχικό διάστημα. Ένα τέτοιο μήνυμα με τη ταξινομημένη λίστα των ciphertexts δημιουργείται αρχικά από όλους τους agents των γιατρών. Ο κάθε agent προετοιμάζει τη ταξινομημένη λίστα των ciphertexts του ως

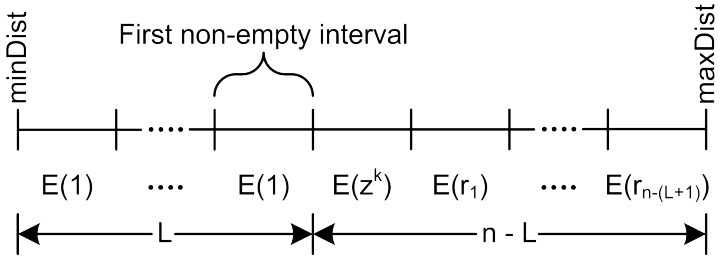
εξής: Για το γιατρό D_i, όπου $i = 1, 2, \ldots, N$, τότε το $\ell_i \in 1, 2, \ldots, n$ είναι ο αριθμός των υποδιαστημάτων που περιέχουν την απόσταση του γιατρού. Στη συνέχεια, τα ciphertexts των πρώτων ℓ_i υποδιαστημάτων θα είναι ο κρυπτογραφημένος αριθμός "1". Για τα υπόλοιπα υποδιαστήματα κρυπτογραφείται ο αριθμός z, όπου $z > 1$ είναι μια γνωστή τιμή σε όλους τους agents. Ένα παράδειγμα ενός αρχικού μηνύματος παρουσιάζεται στο Σχήμα 5.5.



Σχήμα 5.5 Το αρχικό κρυπτογραφημένο μήνυμα.

Όταν ένας agent λαμβάνει ένα μήνυμα από κάποιον άλλο agent, υπολογίζει το νέο μήνυμα που θα πρέπει να προωθήσει στον επόμενο agent με τον ακόλουθο τρόπο: Κάθε ένα από τα πρώτα $\ell_i + 1$ ciphertexts του νέου προς αποστολή μηνύματος είναι το γινόμενο των αντίστοιχων ciphertexts του δικό του μηνύματος και του μηνύματος που έλαβε από τον άλλο agent. Τα υπόλοιπα $n-(\ell_i+1)$ ciphertexts του νέου προς αποστολή μηνύματος παίρνουν τη τιμή των αντίστοιχων ciphertexts του δικού του μηνύματος. Το αποτέλεσμα αυτού του υπολογισμού είναι ένα νέο προς αποστολή μήνυμα το οποίο διαβιβάζεται έπειτα στον επόμενο κόμβο ή κόμβους.

Ο κατανεμημένος αυτός υπολογισμός εκτελείται σε μια λογική δυαδική δεντρική τοπολογία στην οποία τα φύλλα του δέντρου είναι οι N agents των γιατρών. Επειδή, το βάθος του δέντρου είναι $\lfloor \log_2 N \rfloor + 1$ αυτό έχει σαν αποτέλεσμα ο συνολικός υπολογισμός να πραγματοποιηθεί μετά από $\lfloor \log_2 N \rfloor + 1$ παράλληλα βήματα.



Σχήμα 5.6 Το τελικό κρυπτογραφημένο μήνυμα.

Η γενική μορφή του τελικού κρυπτογραφημένου μηνύματος που προκύπτει από τον

παραπάνω υπολογισμό παρουσιάζεται στο Σχήμα 5.6. Εάν υποθέσουμε ότι L είναι ο αριθμός των πρώτων ciphertext που κρυπτογραφούν τον αριθμού "1".Τότε, η τιμή του L υποδηλώνει ότι τα πρώτα $L-1$ υποδιαστήματα είναι κενά (δηλ. δεν υπάρχει κανένας γιατρός στην απόσταση αυτών των υποδιαστημάτων) και το L υποδιάστημα είναι το πρώτο στο οποίο υπάρχει τουλάχιστον ένας γιατρός. Αντίστοιχα, ο εκθέτης k του αριθμού z στο υποδιάστημα $L+1$ υποδηλώνει τον αριθμό των γιατρών που υπάρχουν μέσα στο προηγούμενο υποδιάστημα (δηλ. το L). Τα ciphertexts των επόμενων $n-(L+1)$ υποδιαστημάτων είναι η κρυπτογράφηση μερικών τυχαίων δυνάμεων του z και δεν λαμβάνονται υπόψη. Τέλος, η υπηρεσία NSG αποκρυπτογραφεί το τελικό μήνυμα και βρίσκει πιο είναι το ελάχιστο διάστημα στο οποίο υπάρχουν γιατροί και τον αριθμό αυτών.

5.5.4 Onion Routing

Στη Φάση 2 του κατανεμημένου υπολογισμού χρησιμοποιείται το Onion Routing [36, 37], το οποίο αποτελεί μια δημοφιλή τεχνική για ανώνυμη επικοινωνία μέσω του διαδικτύου. Μια απλοποιημένη περιγραφή του Onion Routing είναι: Ένας κόμβος που θέλει να στείλει ένα μήνυμα σε έναν άλλον κόμβο, δεν στέλνει το μήνυμα απευθείας στον προορισμό του. Πιο συγκεκριμένα, ο αποστολέας επιλέγει μια τυχαία διαδρομή μέσω κάποιων ενδιάμεσων κόμβων και καταλήγει στον κόμβο του προορισμού. Επιπλέον, ο αποστολέας κρυπτογραφεί το μήνυμα πολλαπλές φορές με τα κλειδιά των ενδιάμεσων κόμβων. Έτσι το μήνυμα πακετάρεται με πολλαπλά στρώματα κρυπτογράφησης και παρομοιάζεται με ένα "κρεμμύδι" ("onion"). Κάθε ενδιάμεσος κόμβος που παραλαμβάνει το μήνυμα, αφαιρεί ένα στρώμα κρυπτογράφησης με σκοπό να βρει τις οδηγίες δρομολόγησης, και στέλνει το μήνυμα στον επόμενο κόμβο που επαναλαμβάνει με τη σειρά του αυτή τη διαδικασία. Αυτή η τεχνική αποτρέπει στους κόμβους που μεσολαβούν να γνωρίσουν τον αποστολέα, τον παραλήπτη αλλά και το ίδιο το περιεχόμενο του μηνύματος. Επιπλέον, ο κάθε κόμβος το μόνο πράγμα που γνωρίζει είναι ο προηγούμενος και ο επόμενος του κόμβο.

Το πλεονέκτημα που παρουσιάζει το Onion Routing είναι ότι δεν είναι απαραίτητο να εμπιστεύεται κανείς κάθε συνεργαζόμενο κόμβο/δρομολογητή, γιατί έστω και ένας ή περισσότεροι κόμβοι να είναι τίμιοι, τότε η ανώνυμη επικοινωνία μπορεί ακόμη να επιτευχτεί. Αυτό γίνεται γιατί κάθε κόμβος/δρομολογητής σε ένα δίκτυο onion routing δέχεται τα μηνύματα, τα επανακρυπτογραφεί και μετά τα διαβιβάζει σε άλλο onion κόμβο/δρομολογητή. Ένας επιτιθέμενος που έχει τη δυνατότητα να ελέγχει

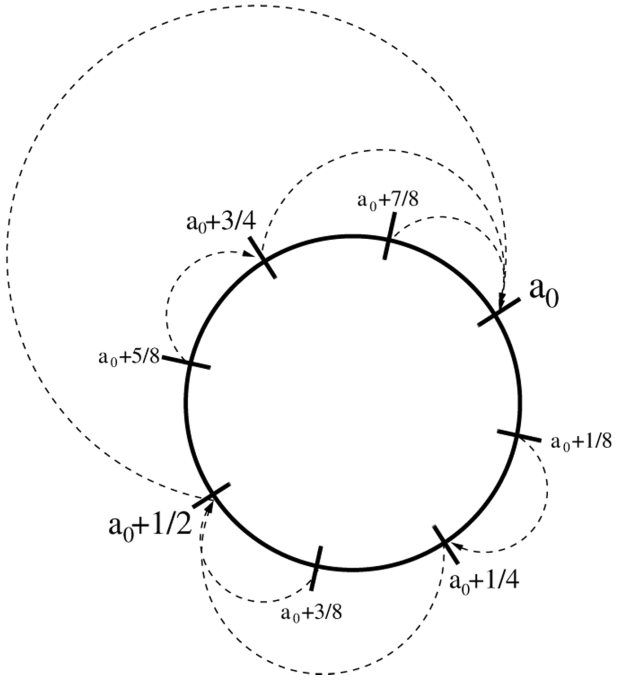
κάθε onion δρομολογητή σε ένα δίκτυο είναι ικανός να παρακολουθήσει την πορεία ενός μηνύματος στο δικτύου, αλλά ένας επιτιθέμενος με περιορισμένες όμως δυνατότητες θα δυσκολευτεί, εάν ελέγχει έναν ή περισσότερους κόμβους, να παρακολουθήσει ολόκληρη τη διαδρομή.

Προκειμένου να εξασφαλιστεί η ανωνυμία στην αποστολή ενός μηνύματος όπως περιγράφτηκε στη Φάση 2 (Ενότητα 5.5.2), μια λύση θα ήταν να χρησιμοποιήσει το δίκτυο Tor [38], το οποίο αποτελεί τη δεύτερη γενιά της πλατφόρμας Onion Routing. Το δίκτυο Tor είναι ευρέως διαδεδομένο, ως μια γενικού σκοπού πλατφόρμα για onion routing. Μια άλλη λύση θα ήταν να χρησιμοποιηθεί μια προσέγγιση του onion routing μεταξύ της κοινότητας των agents, όπου οι agents θα στέλνουν τα μήνυμά τους μέσω μιας τυχαίας διαδρομής από agents που θα επιλέγουν.

5.5.5 Δικτυακή τοπολογία

Ένα κρίσιμο κομμάτι της λύσης του NDP είναι η λογική δικτυακή τοπολογία των agents. Για αυτό το λόγο, χρησιμοποιούνται τεχνολογίες δικτύωσης από τα Peer-To-Peer (P2P) δίκτυα. Πιο συγκεκριμένα, εφαρμόζονται τεχνικές που αναπτύχτηκαν στα πλαίσια του Quantum P2P δικτύου [32] και το οποίο είναι βασισμένο στην αρχιτεκτονική του Chord [26].

Η τοπολογία του δικτύου που επιλέγθηκε έχει τα ακόλουθα χαρακτηριστικά: Οι agents οργανώνονται σε ένα λογικό δακτύλιο που χρησιμοποιείται ως θεμέλιο (ακρογωνιαίος λίθος) της τοπολογίας. Κάθε κόμβος στο δακτύλιο, γνωρίζει ποιος είναι ο προηγούμενος (predecessor) και ο επόμενος (successor) του κόμβος. Επιπρόσθετα, για να αυξηθεί η ανοχή της τοπολογίας σε αλλαγές/αποτυχίες κόμβων, κάθε κόμβος διατηρεί συνδέσεις σε ένα σύνολο από διαδοχικούς κόμβους. Επιπλέον, κάθε κόμβος διατηρεί ένα σύνολο συνδέσεων, που αποκαλούνται fingers, σε κόμβους που βρίσκονται σε γεωμετρικά αυξανόμενη απόσταση (δύναμη του δύο) μέσα στο δακτύλιο. Αυτές οι συνδέσεις επιτρέπουν στο δίκτυο να εμφανίζει τα χαρακτηριστικά μιας λογικής δυαδικής δεντρικής τοπολογίας [39]. Οι συνδέσεις των υπάρχοντων κόμβων (διαδοχικών και fingers) και η δημιουργία συνδέσεων με νέους κόμβους πραγματοποιείται με διαδικασίες σταθεροποίησης (stabilization) που είναι παρόμοιες με τις διαδικασίες σταθεροποίησης του Chord δικτύου. Η προτεινόμενη δικτυακή αρχιτεκτονική παρέχει μια πλήρως αποκεντροποιημένη και επεκτάσιμη (scalable) τοπολογία δικτύων για τους agents των γιατρών. Ένα παράδειγμα αυτής της δυαδικής δεντρικής τοπολογίας εκτέλεσης του κατανεμημένου υπολογισμού σε ένα λογικό δακτύλιο παρουσιάζεται στο Σχήμα 5.7.



Σχήμα 5.7 Δυαδική δεντρικής τοπολογίας εκτέλεσης του κατανεμημένου υπολογισμού σε δακτύλιο [39].

5.6 Ασφάλεια του κατανεμημένου υπολογισμού

Σε αυτή την ενότητα, δείχνεται ότι το προτεινόμενο πρωτόκολλο για τη λύση του NDP δεν παραβιάζει την ιδιωτικότητα της θέσης των γιατρών. Η ασφάλεια του πρωτοκόλλου στηρίζεται στο μοντέλο των "Honest-But-Curious" (HBC) χρηστών (Ορισμός 1). Κατά αρχήν θα πρέπει να επισημανθεί ότι κάθε γιατρός δεν χρησιμοποιεί (στον εκάστοτε υπολογισμό) την ακριβή του θέση παρά μόνο την απόστασή του από τη θέση του επείγοντος περιστατικού. Η ασφάλεια του κρυπτογραφικού συστήματος ElGamal και της ομομορφικής του ιδιότητας εξασφαλίζει ότι οι αποστάσεις που αντιπροσωπεύουν τους γιατρούς δεν μπορούν να συνδεθούν με τη ταυτότητα οποιοδήποτε γιατρού. Επιπλέον, η ασφάλεια του Onion Routing προστατεύει την ανωνυμία των κοντινότερων γιατρών με την αποκάλυψη της απόστασής τους στη Φάση 2. Παρακάτω, παρουσιάζονται λεπτομερώς οι χαρακτηριστικές ιδιότητες ασφαλείας της κάθε φάσης ξεχωριστά.

- **Φάση 1**
  - Κάθε γιατρός χρησιμοποιεί την ιδιωτική του θέση και τη θέση του επείγοντος περιστατικού για να υπολογίσει την απόστασή του. Μόνο αυτή η απόσταση

χρησιμοποιείται στο κατανεμημένο υπολογισμό και όχι η ιδιωτική του θέση. Επιπλέον, οι γιατροί επισημαίνουν κατά τη διάρκεια του υπολογισμού μόνο το διάστημα στο οποίο βρίσκεται η απόστασή τους, και όχι την ακριβή τιμή της απόστασής του.

- Κάθε agent που λαμβάνει ένα μήνυμα από κάποιον άλλο agent δεν μπορεί να αποκομίσει πληροφορίες σχετικές με το περιεχόμενο του μηνύματος, επειδή τα ciphertexts κρυπτογραφούνται με ElGamal κρυπτογράφηση.
- Όλα τα ciphertexts των μηνυμάτων, που ταξιδεύουν από κόμβο σε κόμβο, αλλάζουν σε κάθε κόμβο που περνούν, ακόμη και αν οι κόμβοι αυτοί βρίσκονται εκτός διαστήματος, πολλαπλασιάζοντας τα ciphertexts με τον κρυπτογραφημένο αριθμό "1".
- Στο τέλος κάθε γύρου της Φάσης 1, το τελικό μήνυμα αποκαλύπτει τον αριθμό των γιατρών στο κοντινότερο διάστημα που περιέχει κάποιο γιατρό. Με αυτό το τρόπο κανείς γιατρός δεν μπορεί να συνδεθεί με τη ταυτότητα του. Συνεπώς, η Φάση 1 διασφαλίζει k-anonymity (Ορισμός 2), όπου $k = N$ και N είναι το πλήθος των agents στο δίκτυο.
- Ωστόσο, σε κάθε γύρο αποκαλύπτονται στατιστικές πληροφορίες για τον αριθμό των γιατρών που βρίσκονται στο εκάστοτε κοντινότερο διάστημα. Αυτές οι πληροφορίες όμως δεν παραβιάζουν την ιδιωτικότητα θέσης των γιατρών.

- **Φάση 2**
  - Οι χαρακτηριστικές ιδιότητες ασφαλείας του Onion Routing εξασφαλίζουν ότι οι ακριβείς αποστάσεις των γιατρών που βρίσκονται στο κοντινότερο διάστημα (από τη Φάση 1) αποστέλλονται ανώνυμα στον root-κόμβο. Επιπρόσθετα, εξασφαλίζεται k-anonymity (για $k = N$) και σε αυτή τη φάση επίσης.
  - Το Onion Routing θεωρείται ότι λειτουργεί σωστά και αποτελεσματικά. Περισσότερες λεπτομέρειες σχετικά με την ασφάλεια του Onion Routing βρίσκονται στην αναφορά [38].

- **Φάση 3**
  - Σε αυτήν τη φάση, ανακοινώνεται στο δίκτυο των agents το τυχαίο ID που αντιπροσωπεύει το κοντινότερο γιατρό. Ο agent που θα αναγνωρίσει αυτό το

ID μπορεί να έρθει σε επαφή με την υπηρεσία NSG και να αποκαλύψει, εάν το επιθυμεί, τη ταυτότητά του, ενώ ταυτόχρονα διασφαλίζεται η ιδιωτικότητα θέσης όλων των άλλων γιατρών.

Ορισμός 1 (Honest-But-Curious (HBC)) *Ένας Honest-But-Curious [40] συμμετέχοντας (εχθρός) ακολουθεί πιστά το προκαθορισμένο πρωτόκολλο, αλλά μπορεί να κρατήσει τα ενδιάμεσα αποτελέσματα του υπολογισμού, π.χ. τα μηνύματα που ανταλλάσσονται, και να προσπαθήσει να εξαγάγει πρόσθετες πληροφορίες από αυτά.*

Ορισμός 2 (k-anonymity) *Ένας απλός ορισμός του k-anonymity στα πλαίσια του προβλήματος NDP είναι ότι δεν μπορούν να υπάρξουν λιγότεροι από k γιατροί (ως οντότητες) που μπορούν να συνδεθούν με μια συγκεκριμένη απόσταση. Ένας πιο γενικός ορισμός του k-anonymity που ισχύει επίσης και στις βάσεις δεδομένων δίνεται στο [41].*

5.7 Πειραματικά αποτελέσματα

Για την επιβεβαίωση της ρεαλιστικότητας της λύσης του NDP και την εξέταση της πρακτικής εφαρμογής του αναπτύχθηκε προγραμματιστικά μια πρότυπη εφαρμογή του NDP. Η εφαρμογή αυτή υλοποιήθηκε σε Java και για τις κρυπτογραφικές ανάγκες του πρωτοκόλλου χρησιμοποιήθηκε η βιβλιοθήκη της Bouncycastle [42]. Επίσης, ως προσωπικός agent για τη διαχείριση των προσωπικών δεδομένων των γιατρών χρησιμοποιήθηκε ο προσωπικός agent που αναπτύχθηκε για την πλατφόρμα του Polis (Ενότητα 2.11). Σε αυτήν εδώ τη προσέγγιση, η διαχείριση της τρέχουσας θέσης κάθε γιατρού πραγματοποιείτε από έναν Polis agent.

Σε αυτή την πρότυπη εφαρμογή, το κρυπτογραφικό πρωτόκολλο είναι πλήρως υλοποιημένο με τη χρήση έτοιμων βιβλιοθηκών. Σε αυτό το σημείο της ανάπτυξης οι λειτουργίες της δικτυακής τοπολογίας δεν είναι ακόμη πλήρως υλοποιημένες. Πιο συγκεκριμένα, το δίκτυο χρησιμοποιεί μόνο τη τοπολογία δακτυλίου χωρίς όμως αυτό να επηρεάζει τη σωστή εκτέλεση του κατανεμημένου υπολογισμού. Οι επιπτώσεις που έχει στην εκτέλεση, η χρήση της απλής αυτής τοπολογίας, αφορά περισσότερο τη ταχύτητα/απόδοση του υπολογισμού και την ανοχή σε πιθανές αλλαγές/αποτυχίες κάποιων κόμβων.

Ακόμα και σε αυτό το σημείο της ανάπτυξης, ο αριθμός των agent περιορίζεται μόνο από τεχνικά ζητήματα σχετικά με τη διαχείριση μεγάλου αριθμού agents σε πειραματικό περιβάλλον. Στο πείραμα που εκτελέστηκε για τη λύση του NDP συμμετείχαν συνολικά

30 agents και χρειάστηκαν 40 δευτερόλεπτα για την πραγματοποίηση του υπολογισμού. Παρακάτω περιγράφεται αναλυτικά ένα αντίστοιχο πείραμα που συμμετείχαν 4 agents και η υπηρεσία NSG.

Το πείραμα καλύπτει συνολικά μια τετραγωνική έκταση των 10000 km^2 και οι θέσεις των γιατρών και του επείγοντος περιστατικού επιλέγονται τυχαία να βρίσκονται μέσα στην ανωτέρω περιοχή. Κάθε ένας agent επίλεγει τυχαία τη θέση του και η υπηρεσία NSG είναι αυτή που επιλέγει τυχαία τη θέση του επείγοντος περιστατικού. Η λύση του NDP προσπαθεί να βρει μέσα σε μια απόσταση το πολύ των 75 km το κοντινότερο γιατρό. Οι τιμές των εσωτερικών παραμέτρων για τη συγκεκριμένη περίπτωση επιλέχθηκαν να είναι $K = 2$ και $z = 2$.

Η υπηρεσία NSG επιλέγει σε αυτή την περίπτωση ως root-κόμβο τον $Agent\_1$ και διαβιβάζει τη θέση του επείγοντος περιστατικού σε αυτόν. Οι συντεταγμένες της θέσης του επείγοντος περιστατικού είναι $L_{em} = [41.140110, 24.913660]$ και οι ακριβείς αποστάσεις των 4 agents από αυτή τη θέση του επείγοντος περιστατικού είναι:

$$Agent\_1 \Rightarrow 17.544817\ km$$
$$Agent\_2 \Rightarrow 53.157742\ km$$
$$Agent\_3 \Rightarrow 25.797003\ km$$
$$Agent\_4 \Rightarrow 66.221868\ km$$

Το κρυπτογραφικό πρωτόκολλο αρχίζει με τη Φάση 1. Στον πρώτο γύρο, το διάστημα [0, 75] km χωρίζεται σε 5 ίσα διαστήματα και κατά συνέπεια η κρυπτογραφημένη αναπαράσταση των αποστάσεων των agents (σε km) είναι:

	$0-15$	$15-30$	$30-45$	$45-60$	$60-75$
$Agent\_1$	$E(1)$	$E(1)$	$E(2)$	$E(2)$	$E(2)$
$Agent\_2$	$E(1)$	$E(1)$	$E(1)$	$E(1)$	$E(2)$
$Agent\_3$	$E(1)$	$E(1)$	$E(2)$	$E(2)$	$E(2)$
$Agent\_4$	$E(1)$	$E(1)$	$E(1)$	$E(1)$	$E(1)$

Το τελικό κρυπτογραφημένο μήνυμα του $1^{ου}$ γύρου είναι:

	$0-15$	$15-30$	$30-45$	$45-60$	$60-75$
$result$	$E(1)$	$E(1)$	$E(2^2)$	$E(2)$	$E(2)$

Η αποκρυπτογράφηση των ciphertexts του τελικού μηνύματος αποκαλύπτει ότι το κοντινότερο διάστημα μέσα στο οποίο υπάρχουν γιατροί είναι το [15, 30) (σε km) και στο οποίο υπάρχουν συγκεκριμένα δύο γιατροί. Από τη στιγμή που ο αριθμός

των γιατρών στο κοντινότερο διάστημα είναι μικρότερος ή ίσος από το K, η Φάση 1 ολοκληρώνεται. Στη Φάση 2, ο root-κόμβος με broadcast μήνυμα αποστέλλει αυτό το διάστημα σε όλους τους κόμβους. Κάθε ένας από τους δύο κόμβους που βρίσκεται σε αυτό το διάστημα, αποστέλλει την ακριβή απόσταση του μαζί με ένα τυχαίο ID στην υπηρεσία NSG. Οι κόμβοι χρησιμοποιούν το Onion Routing για να στείλουν το μήνυμά τους ανώνυμα.

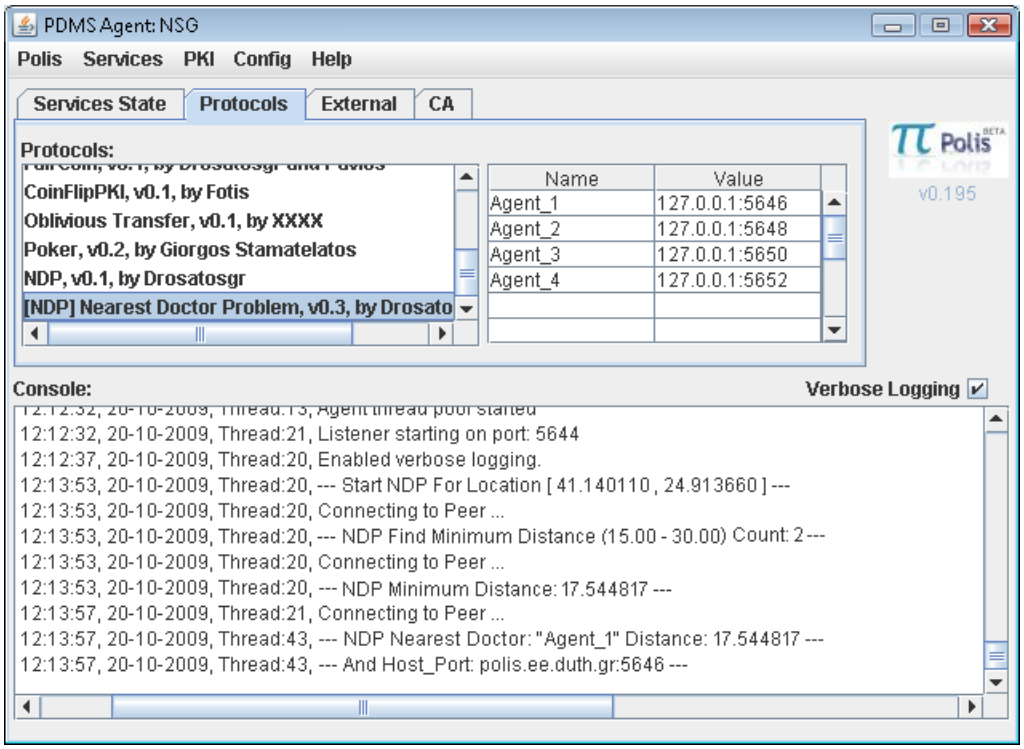
Η υπηρεσία NSG λαμβάνει τις ακόλουθες δύο ακριβείς αποστάσεις με τα αντίστοιχα τυχαία ID:

$$[Dist = 17.544817, ID = 56770656]$$
$$[Dist = 25.797003, ID = 45413392]$$

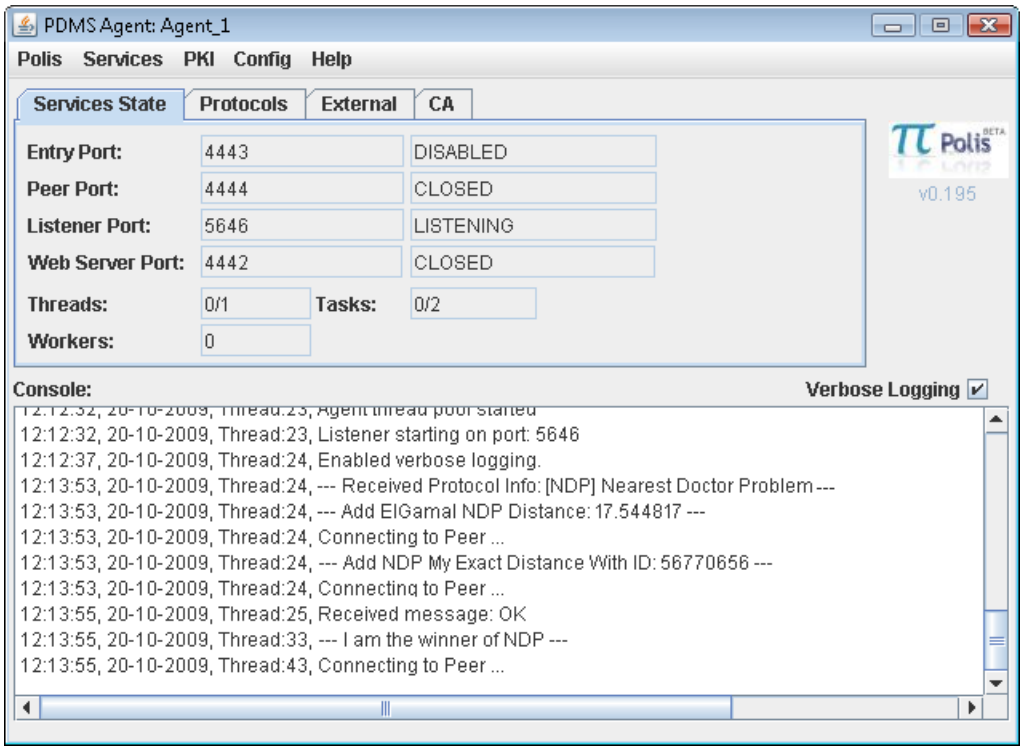
Η υπηρεσία NSG από αυτές τις δύο αποστάσεις διαπιστώνει ότι η ελάχιστη απόσταση είναι 17.544817 km. Στη Φάση 3, η υπηρεσία NSG αποστέλλει το τυχαίο αναγνωριστικό ID που αντιστοιχεί σε αυτή την ελάχιστη απόσταση στον root-κόμβο, ο οποίος μεταβιβάζει με broadcast μήνυμα αυτό το ID στο δίκτυο των γιατρών.

$$ID : 56770656$$

Τελικά, ο $Agent\_1$ αναγνωρίζει ότι είναι ο κοντινότερος γιατρός και έρχεται άμεσα σε επαφή με την υπηρεσία NSG για να προσφέρει τις υπηρεσίες του. Στο Σχήμα 5.8 παρουσιάζεται ένα στιγμιότυπο της υπηρεσίας NSG κατά τη φάση εκτέλεσης του πειράματος και στο Σχήμα 5.9 παρουσιάζεται ένα στιγμιότυπο του $Agent\_1$.



Σχήμα 5.8 Στιγμιότυπο του NDP Service Gateway (NSG).



Σχήμα 5.9 Στιγμιότυπο του *Agent_1*.

Κεφάλαιο 6

Συμπεράσματα

Τα πειραματικά αποτελέσματα της πρότυπης υλοποίησης του NDP επιβεβαίωσαν ότι πράγματι είναι εφικτό να εφαρμοστεί η λύση που προτείνεται για το NDP. Ωστόσο, υπάρχουν ακόμη ανοιχτά σημαντικά ζητήματα, τόσο τεχνικά όσο και μη τεχνικά. Ένα από τα ζήτημα αφορά την πιθανότητα αποδοχής ή όχι μιας τέτοιας λύσης, όπως του NDP, από τις κοινότητες των γιατρών ή άλλων κοινωνικών ομάδων που θα μπορούσαν να χρησιμοποιήσουν ένα τέτοιο σύστημα. Προφανώς, υπάρχουν και αρκετά άτομα που μπορεί να μην είναι πρόθυμα να υιοθετήσουν τέτοιου είδους τεχνολογίες, όπως η λύση του NDP, στη καθημερινή τους ζωή. Ωστόσο, τέτοιου είδους δυσκολίες συνήθως υπάρχουν σε κάθε νέα τεχνολογία που πάει να εφαρμοστεί στην πράξη. Πιστεύουμε μάλιστα ότι η κοινότητα των γιατρών δεν θα πρέπει να αισθάνεται ότι απειλείται σε καμία περίπτωση μέσα από μια εφαρμογή όπως αυτή του NDP, για τους ακόλουθους λόγους:

1. Η ιδιωτικότητα της θέσης του κάθε γιατρού προστατεύεται από τη λύση του NDP και ταυτόχρονα βρίσκεται κάτω από τον απόλυτο έλεγχο του ίδιου του γιατρού.

2. Η λύση που προτείνεται είναι απλή και αρκετά φτηνή ώστε να είναι εφικτή ακόμη και με τις τρέχουσες τεχνολογίες πληροφορικής και επικοινωνιών (ICT).

3. Τα οφέλη μιας εφαρμογής όπως το NDP για τη δημόσια υγεία είναι εξαιρετικά σημαντικά.

Βέβαια, υπάρχουν ακόμη διάφορα ζητήματα που πρέπει να αναφερθούν. Ακόμη και αν οι ασύρματες επικοινωνίας όπως το 3G, το Wi-Fi και οι δορυφορικές επικοινωνίες είναι πλέον ευρέως διαθέσιμες υπάρχουν ακόμη τεχνικά και οικονομικά ζητήματα. Για

παράδειγμα, μια προσωπική φορητή συσκευή (π.χ. ένα κινητό τηλέφωνο) θα πρέπει να ενημερώνει σε τακτά χρονικά διαστήματα τη θέση του γιατρού στον προσωπικό του agent που διαχειρίζεται τα προσωπικά του δεδομένα. Αυτό μπορεί να προκαλέσει μεγάλη κατανάλωση ενέργειας στη φορητή συσκευή και ταυτόχρονα μεγάλο κόστος για την ασύρματη μεταφορά δεδομένων με αποτέλεσμα να καθίσταται τη χρήση του απαγορευτική. Ωστόσο, οι τρέχουσες εξελίξεις στη τεχνολογία των φορητών συσκευών και των τηλεπικοινωνιακών υπηρεσιών προδιαθέτουν ότι αυτά τα ζητήματα/περιορισμοί θα ξεπεραστούν σχετικά σύντομα.

Επιπρόσθετα, σημαντικές τεχνικές δυσκολίες υπάρχουν και όσον αφορά την ανοχή της προτεινόμενης λύσης σε πιθανές απώλειες κόμβων (fault tolerance) και την επεκτασιμότητα (scalability) της τοπολογίας του δικτύου. Μια αναζήτηση του κοντινότερου γιατρού μπορεί να πάρει πάρα πολύ χρόνο εάν το πλήθος των γιατρών μιας κοινότητας είναι της τάξεως των πολλών εκατοντάδων και παραπάνω. Επιπλέον, μια πιθανή αποτυχία κάποιων κόμβων είναι περισσότερο πιθανή όσο το δίκτυο γίνεται μεγαλύτερο και μπορεί να επηρεάσει στο σύνολο την εκτέλεση του κατανεμημένου υπολογισμού. Ωστόσο, πιστεύεται ότι η πλήρως αναπτυγμένη δικτυακή πλατφόρμα η οποία θα βασίζεται στο Quantum και στο Chord θα μπορέσει να λύσει αυτά τα τεχνικά ζητήματα της δικτυακής τοπολογίας.

Τέλος, η χρησιμοποίηση μιας πιο κατάλληλης μετρικής απόστασης θα μπορούσε να ήταν μια πιθανή βελτίωση της λύσης του NDP. Στην παρούσα υλοποίηση, γίνεται χρήση της great-circle απόστασης (που χρησιμοποιείτε για την εύρεση της απόστασης δύο σημείων στην επιφάνεια μιας σφαίρας). Για παράδειγμα, θα μπορούσε να χρησιμοποιηθεί λογισμικό πλοήγησης (Navigation) στη πλευρά του agent που να χρησιμοποιεί τη GPS θέση (του γιατρού και του επείγοντος περιστατικού) για να υπολογίσει το χρόνο που ο γιατρός θα χρειαζόταν για να φθάσει στη θέση του επείγοντος περιστατικού. Ένας τέτοιος υπολογισμός της "απόστασης" θα μπορούσε να ήταν πιο αποτελεσματικός για το NDP.

Σε μελλοντικές εργασίες, κρίνεται αναγκαίο να πραγματοποιηθούν διάφορες επεκτάσεις, όπως την ολοκλήρωση της υλοποίησης της δικτυακής τοπολογίας, με τέτοιο τρόπο ώστε να μπορεί να είναι δυνατή η εκτέλεση του κατανεμημένου υπολογισμού με τη λογική του δυαδικού δέντρου μέσα στο P2P δίκτυο. Αυτό θα επιτρέψει την πραγματοποίηση ενός μεγαλύτερου και πιο ρεαλιστικού συνόλου πειραμάτων, υποστηρίζοντας αποδοτικότερη εκτέλεση κατανεμημένων υπολογισμών μεταξύ των εκατοντάδων ή ακόμα και χιλιάδων agents. Μια ακόμα σημαντική κατεύθυνση έρευνας αφορά σε θέματα ιδιωτικότητας και ασφαλείας του NDP κάτω από διαφορετικές περιπτώσεις

ασφάλειας, όπως για παράδειγμα με την υπόθεση ότι υπάρχει και κάποιος αριθμός από κακόβουλους (malicious) κόμβους στο δίκτυο των γιατρών. Κλείνοντας, θα πρέπει επίσης να διερευνηθεί η πιθανότητα διαρροής της ιδιωτικότητας ύστερα από την εκτέλεση ενός μεγάλου αριθμού διαδοχικών αιτημάτων αναζήτησης.

Βιβλιογραφία

[1] S. D. Warren and L. D. Brandeis, "The Right to Privacy" Harvard Law Review, Vol. IV, No. 5 (December 15, 1890).

[2] A. Westin, "Privacy and Freedom" New York, U.S.A.: Atheneum (1967).

[3] Wikipedia, "Informational self-determination", March 2010, http://en.wikipedia.org/wiki/Informational_self-determination.

[4] Αρχή Προστασίας Δεδομένων Προσωπικού Χαρακτήρα, "Νομοθεσία σχετικά με την προστασία προσωπικών δεδομένων", Φεβρουάριος 2010, http://www.dpa.gr.

[5] B. Schneier, *Applied Cryptography*, 2nd ed. (John Wiley & Sons, Inc., 1996).

[6] Wikipedia, "SHA hash functions", February 2010, http://en.wikipedia.org/wiki/SHA_hash_functions.

[7] Wikipedia, "RSA", February 2010, http://en.wikipedia.org/wiki/RSA.

[8] Β.Α. Κάτος, και Γ.Χ. Στεφανίδης, *Τεχνικές Κρυπτογραφίας & Κρυπτανάλυσης* (Εκδόσεις Ζυγός, 2003).

[9] Wikipedia, "ElGamal Encryption", February 2010, http://en.wikipedia.org/wiki/ElGamal_encryption.

[10] Wikipedia, "Homomorphic encryption", February 2010, http://en.wikipedia.org/wiki/Homomorphic_encryption.

[11] C. Gentry, "Fully homomorphic encryption using ideal lattices" In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pp. 169–178 (ACM, New York, NY, USA, 2009).

[12] Wikipedia, "Public Key Certificate", February 2010, http://en.wikipedia.org/wiki/Public_key_certificate.

[13] Wikipedia, "Transport Layer Security", February 2010, http://en.wikipedia.org/wiki/Transport_Layer_Security.

[14] Wikipedia, "Secure multi-party computation", February 2010, http://en.wikipedia.org/wiki/Secure_multi-party_computation.

[15] A. C. Yao, "Protocols for Secure Computations (extended abstract)" Proceedings of the 21st Annual IEEE Symposium on the Foundations of Computer Science pp. 160–164 (1982).

[16] P. Bogetoft *et al.*, "Multiparty Computation Goes Live", Cryptology ePrint Archive, Report 2008/068, 2008, http://eprint.iacr.org/.

[17] P. S. Efraimidis, G. Drosatos, F. Nalbadis, and A. Tasidou, "Towards Privacy in Personal Data Management" Journal on Information Management & Computer Security 17 (2009).

[18] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system" ACM Transactions on Information Systems (TOIS), Vol. 10 pp. 91–102 (1992).

[19] A. Ward, A. Jones, and A. Hopper, "A New Location Technique for the Active Office" IEEE Personal Communications, Vol. 4 pp. 42–47 (1997).

[20] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System" In *6th ACM MOBICOM*, (Boston, MA, 2000).

[21] Wikipedia, "Global Positioning System", February 2010, http://el.wikipedia.org/wiki/Global_Positioning_System.

[22] A. Oram, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies* (O'Reilly & Associates, 2001).

[23] K. Aberer and M. Hauswirth, *P2P Systems* (CRC press, 2004), Vol. Practical Handbook of Internet Computing.

[24] M. A. Jovanovic, F. S. Annexstein, and K. A. Berman, "Scalability Issues in Large Peer-to-Peer Networks - A Case Study of Gnutella" University of Cincinnati, Laboratory for Networks and Applied Graph Theory (2001).

[25] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks" SIGMETRICS pp. 258–259 (2002).

[26] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications" In *ACM SIGCOMM'01*, pp. 149–160 (2001).

[27] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems" In *Accepted for Middleware*, (2001).

[28] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network" In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 161–172 (ACM, New York, NY, USA, 2001).

[29] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System" In *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability (LLNCS)*, (2001).

[30] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric" In *Proceedings of IPTPS*, pp. 53–65 (2002).

[31] K. Aberer, P. Cudre-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, "P-Grid: a self-organizing structured P2P system" SIGMOD Record, Vol. 32 pp. 29–33 (2003).

[32] G. Stamatelatos, G. Drosatos, and P. S. Efraimidis, "Quantum: A Peer-to-Peer Network for Distributed Computations with Enhanced Privacy" In *EYRHKA 2009 Conference Proceedings*, pp. 201–210 (2009).

[33] Europe's Information Society, "eSafety", 2009, http://ec.europa.eu/esafety.

[34] M. Yokoo and K. Suzuki, "Secure Multi-agent Dynamic Programming based on Homomorphic Encryption and its Application to Combinatorial Auctions" In *AAMAS'02*, (2002).

[35] D. Bickson, D. Dolev, G. Bezman, and B. Pinkas, "Peer-to-Peer Secure Multi-party Numerical Computation" IEEE International Conference on Peer-to-Peer Computing pp. 257–266 (2008).

[36] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing" IEEE Journal on Selected Areas in Communications 16 (1998).

[37] Wikipedia, "Onion routing", February 2010, http://en.wikipedia.org/wiki/Onion_routing.

[38] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router" In *Proceedings of the 13th USENIX Security Symposium*, pp. 303–320 (2004).

[39] D. R. Karger and M. Ruhl, "Diminished Chord: A Protocol for Heterogeneous Subgroup Formation in Peer-to-Peer Networks" In *IPTPS*, pp. 288–297 (2004).

[40] A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. De Capitani di Vimercati, *Digital privacy* (Auerbach Publications, Taylor & Francis Group, 6000 Broken Sound ParkWay NW, 2008).

[41] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati, *Advances in Information Security* (Springer US, 2007), No. 4, pp. 323–353.

[42] D. Hook, *Beginning Cryptography with Java* (Wiley Publishing, Inc., Indianapolis, In 46256, USA, 2005).

Παράρτημα Α

Κώδικας υλοποίησης σε Java

Α.1 Κώδικας υπολογισμού great-circle απόστασης

```
package polis.protocols.ndpElGamal.findRadialDistance;

import java.io.Serializable;
import java.util.Random;

public class Lat_Lon implements Serializable{
        //Latitude - Longitude Greece
        public static final double MinLat = 34.0;
        public static final double MaxLat = 41.0;
        public static final double MinLon = 20.0;
        public static final double MaxLon = 28.0;

        public double lat = 0.0;
        public double lon = 0.0;

        public Lat_Lon(double lat, double lon){
                this.lat = lat;
                this.lon = lon;
        }

        public Lat_Lon(){
                Random generator = new Random();
                this.lat = generator.nextDouble()*(MaxLat-MinLat) + MinLat;
                this.lon = generator.nextDouble()*(MaxLon-MinLon) + MinLon;
        }

        public double getLatitude(){
                return this.lat;
        }

        public double getLongitude(){
                return this.lon;
```

```
        }

        public void setLatitude(double lat){
                this.lat = lat;
        }

        public void setLongitude(double lon){
                this.lon = lon;
        }

        public String toString(){
                return "[ "+this.lat+" , "+this.lon+" ]";
        }

        public double getRadialDistance_HaversineFormula(Lat_Lon currentLocation){
                double R = 6371; // earthŏs radius to km
                // Haversine formula ////////////////////////////////////
                double dLat = toRad(this.lat-currentLocation.lat);
                double dLon = toRad(this.lon-currentLocation.lon);
                double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
                Math.cos(toRad(this.lat)) * Math.cos(toRad(currentLocation.lat)) *
                Math.sin(dLon/2) * Math.sin(dLon/2);
                double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
                return (R * c);
        }


        public double getRadialDistance_SphericalLawOfCosines(Lat_Lon currentLocation){
                double R = 6371; // earthŏs radius to km
                // Spherical law of cosines ////////////////////////////////
                return Math.acos(
                        Math.sin(toRad(this.lat))*Math.sin(toRad(currentLocation.lat)) +
                        Math.cos(toRad(this.lat))*Math.cos(toRad(currentLocation.lat)) *
                        Math.cos(toRad(currentLocation.lon-this.lon))) * R;
        }


        private double toRad(double degress) {  // convert degrees to radians
                return degress * Math.PI / 180;
        }

}
```

A.2 Κώδικας της ταξινομημένης λίστας των ElGamal ciphertexts

```
package polis.protocols.ndpElGamal.elGamalRequirements;

import java.io.Serializable;
import java.math.BigInteger;
```

```
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.util.ArrayList;
import java.util.List;
import org.bouncycastle.jce.interfaces.ElGamalPrivateKey;
import org.bouncycastle.jce.interfaces.ElGamalPublicKey;
import polis.protocols.ndpElGamal.findRadialDistance.Lat_Lon;

public class ListElGamalDistance implements Serializable {

        private static final long serialVersionUID = -186938350866972758L;

        public List ListElGamal = null;
        public PublicKey pkey = null;
        public Lat_Lon requestedLocation = null;
        public static final String NameProtocol = "ListElGamalDistance";

        private static final BigInteger ZERO = BigInteger.valueOf(0);
        private static final BigInteger ONE = BigInteger.valueOf(1);
        private static final BigInteger TWO = BigInteger.valueOf(2);

        private static final BigInteger Z = BigInteger.valueOf(2);

        public ListElGamalDistance(Lat_Lon requestedLocation, PublicKey pkey,
                        double minvalue, double maxvalue, double parts) {
                this.requestedLocation = requestedLocation;
                this.pkey = pkey;
                ListElGamal = new ArrayList();
                loadConstractor(minvalue, maxvalue, parts);
        }

        private void loadConstractor(double minvalue, double maxvalue, double parts) {
                double step = (maxvalue - minvalue) / (parts - 1.0);
                for (double i = minvalue; i < maxvalue; i = i + step) {
                        ListElGamal.add(new CurrentDistance(encrypt(ONE), i));
                }
                if (ListElGamal.size() < parts) {
                        ListElGamal.add(new CurrentDistance(encrypt(ONE), maxvalue));
                }
        }

        public CipherTextElGamal encrypt(BigInteger value) {
                ElGamalPublicKey key = (ElGamalPublicKey) pkey;
                BigInteger p = key.getParameters().getP();

                if (value.compareTo(p) >= 0) {
                        System.err.println("Input too large for ElGamal cipher.");
                }

                int pBitLength = p.bitLength();
                SecureRandom random = new SecureRandom();
                BigInteger k = new BigInteger(pBitLength, random);
```

```
        while (k.equals(ZERO) || (k.compareTo(p.subtract(TWO)) > 0)) {
                k = new BigInteger(pBitLength, random);
        }

        BigInteger g = key.getParameters().getG();
        return new CipherTextElGamal(value.multiply(key.getY().modPow(k, p))
                        .mod(p), g.modPow(k, p));
}

public BigInteger decrypt(PrivateKey key, CipherTextElGamal cipher) {
        BigInteger m = null;
        ElGamalPrivateKey privKey = (ElGamalPrivateKey) key;
        BigInteger biGamma = cipher.getGAMMA();
        BigInteger biPhi = cipher.getPHI();
        BigInteger p = privKey.getParameters().getP();
        m = biGamma.modPow(p.subtract(ONE).subtract(privKey.getX()), p)
                        .multiply(biPhi).mod(p);
        return m;
}

public double addYourDistance(Lat_Lon MyLocation) {
        double distance = this.requestedLocation
                        .getRadialDistance_HaversineFormula(MyLocation);
        double first = 0;
        double second = 0;
        for (int i = 0; i < ListElGamal.size(); i++) {
                CipherTextElGamal newcipher = null;
                second = ((CurrentDistance) ListElGamal.get(i)).getDistance();
                if(second<=distance||(first < distance && distance <= second)){
                        newcipher = encrypt(ONE);
                } else {
                        newcipher = encrypt(Z);
                }
                first = second;
                ((CurrentDistance) ListElGamal.get(i)).getCipherTextElGamal()
                                .setPHI(
                                ((CurrentDistance) ListElGamal.get(i))
                                        .getCipherTextElGamal().getPHI()
                                        .multiply(newcipher.getPHI()));
                ((CurrentDistance) ListElGamal.get(i)).getCipherTextElGamal()
                                .setGAMMA(
                                ((CurrentDistance) ListElGamal.get(i))
                                        .getCipherTextElGamal().getGAMMA()
                                        .multiply(newcipher.getGAMMA()));
        }
        return distance;
}

public double[][] findResults(PrivateKey key) {
        double[][] ret = new double[ListElGamal.size()][2];
        for (int i = 0; i < ListElGamal.size(); i++) {
                ret[i][0]=((CurrentDistance)ListElGamal.get(i)).getDistance();
```

```
                        ret[i][1]=decrypt(key,
                                        ((CurrentDistance) ListElGamal.get(i))
                                        .getCipherTextElGamal()).doubleValue();
                }
                return ret;
        }

        public double[] findMinimum(PrivateKey key) {
                double min = 0;
                double max = 0;
                double count = 0;
                double[][] ret = findResults(key);
                for (int i = 0; i < ret.length; i++) {
                        if (ret[i][1] == 1) {
                                max = ret[i][0];
                                if (i > 0) {
                                        min = ret[i - 1][0];
                                } else {
                                        min = 0;
                                }
                                if (i < (ret.length - 1)) {
                                        count=(Math.log(ret[i+1][1])/Math.log(2.0));
                                } else {
                                        count = 0;
                                }
                        }
                }
                return new double[] { min, max, count };
        }

}
```

A.3 Κώδικας του πρωτοκόλλου που εκτελείται στον root-κόμβο

```
public Object receive() throws Exception {
        packet = (NDPPacket) con.objectInputStream.readObject();
        ListLocalInfoElGamalProtocol info = new ListLocalInfoElGamalProtocol(
                                        agent.entity.getEntityDirectory());
        InfoElGamalProtocol iep = info.FindRequest(packet.HashRequest);
        if (iep != null) {
        // Check if the nood is the root-node
        if (packet.NameProtocol.equals(ListElGamalDistance.NameProtocol)) {
                double[] ret = ((ListElGamalDistance) packet.obj)
                                .findMinimum(iep.pair.getPrivate());
                agent.logLine("--- ElGamal NDP Find Minimum Distance ("
                                + ret[0] + " - " + ret[1] + ") Count: " + ret[2]
                                + " ---");
                if (ret[2] > 5) {
                        // Run again Phase 1
```

```
                ListElGamalDistance listelgamal = new ListElGamalDistance(
                        ((ListElGamalDistance) packet.obj).requestedLocation,
                        ((ListElGamalDistance) packet.obj).pkey, ret[0],
                        ret[1], parts);
                RoutingTable rt = new RoutingTable();
                for (int i = 0; i < parameterName.length - 3; i++) {
                        if (!getParameterValue(i).equals("")) {
                                rt.addNode(getParameterValue(i));
                        }
                }
                rt.addNode(agent.agentData.listenerExternalAddress.host
                                + ":" + agent.agentData.listenerPort);
                packet.obj = null;
                packet.obj = listelgamal;
                packet.routingtable = null;
                packet.routingtable = rt;
        } else if (ret[2] > 0 && ret[2] <= 5) {
                // Start Phase 2
                ReservoirElGamalProtocol rep = new ReservoirElGamalProtocol(
                                iep.getLocation(), iep.getKeyPair().getPublic(),
                                ret[0], ret[1], 5);
                RoutingTable rt = new RoutingTable();
                for (int i = 0; i < parameterName.length - 3; i++) {
                        if (!getParameterValue(i).equals("")) {
                                rt.addNode(getParameterValue(i));
                        }
                }
                rt.addNode(agent.agentData.listenerExternalAddress.host
                                + ":" + agent.agentData.listenerPort);
                packet.obj = null;
                packet.obj = rep;
                packet.NameProtocol = ReservoirElGamalProtocol.NameProtocol;
                packet.routingtable = null;
                packet.routingtable = rt;
        } else if (ret[2] == 0) {
                // Run again Phase 1 with bigger interval
                if (0 == JOptionPane.showConfirmDialog(
                        null,
                        "I can't find anyone to this distance! " +
                        "Do you want to continue with double Max Value?",
                        "Question", 0)) {
                ListElGamalDistance listelgamal = new ListElGamalDistance(
                        ((ListElGamalDistance) packet.obj).requestedLocation,
                        ((ListElGamalDistance) packet.obj).pkey,
                        ret[0], ret[1] * 2.0, parts);
                        RoutingTable rt = new RoutingTable();
                        for (int i = 0; i < parameterName.length - 3; i++) {
                                if (!getParameterValue(i).equals("")) {
                                        rt.addNode(getParameterValue(i));
                                }
                        }
                rt.addNode(agent.agentData.listenerExternalAddress.host
                                + ":" + agent.agentData.listenerPort);
```

```
                                packet.obj = null;
                                packet.obj = listelgamal;
                                packet.routingtable = null;
                                packet.routingtable = rt;
                        }
                }
        }else if(packet.NameProtocol.equals(ReservoirElGamalProtocol.NameProtocol)){
                // Start Phase 3
                double min = ((ReservoirElGamalProtocol) packet.obj)
                                .getMinimumDistance(iep.getKeyPair().getPrivate());
                agent.logLine("--- ElGamal NDP Minimum Distance: " + min
                                + " ---");
                iep.setWinDistance(min);
                info.saveToFile(agent.entity.getEntityDirectory());
                BroadCastProtocol bcp = new BroadCastProtocol(
                                packet.HashRequest, iep.getLocation(), CreateAHash
                                        .byteToBase64(CreateAHash.getHash(String
                                                .valueOf(min))),
                                agent.agentData.listenerExternalAddress.host + ":"
                                                + agent.agentData.listenerPort);
                RoutingTable rt = new RoutingTable();
                for (int i = 0; i < parameterName.length - 3; i++) {
                        if (!getParameterValue(i).equals("")) {
                                rt.addNode(getParameterValue(i));
                        }
                }
                packet.obj = null;
                packet.obj = bcp;
                packet.NameProtocol = BroadCastProtocol.NameProtocol;
                packet.routingtable = null;
                packet.routingtable = rt;
        }else if(packet.NameProtocol.equals(DirectAnswerToBroadCast.NameProtocol)){
                // Find the nearest doctor
                if (((DirectAnswerToBroadCast) packet.obj).mydistance == iep
                                .getWinDistance()) {
                        agent.logLine("--- ElGamal NDP Winner: \""
                                + ((DirectAnswerToBroadCast) packet.obj).EntityName
                                + "\" Distance: "
                                + ((DirectAnswerToBroadCast) packet.obj).mydistance
                                + " ---");
                        agent.logLine("--- And Host Port: "
                                + ((DirectAnswerToBroadCast) packet.obj).hostportWinner
                                + " ---");
                } else {
                        agent.logLine("--- Probably \""
                                + ((DirectAnswerToBroadCast) packet.obj).EntityName
                                + "\" is Malicious! ---");
                }
        }
        }
        this.send();
        return null;
}
```

A.4 Κώδικας του πρωτοκόλλου που εκτελείται στους ερωτηθέντες κόμβους

```
public Object receive() throws Exception {
        packet = (NDPPacket) con.objectInputStream.readObject();
        ListOfTempLocations lotl = new ListOfTempLocations(agent.entity
                        .getEntityDirectory());
        Lat_Lon mylocation = null;
        if (lotl.existHashRequest(packet.HashRequest)) {
                mylocation = lotl.FindMyLocation(packet.HashRequest);
        } else {
                mylocation = new Lat_Lon();
                lotl.addMyLocation(packet.HashRequest, mylocation);
        }
        lotl.saveToFile(agent.entity.getEntityDirectory());
        if (packet.NameProtocol.equals(ListElGamalDistance.NameProtocol)) {
                // Start Phase 1
                double distance = ((ListElGamalDistance) packet.obj)
                                .addYourDistance(mylocation);
                agent.logLine("--- Add ElGamal NDP Distance (" + distance
                                + ") " + mylocation + " ---");
        }else if(packet.NameProtocol.equals(ReservoirElGamalProtocol.NameProtocol)){
                // Start Phase 2
                if (((ReservoirElGamalProtocol) packet.obj)
                                .iAmInGap(mylocation)) {
                        String hash = ((ReservoirElGamalProtocol) packet.obj)
                                        .getHashOfMyDistance(mylocation);
                        lotl.addMyHashOfDistance(packet.HashRequest, hash);
                        lotl.saveToFile(agent.entity.getEntityDirectory());
                        agent.logLine("--- Add NDP My Exact Distance With Hash: "
                                        + hash + " ---");
                } else {
                        agent.logLine("--- Not Add NDP Distance ---");
                }
                ((ReservoirElGamalProtocol) packet.obj)
                                .addExactDistanceIfYouAreInGap(mylocation);
        } else if (packet.NameProtocol.equals(BroadCastProtocol.NameProtocol)) {
                // Start Phase 3
                String myhash = lotl.FindMyHashOfDistance(packet.HashRequest);
                String reqhash = ((BroadCastProtocol) packet.obj)
                                .getHashOfDistance();
                if (myhash != null) {
                if (myhash.equals(reqhash)) {
                NDPPacket responcepacket=((BroadCastProtocol)packet.obj).ResponcePacket;
                responcepacket.obj = new DirectAnswerToBroadCast(
                                agent.getEntityName(),
                                new Host_Port(
                                agent.agentData.listenerExternalAddress.host
                                        + ":"
                                        + agent.agentData.listenerPort),
                                ((BroadCastProtocol) packet.obj).requestedLocation
```

```
                                        .getRadialDistance_HaversineFormula(lotl
                                        .FindMyLocation(packet.HashRequest)));
                                Host_Port hp = responcepacket.routingtable
                                                .getNextNode();
                                try {
                                        PolisConnection newcon = connectToPeer(
                                                        hp.getHost(), hp.getPort());
                                        newcon.objectOutputStream
                                                        .writeObject(responcepacket);
                                        newcon.objectOutputStream.flush();
                                } catch (Exception e) {
                                        System.err.println("Doesn't exist this Host:"
                                                + hp.getHost() + "\nError: " + e);
                                }
                                agent.logLine("--- I am the winner of NDP ---");
                        } else {
                                agent.logLine("--- I am a loser of NDP ---");
                        }
                } else {
                        agent.logLine("--- I am a loser of NDP ---");
                }
        }
        this.send();
        return null;
}
```