Πανεπιστήμιο Πειραιώς - Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Πρωτόκολλα Ανώνυμης Επικοινωνίας: Μελέτη περίπτωσης-Το πρωτόκολλο BAR και επεκτάσεις του πρωτοκόλλου Anonymous Communication Protocols: A case study of the BAR protocol and some extensions
Ονοματεπώνυμο Φοιτητή	Μανδαλενάκη Μαρία-Ευφραιμία
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΣΠ 14052
Επιβλέπων	Κοτζανικολάου Παναγιώτης, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Μάρτιος 2016**

Τριμελής Εξεταστική Επιτροπή

Κοτζανικολάου Παναγιώτης	Ψαράκης Μιχαήλ	Πατσάκης Κωνσταντίνος
Επίκουρος Καθηγητής	Επίκουρος Καθηγητής	Λέκτορας

*Man is **least** himself*
when he talks in his own person.

*Give him a **mask**,*
and he will tell you the truth.

Oscar Wilde

Περιεχόμενα

Κατάλογος εικόνων

Ευχαριστίες

Στο πλαίσιο της παρούσας μεταπτυχιακής διατριβής ήρθα σε επαφή με όρους και τεχνολογίες νέους προς εμένα, αλλά κυρίως με ένα σύνολο από ανθρώπους τους οποίους θα ήθελα να ευχαριστήσω για την τεχνική και την ηθική υποστήριξη που μου παρείχαν σε όλη τη διάρκεια αυτής της προσπάθειας.

Αρχικά, θα ήθελα να ευχαριστήσω τον κο. Παναγιώτη Κοτζανικολάου, επίκουρο καθηγητή του τμήματος και επιβλέπων καθηγητή της διατριβής μου, για την εμπιστοσύνη που μου έδειξε δίνοντάς μου τη δυνατότητα να ασχοληθώ με την επέκταση ενός θέματος το οποίο έχει ήδη αποσπάσει πολλές και αξιόλογες κριτικές, καθώς και για την πολύτιμη βοήθεια καθ' όλη τη διάρκεια της δημιουργίας και της συγγραφής.

Στην προσπάθεια αυτή συνέβαλε επίσης ο Γιώργος Χατζησωφρονίου, ο οποίος αποτελούσε τον αρχικό εμπνευστή και δημιουργό της ιδέας του πρωτοκόλλου και τον οποίο θα ήθελα να ευχαριστήσω θερμά για την υποστήριξη που μου παρείχε με τις χρήσιμες υποδείξεις και τα σχόλιά του.

Στη συνέχεια, θα ήθελα να προσθέσω πως η ολοκλήρωση της παρούσας διπλωματικής εργασίας χρηματοδοτήθηκε από το Ι.Κ.Υ. στο πλαίσιο του προγράμματος χορήγησης υποτροφιών για μεταπτυχιακές σπουδές πρώτου κύκλου (μάστερ) στην Ελλάδα με ένταξη στην αγορά εργασίας, ακαδ. έτους 2014-2015. Επομένως, θα ήθελα να ευχαριστήσω θερμά το Ι.Κ.Υ. για την υποτροφία που μου έδωσαν, καθώς και για την ευκαιρία να εργαστώ στα πλαίσια πρακτικής άσκησης, σε έναν από τους πλέον ισχυρούς και αξιόλογους οργανισμούς της Ελλάδας, την Εθνική Τράπεζα.

Τέλος, η εργασία αυτή δε θα μπορούσε να ολοκληρωθεί δίχως την αμέριστη συμπαράσταση και στήριξη της οικογένειας, των συναδέλφων και των φίλων μου, καθώς και ενός ανθρώπου που τα τελευταία χρόνια με υποστηρίζει σε κάθε βήμα της ζωής μου.

Μανδαλενάκη Μαρία

Επιτελική Σύνοψη

Το τελευταίο διάστημα το θέμα της ανωνυμίας στο Διαδίκτυο έχει επανέλθει δυναμικά, καθώς οι τεχνολογίες που αναπτύσσονται έχουν ενισχύσει τις φοβίες των χρηστών σχετικά με τη διασφάλιση της ταυτότητάς τους. Στη διεθνή βιβλιογραφία έχουν προταθεί διάφορα πρωτόκολλα τα οποία υποστηρίζουν διάφορες ιδιότητες της ανωνυμίας, κατά τη χρήση υπηρεσιών διαδικτύου. Αντικείμενο της παρούσας μεταπτυχιακής διατριβής είναι η ανάλυση των υφιστάμενων πρωτοκόλλων ανωνυμίας, καθώς επίσης και η μελέτη της περίπτωσης του πρωτοκόλλου BAR (Broadcast Anonymous Routing). Το πρωτόκολλο BAR παρέχει ισχυρή ανωνυμία τόσο για τον αποστολέα όσο και για τον παραλήπτη του μηνύματος, ενώ ταυτόχρονα παρέχει και μη-συνδεσιμότητα. Επιπλέον, καθιστά πολύ δύσκολο, ακόμα και για έναν καθολικό παρατηρητή, να επιβεβαιώσει ακόμα και αν ένας ή περισσότεροι χρήστες πράγματι ανταλλάσσουν μηνύματα σε μία συγκεκριμένη περίοδο. Ειδικότερα, ο κύριος στόχος αυτής της εργασίας ήταν η υλοποίηση και δοκιμή των υπο-πρωτοκόλλων που περιλαμβάνονται στην αρχιτεκτονική του BAR, καθώς και η επέκταση της ήδη υπάρχουσας υλοποίησης με σκοπό την ομαλή λειτουργία του συστήματος σε μεγάλο αριθμό χρηστών. Επίσης, δευτερεύων στόχος της παρούσας διπλωματικής ήταν η σχεδίαση μίας εφαρμογής, η οποία θα εξυπηρετούσε τις ανάγκες του πρωτοκόλλου και θα παρείχε ένα γραφικό περιβάλλον για τη διευκόλυνση των χρηστών. Σημαντικό γνώρισμα για μία τέτοια εφαρμογή διαδραματίζει η ασφάλεια των δεδομένων. Έτσι, έχει γίνει προσπάθεια να διασφαλιστεί η μεταδιδόμενη πληροφορία και τα στοιχεία του χρήστη με σύγχρονες τεχνικές κωδικοποίησης, καθώς είναι αναγκαίο να αποτρέπεται η πρόσβαση από οποιονδήποτε.

Λέξεις κλειδιά: BAR, Ανωνυμία αποστολέα, Ανωνυμία παραλήπτη, Μη συνδεσιμότητα, DC-net, Mix-net, Πολυεκπομπή, Επικοινωνία πραγματικού χρόνου.

Abstract

Nowadays, Internet anonymity is an issue that has gained the attention of Internet users, since modern web technologies have strengthened users' fears about the protection of their online identity. Several anonymous communication protocols have been proposed in the literature, which support various anonymity properties of online Internet users. The goal of this thesis is to analyze existing anonymous communication protocols. As a case study, the Broadcast Anonymous Routing protocol (BAR) is analyzed and several extensions are proposed. BAR is an anonymous communication protocol, a system that provides strong sender-anonymity, receiver-anonymity, as well as unlinkability. Furthermore, BAR does not allow a strong adversary, such as a global passive adversary, to verify whether some users are actually communicating at some time period. In particular, the main objective of this work was the development of all the sub-protocols defined in BAR, and the extension of its existing implementation, in order to provide a robust functionality of the system which scales well as the number of users increases. Moreover, a secondary objective of this thesis was the design of an application, which would serve the needs of the protocol and provide a graphical interface for the convenience of users. A key issue for such an application is data security. Thus, efforts have been made to ensure the data transmitted and user's information, with the use of modern encoding techniques, which is a necessity to prevent unauthorized access.

Keywords: BAR, Sender anonymity, Receiver anonymity, Unlinkability, DC-net, Mix-net, Broadcasting, Real-time communications.

Κεφάλαιο 1

Εισαγωγή

Η ταχεία ανάπτυξη των εφαρμογών του Διαδικτύου έχει κάνει την ανωνυμία της επικοινωνίας ολοένα και πιο σημαντική απαίτηση ασφάλειας. Παρόλο που η κρυπτογράφηση μπορεί να προστατεύσει τα δεδομένα της επικοινωνίας από κάποιον κακόβουλο χρήστη, δεν μπορεί να κρύψει αρκετές σημαντικές πληροφορίες, όπως για παράδειγμα το γεγονός ότι δύο συγκεκριμένοι χρήστες επικοινωνούν. Συνεπώς, ο επιτιθέμενος μπορεί να λάβει σημαντικά στοιχεία σχετικά όπως την κίνηση στο δίκτυο, τις πραγματικές ταυτότητες των οντοτήτων, το δίκτυο του αποστολέα και του παραλήπτη ή τις διευθύνσεις δικτύου τους, δημιουργώντας έτσι ένα πλήθος από σοβαρές συνέπειες.

Ας πάρουμε για παράδειγμα ένα στρατιωτικό δίκτυο επικοινωνίας [35]. Μία απότομη μεταβολή του τρόπου διεξαγωγής της κυκλοφορίας μπορεί να υποδεικνύει κάποιες επικείμενες δραστηριότητες. Κάτι τέτοιο είναι εξαιρετικά επικίνδυνο, καθώς οι αντίπαλοι μπορούν εύκολα να εντοπίσουν κρίσιμους κόμβους του δικτύου και στη συνέχεια, να ξεκινήσουν στοχευμένες επιθέσεις εναντίον τους. Το γεγονός αυτό καθιστά την ιδιωτικότητα της πηγής μία βασική απαίτηση ασφάλειας για την κυβέρνηση και τις στρατιωτικές επικοινωνίες.

Επιπλέον, υπάρχουν άνθρωποι που αναζητούν ευαίσθητες πληροφορίες και επιθυμούν να διατηρήσουν την ανωνυμία τους, προκειμένου να αποφευχθεί ο στιγματισμός και η φυσική ή κοινωνική ζημιά τους. Το συγκεκριμένο πρόβλημα έχει δημιουργήσει και την ανάγκη για ελευθερία της ανταλλαγής πληροφοριών. Κάποιες οργανώσεις μπορεί να θεωρούν την έκθεσή τους μέσα από συζητήσεις ή διαδόσεις, ενοχλητική ή ακόμα και επιβλαβή. Έτσι, θα προσπαθήσουν να καταστείλουν την ανταλλαγή των ανεπιθύμητων πληροφοριών.

Η έρευνα σχετικά με τις ανώνυμες επικοινωνίες ξεκίνησε το 1981 από τον Chaum [3], ενώ από τότε έχει επεκταθεί σε πολλούς τομείς. Παρακάτω, θα παρουσιαστούν οι βασικοί ορισμοί των ανώνυμων επικοινωνιών, όπως περιγράφηκαν από τους Pfitzmann και Hansen [1]:

- *Ανωνυμία (Anonymity)*
  Η ανωνυμία ενός ατόμου ορίζεται ως η κατάσταση στην οποία το άτομο δεν είναι αναγνωρίσιμο μέσα σε ένα σύνολο χρηστών, το οποίο ονομάζεται σύνολο ανωνυμίας (anonymity set). Για να ενεργοποιηθεί η ανωνυμία ενός ατόμου, θα πρέπει το σύνολο να έχει κοινά χαρακτηριστικά. Το σύνολο ανωνυμίας είναι η συλλογή όλων των πιθανών χρηστών. Όσον αφορά τις ενεργές οντότητες, όπως οι αποστολείς, το σύνολο της ανωνυμίας αποτελείται από άτομα τα οποία θα μπορούσαν να προκαλέσουν μία ενέργεια. Αντίστοιχα, για τις παθητικές οντότητες, όπως οι παραλήπτες, το σύνολο της ανωνυμίας αποτελείται από χρήστες που θα μπορούσαν να ενεργοποιηθούν.
  Με τον τρόπο αυτό, τόσο ο αποστολέας όσο και ο παραλήπτης μπορούν να είναι ανώνυμοι μόνο με τα αντίστοιχα σύνολα ανωνυμίας τους. Τα σετ του αποστολέα και του παραλήπτη μπορεί να είναι ασυνεχή, ίδια ή να επικαλύπτονται.

- *Μη συνδεσιμότητα (Unlinkability)*
  Η μη συνδεσιμότητα εξασφαλίζει ότι ο χρήστης μπορεί να κάνει πολλαπλές χρήσεις των πόρων ή υπηρεσιών χωρίς κάποιος τρίτος να είναι σε θέση να συνδέσει αυτές τις χρήσεις μαζί. Η μη συνδεσιμότητα απαιτείται για την προστασία της ταυτότητας του χρήστη κατά της διερεύνησης των εργασιών του.
  Η ανωνυμία μπορεί να οριστεί ως μη συνδεσιμότητα ενός στοιχείου ενδιαφέροντος και ενός αντικειμένου. Με άλλα λόγια, η ανωνυμία ενός χρήστη σημαίνει ότι δεν μπορεί να συσχετιστεί με κάποιο μήνυμα σε ένα δίκτυο επικοινωνίας και ένα συγκεκριμένο μήνυμα δεν μπορεί να συσχετιστεί με κάποιον αποστολέα.
  Η μη συνδεσιμότητα διαφέρει από την ψευδωνυμία από το γεγονός ότι, αν και στην ψευδωνυμία η ταυτότητα του χρήστη δεν είναι γνωστή, μπορεί να συσχετιστεί με κάποιες δράσεις.

- *Μη παρατηρησιμότητα (Unobservability)*
  Με τον όρο μη παρατηρησιμότητα ορίζεται η κατάσταση ενός αντικειμένου ενδιαφέροντος να είναι όμοιο με τα υπόλοιπα. Αυτό σημαίνει ότι τα πραγματικά μηνύματα που ανταλλάσσονται δεν είναι ευδιάκριτα από τα ψευδή. Παρόμοια με τα σύνολο ανωνυμίας υπάρχουν και τα σύνολα μη

παρατηρησιμότητας (unobservability set). Ομοίως, η μη παρατηρησιμότητα του αποστολέα σημαίνει ότι δεν είναι διακριτό αν κάποιος αποστολέας στέλνει, μέσα σε ένα σύνολο. Αντίστοιχα, μη παρατηρησιμότητα του παραλήπτη σημαίνει ότι σε ένα unobservability set δεν είναι αισθητό αν κάποιος χρήστης λαμβάνει. Επίσης, υπάρχει και η μη παρατηρησιμότητα σχέσης, η οποία σημαίνει ότι δεν είναι διακριτό αν στέλνεται κάτι από ένα σύνολο αποστολέων σε ένα σύνολο παραληπτών.

Στη συνέχεια, θα γίνει μία εισαγωγή στους ορισμούς της ανωνυμίας και της κρυπτογραφίας, καθώς και των βασικών συστημάτων που τις αποτελούν. Επίσης, θα αποσαφηνιστεί ο σκοπός της παρούσας διατριβής και η δομή που ακολουθήθηκε.

1.1 Ανωνυμία στο Διαδίκτυο

Στις μέρες μας, οι περισσότεροι χρήστες του Διαδικτύου συμμετέχουν σε διάφορες συζητήσεις και επικοινωνούν μεταξύ τους χρησιμοποιώντας ψευδώνυμα, τα οποία πολλές φορές δημιουργούν μία δική τους ταυτότητα και διαχωρίζονται από τον πραγματικό συγγραφέα. Σύμφωνα με το Πανεπιστήμιο της Στοκχόλμης [56], η συγκεκριμένη μέθοδος δημιουργεί περισσότερη ελευθερία της έκφρασης και συνεισφέρει στην αποποίηση της ευθύνης του λόγου.

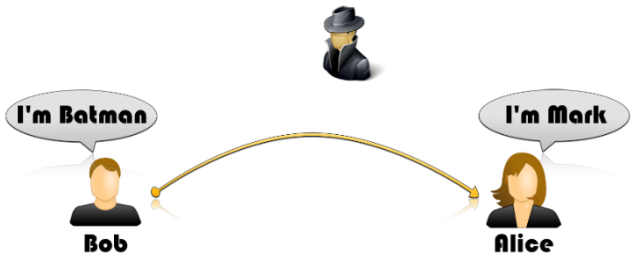
Ωστόσο, ο αρχικός σχεδιασμός του Διαδικτύου δεν εξυπηρετούσε την ανωνυμία. Η IP διεύθυνση του κάθε χρήστη χρησιμεύει ως εικονική διεύθυνση αλληλογραφίας και η πρόσβαση σε κάθε πηγή του Διαδικτύου γίνεται πάντα από τη συγκεκριμένη. Αυτή η διεύθυνση αντιστοιχεί σε κάποιον πάροχο υπηρεσίας Διαδικτύου (Internet Service Provider - ISP), ο οποίος με τη σειρά του μπορεί να παρέχει πληροφορίες σχετικά με τα στοιχεία του πελάτη που έχει μισθώσει τη συγκεκριμένη IP.

Για το λόγο αυτό, έχουν αναπτυχθεί κάποια δίκτυα ανωνυμοποίησης, όπως το I2P [57] και το Tor [15], τα οποία αντιμετωπίζουν το ζήτημα της παρακολούθησης της IP διεύθυνσης. Στα συγκεκριμένα δίκτυα, τα πακέτα, πριν αποσταλούν, κρυπτογραφούνται σε πολλαπλά επίπεδα και ακολουθούν ένα προκαθορισμένο δρομολόγιο μέσα στο ανώνυμο δίκτυο. Έτσι, κάθε δρομολογητής γνωρίζει τον προηγούμενό του ως προέλευση του πακέτου και τον αμέσως επόμενο ως προορισμό.

1.2.1 Τύποι Ανωνυμίας

Με βάση τον ορισμό της, ανωνυμία σημαίνει ότι ο πραγματικός συγγραφέας ενός μηνύματος δεν εμφανίζεται. Επομένως, ανάλογα με τον τρόπο που εφαρμόζεται μπορεί να διαχωριστεί σε κάποιες κατηγορίες, οι οποίες προσδιορίζουν και τον βαθμό δυσκολίας αποκάλυψης της πραγματικής ταυτότητας του συντάκτη ενός μηνύματος.

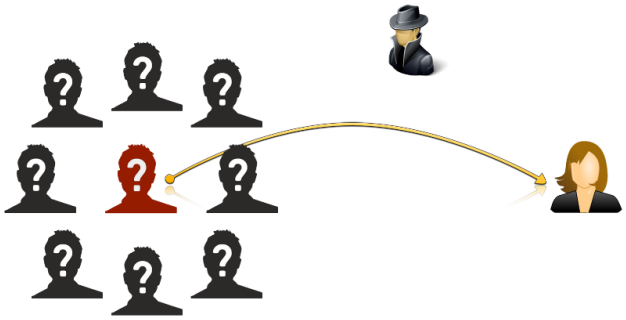
- *Ψευδωνυμία (Pseudonymity)*
  Το ψευδώνυμο είναι ένα εικονικό αναγνωριστικό, το οποίο δεν είναι άμεσα συνδεδεμένο με μία οντότητα. Στην περίπτωση των ψευδωνύμων, είναι πιθανή η σύνδεση και ο εντοπισμός της πραγματικής οντότητας από κάποιον τρίτο. Έτσι, αν αποκαλυφθεί το φυσικό πρόσωπο που κρύβεται πίσω από κάποιο συγκεκριμένο ψευδώνυμο, έχει χαθεί η ανωνυμία του και βρίσκεται εκτεθειμένο.



Εικόνα 1.1 Ψευδωνυμία (Pseudonymity)

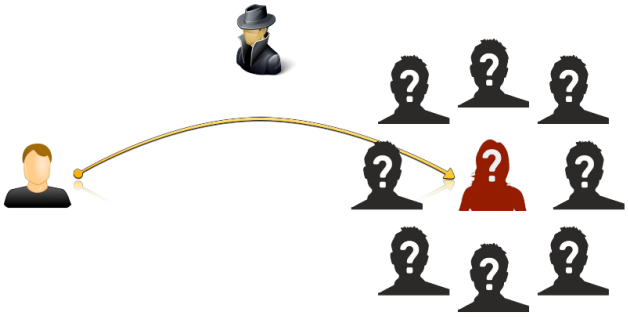
- *Ανωνυμία Αποστολέα (Sender Anonymity)*
  Στην περίπτωση που ο αποστολέας χρησιμοποιεί κάποιο πρόγραμμα που διασφαλίζει την ανωνυμία του, τόσο ο ίδιος ο παραλήπτης όσο και ένας τρίτος παρατηρητής, δεν μπορούν να

αναγνωρίσουν την ταυτότητα του αποστολέα. Το μόνο που παραμένει γνωστό είναι με ποιους χρήστες επικοινώνησε και η περιήγησή του στη συγκεκριμένη συνεδρία.



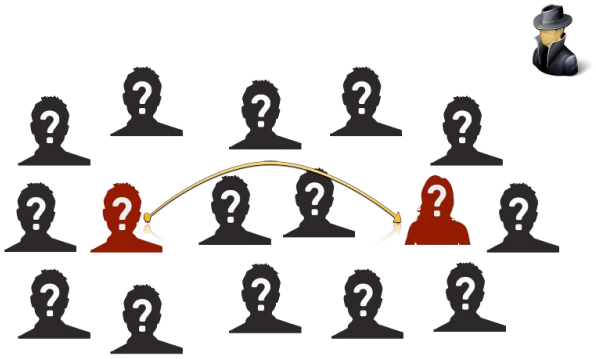
Εικόνα 1.2 Ανωνυμία αποστολέα (Sender anonymity)

- *Ανωνυμία Παραλήπτη (Receiver Anonymity)*
  Και σε αυτή την περίπτωση, αντίστοιχα, ο παραλήπτης χρησιμοποιεί κάποιο πρόγραμμα που θωρακίζει την ανωνυμία του. Έτσι, ο αποστολέας δεν μπορεί να αναγνωρίσει την ταυτότητα του χρήστη, με τον οποίο ήρθε σε επικοινωνία.



Εικόνα 1.3 Ανωνυμία παραλήπτη (Receiver anonymity)

- *Ανωνυμία Αποστολέα-Παραλήπτη (Sender-Receiver Anonymity)*
  Στη συγκεκριμένη περίπτωση, και οι δύο χρήστες που επικοινωνούν, χρησιμοποιούν κάποιο πρόγραμμα για να κατοχυρώσουν την ανωνυμία τους. Έτσι, ένας παρατηρητής δεν μπορεί να γνωρίζει τις πραγματικές τους ταυτότητες, αλλά και το ότι επικοινωνούν. Ο συγκεκριμένος τύπος αποτελεί την πιο δυνατή μορφή ανωνυμίας.



Εικόνα 1.4 Μη συνδεσιμότητα (Sender-Receiver Anonymity or Unlinkability)

1.2.2 Χρήση της Ανωνυμίας

Σε γενικές γραμμές, η ανωνυμία μπορεί να χρησιμοποιηθεί για καλούς, αλλά και κακούς σκοπούς. Επίσης, πολλές φορές μπορεί να είναι επιθυμητή από ένα πρόσωπο και δυσάρεστη για κάποιο άλλο. Για παράδειγμα, μία εταιρία, η οποία χρησιμοποιεί αθέμιτες πρακτικές για την κερδοφορία της, δεν θα επιθυμούσε κάποιος υπάλληλός της να εκθέσει πληροφορίες που θα τη ζημίωναν. Αντίθετα, η κοινωνία μπορεί να έβρισκε σημαντική την αποκάλυψη τέτοιων δεδομένων.

Θετική χρήση της ανωνυμίας μπορεί να θεωρηθεί:

- Σοβαρές καταχρήσεις από οργανισμούς ή πρόσωπα μπορούν να αποκαλυφθούν από ανώνυμες πηγές στα μέσα ενημέρωσης, καθώς και στα αρμόδια σώματα ασφαλείας.
- Οι πολίτες χωρών με καταπιεστικά πολιτικά καθεστώτα μπορούν να χρησιμοποιήσουν την ανωνυμία προκειμένου να εκφράσουν τις πολιτικές τους απόψεις, αποφεύγοντας όμως την δίωξη.
- Οι άνθρωποι, κατοχυρώνοντας την ανωνυμία τους, μπορούν να συζητήσουν ανοιχτά για ζητήματα, τα οποία δειλιάζουν να αναφέρουν σε προσωπικές συζητήσεις. Έρευνες έχουν δείξει ότι οι ανώνυμοι χρήστες αποκαλύπτουν σημαντικά περισσότερες πληροφορίες για τον εαυτό τους.
- Υπάρχει μεγαλύτερη ισότητα στις ανώνυμες συζητήσεις, καθώς κριτήρια όπως το φύλο ή η οικογενειακή κατάσταση δεν επηρεάζουν την αξιολόγησή τους.
- Η ανωνυμία παρέχει θάρρος σε κάποιους ανθρώπους να δημιουργήσουν επαφές που μπορεί να έχουν αξία για τους ίδιους, αλλά και άλλους.

Πέρα από τις θετικές επιδράσεις της ανωνυμίας, υπάρχει και μία σκοτεινή πλευρά της.

- Η ανωνυμία μπορεί να χρησιμοποιηθεί για παράνομη δραστηριότητα και συγκάλυψη εγκλημάτιών και των αδικημάτων τους, όπως η συκοφαντία, η διανομή υλικού παιδικής πορνογραφίας και η πρόκληση βλάβης από πρόθεση (ιοί υπολογιστών).
- Μέσα από ανώνυμες περιηγήσεις, μπορούν να αναζητηθούν επαφές για την εκτέλεση παράνομων πράξεων.
- Ακόμα και όταν η πράξη δεν είναι παράνομη, η ανωνυμία μπορεί να χρησιμοποιηθεί για την προσβολή κάποιου προσώπου.

1.2.3 Τύποι επιτιθέμενων και επιθέσεων

Στο πλαίσιο των πρωτοκόλλων ανώνυμης επικοινωνίας, ο πρωταρχικός στόχος ενός επιτιθέμενου είναι να δημιουργήσει μία αξιόπιστη αντιστοιχία μεταξύ της αποστολής ή λήψης ενός μηνύματος με μία οντότητα. Ένας επιπλέον στόχος είναι να διαταράξει το σύστημα επικοινωνίας, καθιστώντας το έτσι αναξιόπιστο. Παρακάτω, θα γίνει μία κατηγοριοποίηση των επιτιθέμενων με βάση τις ικανότητές τους, το ρόλο τους στο σύστημα, αλλά και τη δύναμή τους [37].

Ως προς τις ικανότητές του, ένα επιτιθέμενος μπορεί να θεωρηθεί ως:

- *Ενεργός αντίπαλος (Active adversary)*
  Μπορεί να εισάγει, να διαγράψει ή να τροποποιήσει τα μηνύματα στο δίκτυο.
- *Παθητικός αντίπαλος (Passive adversary)*
  Παρατηρεί τα δεδομένα που διέρχονται στις συνδέσεις επικοινωνίας.

Ως προς την πρόσβασή του, υπάρχουν δύο τύποι επιτιθέμενου:

- *Αντίπαλος με μερική/τοπική πρόσβαση (Local adversary)*
  Έχει πρόσβαση μόνο σε ένα μέρος του δικτύου.
- *Αντίπαλος με καθολική πρόσβαση (Global adversary)*
  Μπορεί να έχει πρόσβαση στο σύνολο του δικτύου και όλων των μηνυμάτων που ανταλλάχτηκαν.

Ως προς τον ρόλο του, ένας επιτιθέμενος μπορεί να κατηγοριοποιηθεί σε:

- *Εσωτερικός (Internal)*
  Αποτελεί χρήστη του συστήματος και μπορεί να ελέγξει τους κόμβους του δικτύου.
- *Εξωτερικός (External)*

Δεν αποτελεί χρήστη του συστήματος και μπορεί να θέσει σε κίνδυνο μόνο τα κανάλια επικοινωνίας μεταξύ των κόμβων.

Ως προς τη δύναμη και το πλήθος των υπολογιστικών πόρων που διαθέτει, ένας επιτιθέμενος μπορεί να χαρακτηριστεί ως:

- *Αδύναμος αντίπαλος (Weak adversary)*
  Ο συγκεκριμένος τύπος επιτιθέμενου μπορεί να κρυφακούσει στο πρώτο και το τελευταίο βήμα (hop) της επικοινωνίας, καθώς και να τροποποιήσει τα μηνύματα μόνο σε αυτά τις χρονικές στιγμές.
- *Ισχυρός αντίπαλος (Powerful adversary)*
  Ένας ισχυρός επιτιθέμενος μπορεί να παρακολουθεί όλους τους συνδέσμους της επικοινωνίας και να τροποποιεί τις παραμέτρους σε οποιοδήποτε βήμα.
- *Πολλοί επιτιθέμενοι συγκεντρωμένοι σε ομάδα (Collaborating attackers)*
  Μία ισχυρή παρακολούθηση και επίθεση σε ένα σύστημα απαιτεί μεγάλο εύρος υπολογιστικής ισχύς και χρόνου. Για το λόγο ότι κάτι τέτοιο θεωρείται αρκετά δαπανηρό και πολύπλοκο για ένα άτομο, έχουν δημιουργηθεί ομάδες από χρήστες που στοχεύουν σε εισβολές σε συστήματα και κανάλια επικοινωνίας.

Πέρα από τις διαφορετικές κατηγορίες επιτιθέμενων, σε ένα σύστημα μπορούν να πραγματοποιηθούν και διάφοροι τύποι επιθέσεων με σκοπό την εξασθένηση της ανωνυμίας των χρηστών του.

- *Ενεργές επιθέσεις (Active attacks)*
  Μία ενεργή επίθεση πραγματοποιείται όταν ο εισβολέας του συστήματος μπορεί να εισάγει, τροποποιεί ή διαγράφει μηνύματα στο δίκτυο. Ένας συνδυασμός αυτών μπορεί να καθυστερήσει τα μηνύματα σε ένα ανώνυμο δίκτυο επικοινωνίας ή να το κατακλύσει από μηνύματα προκαλώντας έτσι “πλημμύρα”.
- *Παθητικές επιθέσεις (Passive attacks)*
  Στις συγκεκριμένες επιθέσεις, κάποιος κακόβουλος χρήστης παρακολουθεί τη μετάβαση των δεδομένων στα κανάλια επικοινωνίας χωρίς να διαδραματίζουν κάποιο ρόλο στο σύστημα. Οι παθητικές επιθέσεις, οι οποίες πραγματοποιούνται με σκοπό την παρακολούθηση όλων των συνδέσεων ενός δικτύου, ονομάζονται παγκόσμιες (Global Passive Attacks). Οι τέτοιου τύπου επιθέσεις είναι το κύριο μοντέλο απειλής των συστημάτων συνδυασμού (Mix networks).
- *Τοπικές επιθέσεις (Local attacks)*
  Ονομάζονται οι επιθέσεις που πραγματοποιούνται σε ένα μέρος του δικτύου.
- *Καθολικές επιθέσεις (Global attacks)*
  Έτσι χαρακτηρίζονται οι επιθέσεις που πραγματοποιούνται στο σύνολο του δικτύου.
- *Εσωτερικές επιθέσεις (Internal attacks)*
  Οι επιθέσεις που πραγματοποιούνται από κακόβουλους χρήστες του συστήματος με στόχο τον έλεγχο κάποιων κόμβων του δικτύου.
- *Εξωτερικές επιθέσεις (External attacks)*
  Οι επιθέσεις που πραγματοποιούνται από εξωτερικούς κακόβουλους χρήστες με στόχο την έκθεση των καναλιών επικοινωνίας μεταξύ των κόμβων του δικτύου.
- *Επιθέσεις ανατρέποντος κόμβου (Subverted node attacks)*
  Εκτός από την αλληλεπίδρασή του με τις συνδέσεις δικτύου, ένας επιτιθέμενος μπορεί να έχει τον έλεγχο ενός αριθμού κόμβων στο δίκτυο. Έτσι, όλες οι επικοινωνίες που πραγματοποιούνται στους συγκεκριμένους κόμβους είναι διαφανείς στον εισβολέα, ο οποίος έχει και τη δυνατότητα να τροποποιήσει τα μηνύματα που διέρχονται από αυτόν.
- *Επιθέσεις πλημμύρας (Flooding attacks)*
  Η πλημμύρα (flooding) είναι μία επίθεση άρνησης υπηρεσίας (Denial-of-Service – DoS) που έχει σχεδιαστεί για να διακόψει τη λειτουργία ενός δικτύου στέλνοντας μεγάλες ποσότητες δεδομένων, με σκοπό την αύξηση της κυκλοφορίας. Έτσι, κατά τη διάρκεια μίας επίθεσης πλημμύρας, το δίκτυο επιβαρύνεται σε τέτοιο βαθμό με ψευδή πακέτα σύνδεσης, που δεν μπορεί να επεξεργαστεί τα πραγματικά.

- *Επιθέσεις καταναγκασμού (Compulsion attacks)*
  Τα ανώνυμα συστήματα επικοινωνίας συχνά αναπτύσσονται σε περιβάλλοντα με μεγάλη ανισορροπία της εξουσίας. Έτσι, κάθε ένας από τους συμμετέχοντες σε ένα δίκτυο βρίσκονται ευάλωτοι σε επιθέσεις καταναγκασμού. Αυτού του τύπου οι επιθέσεις είναι συνήθως μεγάλου κόστους για όλα τα μέρη και δεν μπορούν να είναι μεγάλης διάρκειας ή πάρα πολλές. Ως επίθεση καταναγκασμού μπορεί να θεωρηθεί η εκτέλεση κάποιας εργασίας υπό την απειλή της βίας.

1.2.4 Επίπεδα Ανωνυμίας

Η ανωνυμία, με βάση την ισχύ της, μπορεί να διαχωριστεί σε κάποιες βαθμίδες, όπως παρουσιάζονται και στην παρακάτω εικόνα.

Εικόνα 1.5 Επίπεδα ανωνυμίας

- *Απόλυτη Προστασία (Absolute Privacy)*
  Στο ένα άκρο του φάσματος είναι η απόλυτη προστασία της ιδιωτικότητας του ατόμου. Με άλλα λόγια, ένας επιτιθέμενος δεν μπορεί να διακρίνει τις περιπτώσεις που ένας χρήστης πραγματικά επικοινωνεί στο σύστημα.
- *Υπεράνω Υποψίας (Beyond Suspicion)*
  Ο επιτιθέμενος μπορεί να αποδείξει την αποστολή μηνύματος στο σύστημα σε συγκεκριμένη χρονική στιγμή, αλλά δεν μπορεί να γνωρίζει τους χρήστες που επικοινωνούν.
- *Πιθανή Αθωότητα (Probable Innocence)*
  Ο χρήστης που ερευνάται από τον εισβολέα είναι πολύ πιθανό να βρίσκεται σε επικοινωνία στο σύστημα, αλλά υπάρχει ίση πιθανότητα να είναι κάποιοι άλλοι χρήστες.
- *Λιγότερη Πιθανή Αθωότητα (Possible Innocence)*
  Ο χρήστης που ερευνάται είναι πολύ πιο πιθανό να επικοινωνεί με άλλο χρήστη στο σύστημα σε μία συγκεκριμένη χρονική στιγμή, αλλά υπάρχει ακόμα μία μηδαμινή πιθανότητα να είναι κάποιος άλλος χρήστης.
- *Εκτεθειμένος Χρήστης (Exposed)*
  Στο συγκεκριμένο επίπεδο, ο εισβολέας μπορεί να αποδείξει ότι ένας χρήστης επικοινωνεί με άλλους σε ένα σύστημα, αλλά δεν μπορεί να είναι σίγουρος για την αληθινή του ταυτότητα.
- *Αποδεδειγμένα Εκτεθειμένος Χρήστης (Provably Exposed)*
  Στο άλλο άκρο του φάσματος είναι η αποδεδειγμένη έκθεση της ταυτότητας του χρήστη. Ο επιτιθέμενος μπορεί να εντοπίσει τα κανάλια επικοινωνίας του χρήστη σε ένα σύστημα, αλλά και να αποδείξει την ταυτότητά του σε άλλους.

1.2 Η Κρυπτογραφία στην υπηρεσία της ανωνυμίας

Προκειμένου να διασφαλιστεί η ανώνυμη επικοινωνία των χρηστών, όλα τα πρωτόκολλα που έχουν αναπτυχθεί κατά καιρούς χρησιμοποιούν κάποιες τεχνικές κρυπτογραφίας. Έτσι, στο συγκεκριμένο κεφάλαιο, θα γίνει μία περιγραφή της κρυπτογραφίας, θα ταξινομηθεί με βάση τα είδη της και θα παρουσιαστούν τα βασικά κρυπτογραφικά εργαλεία που χρησιμοποιούνται.

Η *κρυπτογραφία (cryptography)* είναι η μελέτη τεχνικών που βασίζονται σε μαθηματικά προβλήματα δύσκολο να λυθούν. Εφαρμογή της κρυπτογραφίας είναι η κρυπτογράφησης. *Κρυπτογράφηση (encryption)* είναι ο μετασχηματισμός δεδομένων σε μορφή που είναι αδύνατον να διαβαστεί χωρίς τη γνώση της

σωστής ακολουθίας bit, η οποία ονομάζεται κλειδί και χρησιμοποιείται σε συνδυασμό με κατάλληλο αλγόριθμο. Η αντίστροφη διαδικασία ονομάζεται *αποκρυπτογράφηση (decryption)* και απαιτεί γνώση του κλειδιού. Για μερικούς μηχανισμούς χρησιμοποιείται το ίδιο κλειδί για την κρυπτογράφηση και την αποκρυπτογράφηση, ενώ για άλλους τα κλειδιά διαφέρουν.

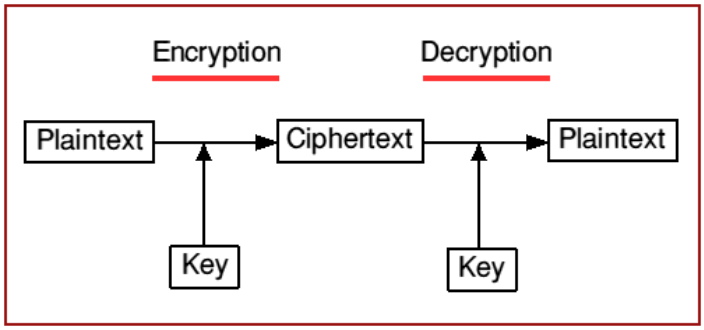
1.2.1 Είδη Κρυπτογραφίας

- *Ασύμμετρη Κρυπτογραφία (Public-Key Cryptography)*
  Η ασύμμετρη κρυπτογραφία χρησιμοποιεί δύο διαφορετικά κλειδιά για την κρυπτογράφηση και αποκρυπτογράφηση. Κάθε χρήστης έχει στην κατοχή του ένα ζεύγος κλειδιών, το ένα καλείται ως δημόσιο κλειδί (public key) και το άλλο ως ιδιωτικό (private key). Το δημόσιο κλειδί δημοσιοποιείται στους άλλους χρήστες, ενώ το ιδιωτικό κρατείται μυστικό.
  Η κρυπτογράφηση με τη χρήση της ασύμμετρης κρυπτογραφίας γίνεται ως εξής: όταν ένας χρήστης Α θέλει να στείλει ένα μήνυμα στον χρήστη Β χρησιμοποιεί το δημόσιο κλειδί του Β, το οποίο είναι κάπου δημοσιευμένο, για να κρυπτογραφήσει το μήνυμα και έπειτα να το στείλει. Ο χρήστης Β, με τη σειρά του, μόλις παραλάβει το κρυπτογραφημένο μήνυμα, χρησιμοποιεί το ιδιωτικό του κλειδί προκειμένου να το αποκρυπτογραφήσει. Κάποιος τρίτος που παρατηρεί την επικοινωνία δεν μπορεί να αποκρυπτογραφήσει το μήνυμα.
- *Συμμετρική Κρυπτογραφία (Secret-Key Cryptography)*
  Στο συγκεκριμένο είδος κρυπτογραφίας ο αποστολέας και ο παραλήπτης ενός μηνύματος χρησιμοποιούν ένα κοινό μυστικό κλειδί, το οποίο έχουν ανταλλάξει από πριν. Ο αποστολέας χρησιμοποιεί το μυστικό κλειδί για να κρυπτογραφήσει το μήνυμα και ο παραλήπτης χρησιμοποιεί το ίδιο κλειδί για να το αποκρυπτογραφήσει. Το κύριο πρόβλημα της συμμετρικής κρυπτογραφίας είναι η συνεννόηση του αποστολέα και του παραλήπτη στο κοινό μυστικό κλειδί.

1.2.2 Κρυπτογραφικά Εργαλεία

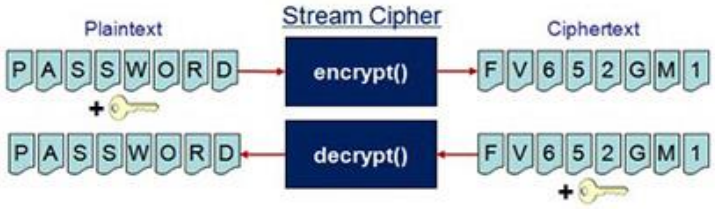
Μέχρι τώρα αναφερθήκαμε στις δύο βασικές κατηγορίες κρυπτοσυστημάτων που ευρέως εφαρμόζονται σήμερα. Περιγράψαμε τις αρχές που τα διέπουν και το είδος των κλειδιών που χρησιμοποιούν (συμμετρικά ή ασύμμετρα). Παρακάτω, θα γίνει μία σύντομη αναφορά στους μηχανισμούς με τους οποίους εφαρμόζεται η κρυπτογραφία γενικότερα.

- *Block Ciphers*
  Είναι ένας τύπος αλγόριθμου συμμετρικής κρυπτογράφησης που μετατρέπει ένα block μη κρυπτογραφημένου κειμένου (plaintext) σε block κρυπτογραφημένου μηνύματος ίδιου μήκους (cipherext). Ο συγκεκριμένος μετασχηματισμός πραγματοποιείται με τη χρήση ενός μυστικού κλειδιού που χορηγείται από τον χρήστη. Η αποκρυπτογράφηση γίνεται με την εφαρμογή του αντίστροφου μετασχηματισμού στο κρυπτογραφημένο κείμενο χρησιμοποιώντας το ίδιο μυστικό κλειδί.
  Οι πιο γνωστοί block cipher αλγόριθμοι είναι: Lucifer [38] / DES [39], 3-DES [40], RC5 [41], Rijndael / AES [42], Blowfish [43].



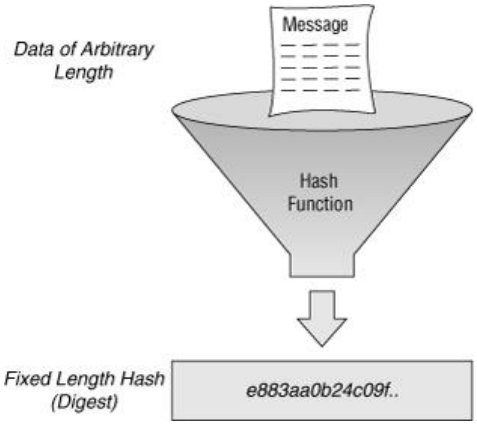
Εικόνα 1.6 Block Ciphers

- *Stream Ciphers*
  Αποτελεί έναν τύπο αλγόριθμου συμμετρικής κρυπτογράφησης, ο οποίος είναι κατά πολύ ταχύτερος από τους block ciphers. Σε αντίθεση με τους block ciphers που λειτουργούν με μεγάλα κομμάτια δεδομένων (blocks), οι stream ciphers τυπικά λειτουργούν με μικρότερες μονάδες απλού κειμένου. Η κρυπτογράφηση ενός συγκεκριμένου κειμένου με έναν block cipher θα καταλήγει πάντα στο ίδιο αποτέλεσμα όταν χρησιμοποιείται το ίδιο κλειδί. Με έναν stream cipher, ο μετασχηματισμός των μικρότερων αυτών μονάδων θα ποικίλει, ανάλογα με τη χρονική στιγμή που αντιμετωπίζονται κατά την διάρκεια της κρυπτογράφησης.
  Οι πιο γνωστοί stream cipher αλγόριθμοι είναι: RC4 [44], eStream portfolio [45].



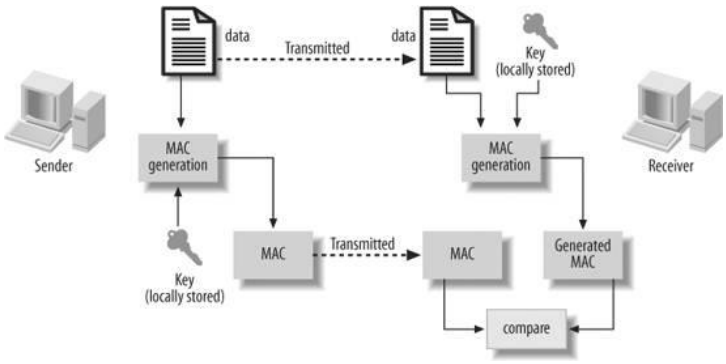
Εικόνα 1.7 Stream Ciphers

- *Hash Functions*
  Ο όρος hash function υποδηλώνει ένα μετασχηματισμό που παίρνει σαν είσοδο ένα μήνυμα οποιουδήποτε μήκους και επιστρέφει σαν έξοδο μία ακολουθία χαρακτήρων περιορισμένου μήκους. Οι hash functions έχει ως βασική ιδιότητα ότι είναι το κείμενο που παράγεται σαν έξοδο είναι μη αναστρέψιμο, δηλαδή γνωρίζοντας την έξοδο δεν μπορεί να βρεθεί το αρχικό μήνυμα εισόδου, και αμφιμονοσήμαντο (ένα προς ένα συνάρτηση).
  Οι πιο γνωστοί hash function αλγόριθμοι είναι: MD4 [46], MD5 [47], SHA-1 [48], SHA-256 [49], SHA-512 [49]



Εικόνα 1.8 Hash Functions

- *Message Authentication Code*
  Ως Message Authentication Code (MAC) [50] ορίζεται ένας κώδικας (checksum) που συνοδεύει το μήνυμα και πιστοποιεί την ταυτότητα του αποστολέα και την ακεραιότητα του μηνύματος. Για την παραγωγή του checksum χρησιμοποιείται κάποιο από τα παραπάνω κρυπτογραφικά κλειδιά σε συνδυασμό με ένα μυστικό κλειδί.



Εικόνα 1.9 Message Authentication Code

- *Μηχανισμοί Διαχείρισης και Ανταλλαγής Κλειδιών*
  Οι μηχανισμοί διαχείρισης κλειδιών (key management) και αυτοί της ανταλλαγής κλειδιών (key exchange) [51] ασχολούνται με την ασφαλή παραγωγή, διανομή και αποθήκευση των κλειδιών κρυπτογράφησης. Η εύρεση απρόσβλητων μεθόδων διαχείρισης και ανταλλαγής κλειδιών είναι πολύ σημαντική διαδικασία στη διατήρηση της ασφάλειας της επικοινωνίας.
  Η έννοια της διαχείρισης κλειδιών αναφέρεται στα ασύμμετρα κρυπτοσυστήματα. Οι χρήστες θα πρέπει να είναι σε θέση να μπορούν να αποκτήσουν με ασφάλεια ένα ζεύγος κλειδιών (ιδιωτικό-δημόσιο) για προστατευμένη επικοινωνία. Για τις μεν δημόσιες κλείδες, θα πρέπει να υπάρχει ένας τρόπος αποθήκευσης και δημοσιοποίησής τους, καθώς και να είναι δυνατή η ανάκτησή τους όποτε χρειάζεται. Επίσης, τα δημόσια κλειδιά θα πρέπει να σχετίζονται με την ταυτότητα του κατόχου, ώστε να αποφεύγονται οι περιπτώσεις πλαστοπροσωπίας. Τέλος, οι χρήστες θα πρέπει να έχουν τη δυνατότητα να φυλάσσουν τα ιδιωτικά τους κλειδιά με ασφάλεια.
  Οι μηχανισμοί της ανταλλαγής κλειδιών εφαρμόζονται κυρίως στα συμμετρικά κρυπτοσυστήματα, στα οποία οι δύο επικοινωνούντες χρήστες θα πρέπει να αποφασίσουν για το κοινό μυστικό κλειδί και να αποκτήσουν ένα αντίγραφο αυτού, χωρίς να το μάθει κάποιος τρίτος.

1.3 Κατηγορίες συστημάτων ανωνυμίας

Κατά καιρούς έχουν αναπτυχθεί διάφορα συστήματα που στοχεύουν στη διασφάλιση της ανωνυμίας στο Διαδίκτυο. Όλα όμως μπορούν να διαχωριστούν σε δύο κύριες κατηγορίες: τα dc-net και τα mix net. Και οι δύο μηχανισμοί στοχεύουν στην επίτευξη της ανωνυμίας του αποστολέα, του παραλήπτη, αλλά και της επικοινωνίας τους, από ισχυρούς επιτιθέμενους στο σύστημα.

1.3.1 DC networks

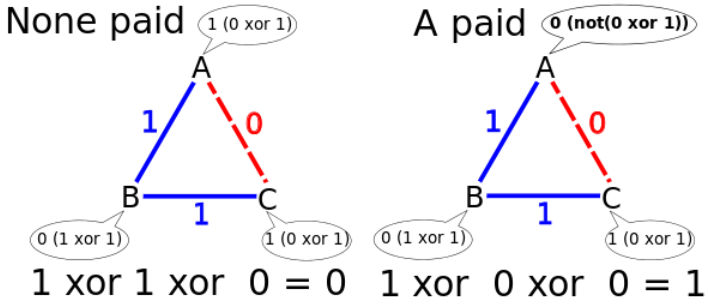
Το 1988, ο David Chaum [2] παρουσίασε ένα πρωτοποριακό πρωτόκολλο ανώνυμης επικοινωνίας. Για την περιγραφή του πρωτοκόλλου χρησιμοποίησε ως παράδειγμα το πρόβλημα των κρυπτογράφων, οι οποίοι δειπνούν και θέλουν να επικοινωνήσουν μεταξύ τους με πλήρη ανωνυμία (dining cryptographer problem). Η γενίκευση του συγκεκριμένου προβλήματος είναι εκείνη που αργότερα θα δημιουργήσει τα δίκτυα *Dining Cryptographer Networks (DC-net)*. Με βάση τον Chaum, για την περιγραφή του συγκεκριμένου προβλήματος υπάρχουν τρεις κρυπτογράφοι, οι οποίοι έχουν καθίσει να δειπνήσουν στο αγαπημένο τους εστιατόριο. Ο σερβιτόρος τους ενημερώνει ότι ο λογαριασμός έχει πληρωθεί ανώνυμα. Αυτό σημαίνει ότι ένας από τους κρυπτογράφους θα μπορούσε να έχει πληρώσει το δείπνο ή η Εθνική Υπηρεσία Ασφάλειας (NSA). Ο κάθε κρυπτογράφος σέβεται το δικαίωμα του άλλου να πληρώσει ανώνυμα, αλλά αναρωτιούνται αν έχει πληρώσει η NSA. Προκειμένου να επιλύσουν την αβεβαιότητά τους, διενεργούν το παρακάτω πρωτόκολλο:

Κάθε κρυπτογράφος ρίχνει ένα κέρμα ανάμεσα σε αυτόν και τον κρυπτογράφο που βρίσκεται στα δεξιά του, έτσι ώστε μόνο οι δυο τους να μπορούν να δουν το αποτέλεσμα. Στη συνέχεια, ο καθένας δηλώνει αν τα δύο κέρματα που έχει δει, αυτό που έριξε και του αριστερού του γείτονα, έχουν πέσει το ίδιο ή διαφορετικό αποτέλεσμα (και τα δύο κέρματα ίδιο αποτέλεσμα = 1, το ένα κορώνα και το άλλο γράμματα =0). Αν κάποιος

από τους κρυπτογράφους είναι ο πληρωτής, θα πρέπει να δηλώσει το αντίθετο από αυτό που βλέπει. Αν από τα αποτελέσματα που θα εμφανιστούν στο τραπέζι προκύπτει περιττός αριθμός, τότε έχει πληρώσει κάποιος κρυπτογράφος, αλλά δεν είναι γνωστό ποιος. Αν προκύψει άρτιος, αποδεικνύεται ότι έχει πληρώσει η NSA για το δείπνο.

Ο συγκεκριμένος αλγόριθμος μπορεί να λειτουργήσει για πεπερασμένο αριθμό συμμετεχόντων, με την προϋπόθεση ότι όλοι είναι ειλικρινείς ως προς τα κλειδιά και τα μηνύματά τους.

Παρόλο που τα δίκτυα DC-net παρέχουν πλήρη ανωνυμία εναντίον ισχυρών αντιπάλων, μόνο ένας χρήστης μπορεί να στείλει μήνυμα σε έναν χρόνο, καθιστώντας έτσι αυτά τα συστήματα απρόσιτα, ενώ είναι πιθανό για τους συμμετέχοντες να γνωρίζουν πότε ένα μήνυμα έχει αποσταλεί. Πολλά μεταγενέστερα πρωτόκολλα ανώνυμης επικοινωνίας βασίζονται ή επεκτείνουν το πρωτόκολλο dc-net, τα οποία και θα αναλυθούν στο επόμενο κεφάλαιο.



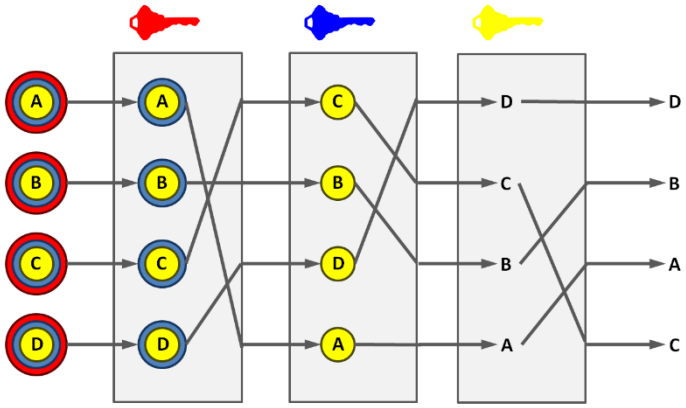
Εικόνα 1.10 DC-networks

1.3.2 Mix networks

Με τον όρο *mix networks (mix-nets)* προσδιορίζονται τα πρωτόκολλα δρομολόγησης, τα οποία δημιουργούν δυσκολία στον εντοπισμό χρησιμοποιώντας μία αλυσίδα *διακομιστών διαμεσολάβησης (proxy servers)*, γνωστών ως *mixes*. Οι proxy servers λαμβάνουν μηνύματα από πολλαπλούς αποστολείς, τα ανακατεύουν και τα στέλνουν πίσω στον επόμενο προορισμό (ενδεχομένως τον επόμενο κόμβο mix) σε τυχαία σειρά. Αυτό έχει ως αποτέλεσμα, τη διάσπαση της σύνδεσης της πηγής του αιτήματος και του προορισμού, γεγονός που καθιστά δύσκολο για τους επιτιθέμενους να προσδιορίσουν τους χρήστες που επικοινωνούν μία τυχαία χρονική στιγμή.

Ως προς την υλοποίηση των συγκεκριμένων δικτύων, κάθε μήνυμα κρυπτογραφείται σε κάθε κόμβο χρησιμοποιώντας την κρυπτογραφία δημόσιου κλειδιού. Το κρυπτογράφημα που προκύπτει περιέχει επίπεδα κρυπτογράφησης, με το αρχικό μήνυμα να βρίσκεται στο κατώτατο στρώμα. Έτσι, κάθε κόμβος αποκρυπτογραφεί με το ιδιωτικό του κλειδί το μήνυμα που λαμβάνει, προκειμένου να ανακαλύψει τον επόμενο κόμβο που θα πρέπει να παραλάβει το νέο μήνυμα.

Η ιδέα των mix-net περιγράφηκε για πρώτη φορά από τον David Chaum [3] το 1981. Στις εφαρμογές που έχουν αναπτυχθεί και στηρίζονται στον συγκεκριμένο μηχανισμό περιλαμβάνονται οι *ανώνυμοι remailers* και η *δρομολόγηση "κρεμμύδι" (onion routing)*.



Εικόνα 1.11 Mix-networks

1.4 Σκοπός της διπλωματικής

Ο σκοπός της συγκεκριμένης διατριβής είναι η παρουσίαση των πρωτοκόλλων που έχουν αναπτυχθεί κατά καιρούς με στόχο την ανώνυμη επικοινωνία, και η μελέτη της περίπτωσης του πρωτοκόλλου BAR. Επίσης, η παρούσα μεταπτυχιακή διατριβή στοχεύει στην υλοποίηση των οντοτήτων και των πρωτοκόλλων του συστήματος BAR που έχουν περιγραφεί αλλά δεν υπάρχουν στην αρχική υλοποίηση του πρωτοκόλλου [34], καθώς στην ανάπτυξη μίας εφαρμογής [54] σε προγραμματιστικό περιβάλλον που θα αποτελέσει μία γραφική απεικόνιση των δυνατοτήτων του.

Το πρωτόκολλο BAR έχει ως στόχο να διασφαλίσει την ανώνυμη επικοινωνία των χρηστών στο Διαδίκτυο. Σε αντίθεση με τις περισσότερες υλοποιήσεις που κυκλοφορούν σήμερα, το BAR διαφυλάττει και τη μη συνδεσιμότητα των χρηστών που επικοινωνούν. Συγκεκριμένα, ένας χρήστης που επιθυμεί να επικοινωνήσει με έναν άλλο χρήστη του πρωτοκόλλου BAR, εισέρχεται στην υπηρεσία, επιλέγει το χρήστη που θέλει να επικοινωνήσει, κρυπτογραφεί το μήνυμα με κλειδί που έχουν ανταλλάξει από πριν και το αποστέλλει στον εξυπηρέτη του πρωτοκόλλου BAR, ο οποίος αναλαμβάνει την αναμετάδοσή του σε όλους τους χρήστες. Επομένως, όλοι οι χρήστες παραλαμβάνουν το μήνυμα, όμως μόνο ένας είναι εκείνος που έχει το κλειδί για να το αποκρυπτογραφήσει.

Το πρωτόκολλο έχει υλοποιηθεί με κύριο γνώμονα τον ίδιο το χρήστη και την ανωνυμία του ως προς την περιήγησή του στο περιβάλλον BAR. Έτσι, κατά την εγγραφή του στην υπηρεσία, ο πελάτης ορίζει ένα ψευδώνυμο, το οποίο θα εμφανίζεται στους υπόλοιπους χρήστες και τις υπηρεσίες του BAR. Επίσης, προκειμένου να διασφαλιστεί το απόρρητο της επικοινωνίας του με άλλους χρήστες του πρωτοκόλλου, όλοι οι πελάτες στέλνουν στον διακομιστή BAR μηνύματα σταθερού μεγέθους σε ορισμένες χρονικές στιγμές, τα οποία μπορεί να είναι είτε κρυπτογραφημένα με κάποιο μήνυμα είτε θόρυβος (τυχαίες συμβολοσειρές).

1.5 Δομή της διπλωματικής

Στη συνέχεια, γίνεται μία αναφορά στη διάρθρωση του κειμένου που θα ακολουθηθεί στη μεταπτυχιακή διατριβή.

Το παρόν κεφάλαιο (**Κεφάλαιο 1)** αποτελεί μία εισαγωγή στον όρο της ανωνυμίας στο Διαδίκτυο, στους κινδύνους που υπάρχουν, καθώς και στα βασικά συστήματα και κρυπτογραφικά εργαλεία. Επίσης, γίνεται μία αναφορά στον σκοπό υλοποίησης της παρούσας διατριβής.

Το **Κεφάλαιο 2** αναλύει τις ήδη υπάρχουσες υλοποιήσεις που στοχεύουν στη διασφάλιση της ανώνυμης περιήγησης στο Διαδίκτυο, καθώς και τις διαφορές τους με το πρωτόκολλο που μελετάται στην παρούσα διατριβή.

Στο **Κεφάλαιο 3** γίνεται μία περιγραφή του πρωτοκόλλου BAR με βάση τις ήδη υπάρχουσα υλοποίησή του, με σκοπό την καλύτερη κατανόηση του τελικού αποτελέσματος.

Στο **Κεφάλαιο 4** αναφέρεται η αρχιτεκτονική της εφαρμογής και ο τρόπος σχεδιασμού των βάσεων δεδομένων που χρησιμοποιήθηκαν, για την καλύτερη κατανόηση του τελικού αποτελέσματος.

Στο **Κεφάλαιο 5** παρουσιάζεται η εφαρμογή που εκπονήθηκε στα πλαίσια της παρούσας διατριβής, η οποία ονομάζεται BarApp [54], πραγματοποιήθηκε με τη μορφή ιστότοπου και παρέχει τη δυνατότητα εισόδου και εγγραφής στο περιβάλλον BAR. Επιπλέον, περιγράφεται ο τρόπος υλοποίησής της και οι δυνατότητες που παρέχει στον χρήστη. Τέλος, περιέχονται τα αποσπάσματα κώδικα που επιβεβαιώνουν τη λειτουργικότητα της εφαρμογής.

Το **Κεφάλαιο 6** αποτελεί τη σύνοψη της μεταπτυχιακής διατριβής κα παρουσιάζονται τα συμπεράσματα και οι μελλοντικές εξελίξεις.

Κεφάλαιο 2

Επισκόπηση Βιβλιογραφίας

Στο προηγούμενο κεφάλαιο, αποσαφηνίστηκε ο ορισμός της ανωνυμίας και παρουσιάστηκαν οι τεχνικές που μπορούν να συνδυαστούν για να επιτευχθούν οι διάφοροι τύποι της.

Όπως αναφέρθηκε και παραπάνω, οι πιο γνωστές και ευρέως αποδεκτές αρχιτεκτονικές για ανώνυμη επικοινωνία στο Διαδίκτυο μπορούν να χωριστούν σε δύο κύριες κατηγορίες: εκείνες που βασίζονται στα δίκτυα mix-net και εκείνες που στηρίζονται στα δίκτυα DC-net. Σε αυτό το κεφάλαιο θα περιγραφούν οι ήδη υπάρχουσες υλοποιήσεις που στοχεύουν στη διασφάλιση της ανώνυμης περιήγησης στο Διαδίκτυο, θα καταγραφούν τα πλεονεκτήματα και μειονεκτήματα της κάθε μίας και θα αποσαφηνιστούν οι επιθέσεις κατά της ανωνυμίας που προλαμβάνονται σε κάθε περίπτωση.

2.1 DC-net συστήματα

Τα δίκτυα *DC-net* [2] επιτρέπουν στους χρήστες τους να στέλνουν και να λαμβάνουν μηνύματα ανώνυμα. Με την παραδοχή ενός αξιόπιστου δικτύου μετάδοσης, έχει αποδειχθεί ότι η ανωνυμία σε αυτά τα δίκτυα προασπίζεται σε μεγάλη κλίμακα. Επίσης, ένα από τα πλεονεκτήματα των DC-net δικτύων είναι ότι παρέχει ανωνυμία και στον αποστολέα, αλλά και στον παραλήπτη χωρίς να στηρίζεται σε κάποιο έμπιστο τρίτο μέρος. Ακόμα, η βασική αρχιτεκτονική DC-net έχει το πολύ ελκυστικό χαρακτηριστικό της μη διαδραστικότητας (non-interactivity). Αυτό σημαίνει ότι κάθε χρήστης του δικτύου μπορεί να δημοσιεύει το μήνυμά του σε έναν γύρο εκπομπής, χαρακτηριστικό που δεν παρουσιάζεται στα mix-net δίκτυα.

Όπως προκύπτει από τις βιβλιογραφικές αναφορές, τα mix-net δίκτυα έχουν χρησιμοποιηθεί ως βάση σε πολλά υλοποιημένα πρωτόκολλα, σε αντίθεση με τα DC-net που έμειναν παραμελημένα. Αυτό συμβαίνει διότι τα DC-net δίκτυα δεν μπορούν να λειτουργήσουν ως διαμεσολαβητές όπως τα mix-net.

Η κύρια ιδέα των DC-net δικτύων είναι ότι κάθε συμμετέχων ρίχνει ένα κέρμα και το αποτέλεσμά του κάθε φορά το ανταλλάσσει μυστικά με κάποιον άλλο χρήστη. Η ισοτιμία των αποτελεσμάτων που έχει κάθε χρήστης, στη συνέχεια μεταδίδεται σε όλη την ομάδα. Δεδομένου ότι κάθε αποτέλεσμα μεταδίδεται δύο φορές, το συνολικό αποτέλεσμα είναι ζυγός αριθμός. Αν ένας συμμετέχων θέλει να στείλει μήνυμα, αντιστρέφει σκόπιμα το αποτέλεσμα και το μεταδίδει. Έτσι, όταν η συνολική ισοτιμία είναι περιττός αριθμός, υποδεικνύεται η μετάδοση ενός bit. Με αυτό τον τρόπο, κανένας, εκτός από τον ίδιο τον εκκινητή, δεν μπορεί να γνωρίζει ποιος ξεκίνησε τη μετάδοση.

Η αδυναμία ανίχνευσης του αποστολέα ενός μηνύματος έχει αποδειχθεί ότι είναι απόλυτη. Αυτό σημαίνει ότι ο χρήστης που μεταδίδει κάποιο μήνυμα είναι δύσκολο να εντοπιστεί, ακόμα και αν ο επιτιθέμενος έχει απεριόριστους υπολογιστικούς πόρους και μπορεί να ελέγξει ένα υποσύνολο των χρηστών.

Ένα από τα σημαντικότερα προβλήματα που αντιμετωπίζουν τα DC-net δίκτυα είναι το πρόβλημα σύγκρουσης στη μετάδοση μέσω παρεμβολών από κακόβουλους χρήστες. Ένας αντίπαλος του συστήματος μπορεί να ξεκινήσει μία επίθεση *άρνησης της υπηρεσίας (Denial-of-Service)*, επιλέγοντας να στείλει ένα μήνυμα σε κάθε γύρο ή απλά να εγκαταλείψει το δίκτυο χωρίς να εντοπιστεί. Με αυτό τον τρόπο, οι ειλικρινείς χρήστες του συστήματος αποτρέπονται από την παράδοση των μηνυμάτων τους. Η επίθεση αυτή δεν μπορεί να εντοπιστεί αφού κάθε κόμβος είναι τόσο ανώνυμος όσο και ο αρχικός εκκινητής.

Ένας ακόμα περιορισμός που εμφανίζεται στην αρχιτεκτονική των DC-net δικτύων είναι η έλλειψη αξιόπιστων μηχανισμών που θα του επιτρέψουν να ελέγξει αν γίνεται έντιμη χρήση του. Με άλλα λόγια, ο κρυπτογράφος που ανακοινώνει τελευταίος το bit της ρίψης των κερμάτων, έχει τη δυνατότητα παραποίησης του τελικού αποτελέσματος, μπλοκάροντας έτσι τη λειτουργία του πρωτοκόλλου.

Τέλος, τα DC-net δίκτυα παρουσιάζουν το πρόβλημα της πολυπλοκότητας, καθώς για τη λειτουργία του πρωτοκόλλου απαιτούνται ζεύγη από κοινά μυστικά κλειδιά μεταξύ των συμμετεχόντων. Κάτι τέτοιο μπορεί να δημιουργήσει προβλήματα στην επικοινωνία στο δίκτυο, αν υπάρχουν πολλοί συμμετέχοντες.

Παρακάτω, θα παρουσιαστούν τα πρωτόκολλα που έχουν αναπτυχθεί κατά καιρούς και στηρίζονται στον μηχανισμό των DC-net συστημάτων.

2.1.1 DC⁺-net

Με βάση τον David Chaum [2], τον αρχικό δημιουργό των δικτύων DC-net, τα τελευταία εγγυώνται ανώνυμη επικοινωνία σε ένα δίκτυο επικοινωνίας. Όμως, για την ομαλή λειτουργία τους, υπάρχει η παραδοχή ότι το δίκτυο μετάδοσης θεωρείται αξιόπιστο, δηλαδή ότι κάθε συμμετέχων στην επικοινωνία είναι ειλικρινής και δεν τροποποιεί τα μηνύματα. Το 1989, ο Michael Waidner [4] του επεσήμανε ότι ένα τέτοιο αξιόπιστο δίκτυο μετάδοσης είναι αδύνατον να υλοποιηθεί με αλγοριθμικά μέσα. Έτσι, στη λύση που προτείνει στο άρθρο του χρησιμοποιεί την ήδη υπάρχουσα υλοποίηση των DC-net δικτύων, αλλά αντικαθιστά το αξιόπιστο δίκτυο επικοινωνίας με κάποιο αυθαίρετο.

Με βάση τον Waidner, αν υποθέσουμε ότι δύο χρήστες ανταλλάσσουν μηνύματα σε ένα δίκτυο και ένας εισβολέας έχει τη δυνατότητα να βλέπει την επικοινωνία τους, αλλά και να τροποποιεί τα μηνύματα, τότε πρέπει με κάποιο τρόπο να διακόπτεται η μετάδοση. Έτσι, στο πρωτόκολλο που περιγράφεται στο άρθρο, το οποίο ονομάστηκε *DC⁺-net*, κάθε χρήστης αρχικά λαμβάνει μία είσοδο από τους υπόλοιπους. Στην περίπτωση, που ο χρήστης που θα λάβει το μήνυμα είναι αξιόπιστος, όπως και ο χρήστης που το έστειλε, τότε αν το μήνυμα που έχει λάβει είναι διαφορετικό από αυτό που έστειλε ο πρώτος χρήστης, διακόπτεται η μετάδοση (*Fail-Stop Broadcast*), καθώς δε θεωρείται ασφαλής. Σε αντίθετη περίπτωση, συνεχίζεται η επικοινωνία τους.

Δυστυχώς, τα δίκτυα DC-net υποφέρουν από το πρόβλημα σύγκρουσης της μετάδοσης λόγω παρεμβολών από κακόβουλους χρήστες. Ένας τέτοιος χρήστης μπορεί να ξεκινήσει επίθεση Denial-of-service, επιλέγοντας να στέλνει μηνύματα σε κάθε γύρο, ή να εγκαταλείψει το δίκτυο προκαλώντας έτσι την αναστάτωσή του. Η επίθεση αυτή δεν μπορεί να εντοπιστεί, καθώς κάθε κόμβος είναι ανώνυμος.

Για να λυθεί αυτό το πρόβλημα, ο Chaum είχε προτείνει τη χρήση παγίδων για να ανιχνεύει τους ανέντιμους παίκτες. Πριν από τη μετάδοση του μηνύματος, κάθε συμμετέχων πρέπει να διατηρήσει ακριβώς μία υποδοχή, την οποία θα την χρησιμοποιεί για να στείλει ένα πραγματικό μήνυμα ή κάποια παγίδα. Ένας ανέντιμος χρήστης θα προσπαθήσει να μεταδώσει μήνυμα σε μία υποδοχή που δεν έχει ανατεθεί σε αυτόν. Αλλά, αν η υποδοχή που θα προσπαθήσει να στείλει είναι παγίδα, τότε ο κακόβουλος χρήστης θα ανιχνευθεί.

Δυστυχώς, όμως είναι πολύ εύκολο για έναν εισβολέα να πλαστογραφήσει την παγίδα σε μία αυθαίρετη υποδοχή. Για αυτό το λόγο, ο Wainder στο DC⁺-net πρωτόκολλο που ανέπτυξε, παρουσίασε τη λύση δημιουργίας των παγίδων τη στιγμή που αναθέτονται οι υποδοχές στους χρήστες. Όμως, και πάλι το σύστημα θα μπορέσει να αναγνωρίσει μόνο έναν ανέντιμο χρήστη με τη χρήση της παγίδας. Έτσι, δεν υπάρχει κάποια άλλη λύση για την ανίχνευση του σφάλματος, πέρα από την αναμετάδοση.

2.1.2 XOR-trees

Ακόμα ένα πρωτόκολλο που βασίζεται στα δίκτυα DC-net είναι τα *δέντρα XOR (XOR-trees)*. Η συγκεκριμένη αρχιτεκτονική προτάθηκε από τους Dolev και Ostrovsky [5] και παρέχει ανωνυμία στον αποστολέα, στον παραλήπτη, αλλά και στην επικοινωνία τους.

Στο συγκεκριμένο σύστημα, παρουσιάζεται η έννοια της προκαθορισμένης σειράς μετάδοσης. Στα προηγούμενα συστήματα που είχαν υλοποιηθεί, οι επεξεργαστές μπορούσαν ταυτόχρονα να ξεκινήσουν τη μετάδοση πληροφοριών, προκαλώντας έτσι μεγάλη καθυστέρηση στις επικοινωνίες. Έτσι, στα δέντρα XOR, όταν εντοπίζεται μία τέτοια κατάσταση, οι επεξεργαστές σταματούν τη μετάδοση και ο καθένας ορίζει ένα τυχαίο διάστημα αναμονής και πραγματοποιεί τότε τη μετάδοσή του. Αυτό έχει ως αποτέλεσμα, ένας εξωτερικός παρατηρητής να συμπεραίνει ότι ο κάθε επεξεργαστής θέλει να επικοινωνήσει με διάφορους άλλους και όχι με κάποιον συγκεκριμένα. Για την επίλυση του συγκεκριμένου προβλήματος έχει δοθεί σε κάθε επεξεργαστή μία χρονοθυρίδα, στην οποία μπορεί να ζητήσει να επικοινωνήσει με άλλους. Όλες αυτές οι κρυπτογραφημένες αιτήσεις φθάνουν σε τυχαία σειρά σε έναν μόνο επεξεργαστή, που με τη σειρά του υπολογίζει τον συνολικό αριθμό των αιτήσεων, χωρίς όμως να μπορέσει να κερδίσει οποιαδήποτε πληροφορία σχετικά με την ταυτότητα των αιτούμενων επεξεργαστών.

Επίσης, στη συγκεκριμένη αρχιτεκτονική έχει οριστεί για πρώτη φορά η χρήση μίας επιπλέον ακολουθίας, που παράγεται από μία γεννήτρια ψευδοτυχαίων αριθμών, η οποία μοιράζεται ανάμεσα στον αποστολέα και τον παραλήπτη για την κρυπτογράφηση και αποκρυπτογράφηση του μηνύματος. Αυτή η νέα προσέγγιση είναι κατάλληλη για τη μετάδοση μίας μεγάλης ακολουθίας από bits, όπως είναι ένα βίντεο.

Το πρωτόκολλο των δέντρων XOR αποτελεί ένα υπολογιστικά αποδοτικό σύστημα, το οποίο όμως διατηρεί το βασικότερο μειονέκτημα των DC-net, καθώς και πάλι μόνο ένα ζεύγος χρηστών μπορούν να επικοινωνήσουν σε συγκεκριμένη χρονική στιγμή. Έτσι, η απόδοση του συστήματος ελαττώνεται λόγω των συγκρούσεων που υπάρχουν με την αύξηση του αριθμού των χρηστών.

2.1.3 Herbivore

Το Herbivore [6] αποτελεί ένα ακόμα πρωτόκολλο βασισμένο στα DC-net δίκτυα. Το πρωτόκολλο Herbivore αναπτύχθηκε το 2002 από τους Sharad Goel και Mark Robson και παρέχει ταυτόχρονα επεκτασιμότητα, αποδοτικότητα και ισχυρή ανωνυμία. Υπάρχουν δύο συστατικά στο συγκεκριμένο σύστημα. Στο χαμηλότερο επίπεδο, ένα κυκλικό πρωτόκολλο ρυθμίζει τον τρόπο αποστολής των bits μεταξύ των συμμετέχοντων κόμβων. Έτσι, επιτυγχάνεται ισχυρή ανωνυμία με την αξιοποίηση των δυνατοτήτων των DC-net σε επίπεδο σύρματος, ενώ ταυτόχρονα επεκτείνεται η βασική αρχιτεκτονική των DC-net με διάφορους τρόπους, ώστε να αξιοποιηθεί το δίκτυο αποτελεσματικά και να ανιχνεύονται παραβιάσεις.

Το κυκλικό πρωτόκολλο παρέχει αποτελεσματική και ανώνυμη επικοινωνία, ωστόσο η απόδοσή του είναι αντιστρόφως ανάλογη με τον αριθμό των ταυτόχρονα ενεργών χρηστών και η ύπαρξη κακόβουλων κόμβων μπορεί να επηρεάσει τους συμμετέχοντες. Έτσι, για να μπορέσει το πρωτόκολλο Herbivore να λειτουργήσει αποτελεσματικά σε δίκτυα ευρείας κλίμακας, χρησιμοποιείται ένας αλγόριθμος ελέγχου της παγκόσμιας τοπολογίας, ώστε να διαιρέσει το δίκτυο σε μικρότερες ανώνυμες κλίκες. Το πρωτόκολλο εγγυάται ότι κάθε κλίκα θα έχει έναν προκαθορισμένο αριθμό κόμβων, που καθορίζεται από τον βαθμό της ανωνυμίας που προσφέρει το σύστημα. Στην περίπτωση που οι ήδη υπάρχουσες κλίκες περιλαμβάνουν μεγάλο αριθμό χρηστών, δημιουργούνται νέες κλίκες, για την βελτιστοποίηση της απόδοσης του συστήματος. Στην αντίθετη περίπτωση που ο αριθμός των ενεργών χρηστών σε μία κλίκα ελαττωθεί σε σχέση με τον προκαθορισμένο αριθμό κόμβων, τότε οι κόμβοι στην εν λόγω κλίκα κατανέμονται σε όλο το δίκτυο. Τέλος, υπάρχει κι ένα ασφαλές πρωτόκολλο εισόδου που στοχεύει να αποθαρρύνει τυχόν προσπάθειες από κακόβουλους χρήστες να ανατρέψουν τη λειτουργικότητα του πρωτοκόλλου Herbivore.

Το πρωτόκολλο Herbivore με την υλοποίησή του έχει διορθώσει πολλά από τα προβλήματα των DC-net δικτύων. Παρέχει ισχυρή ανωνυμία και θεωρείται αποδοτικό για μικρής έκτασης δίκτυα. Όμως, το συγκεκριμένο πρωτόκολλο παρουσιάζει καθυστερήσεις στην επικοινωνία σε μεγάλης εμβέλειας δίκτυα σε σχέση με επόμενες υλοποιήσεις.

2.1.4 Dissent

Το πρωτόκολλο *Dissent* [7] αναπτύχθηκε το 2010 από τους D.I. Wolinsky, H. Corrigan-Gibbs, B. Ford και ανήκει στην κατηγορία των DC-net. Η συγκεκριμένη υλοποίηση παρέχει ανωνυμία στον αποστολέα, στον παραλήπτη, αλλά και στη μεταξύ τους επικοινωνία και αποτελείται από δύο επιμέρους πρωτόκολλα: ένα *πρωτόκολλο ανακατέματος (shuffle protocol)* και ένα *πρωτόκολλο όγκου (bulk protocol)*. Επίσης, επιτρέπει σε μία καλά καθορισμένη ομάδα από χρήστες την ανταλλαγή μηνυμάτων μεταβλητού μεγέθους χωρίς τους κινδύνους της ανάλυσης κυκλοφορίας ή των ανώνυμων επιθέσεων DoS που σχετίζονται με τα δίκτυα mix-net και DC-net.

Το πρωτόκολλο shuffle χρησιμοποιείται προκειμένου να σχηματιστεί ένα δρομολόγιο DC-net μετάδοσης και να διανεμηθούν “αναθέσεις μετάδοσης” που θα επιτρέψουν τον έλεγχο της ορθότητας των δεδομένων των επόμενων DC-net δικτύων. Η τεχνική αυτή απαιτεί σε κάθε γύρο να γίνεται η επαλήθευση των δεδομένων, κάτι που μπορεί να οδηγήσει σε μεγάλες καθυστερήσεις.

Το πρωτόκολλο όγκου (bulk protocol) προσπαθεί να αντιμετωπίσει αποτελεσματικά το πρόβλημα της αποστολής μεγάλου και μεταβλητού μήκους μηνυμάτων.

Επομένως, τα χαρακτηριστικά που προστέθηκαν στο Dissent, το έχουν καταστήσει ένα πρακτικό DC-net πρωτόκολλο για μη διαδραστική επικοινωνία σε ομάδες μεσαίου μεγέθους.

2.1.5 Verdict

Το πρωτόκολλο *Verdict* [8] στηρίχτηκε για την υλοποίησή του στα προληπτικά επαληθεύσιμα DC-net δίκτυα, δηλαδή οι συμμετέχοντες χρησιμοποιούν κρυπτογραφία δημόσιου κλειδιού για την κατασκευή DC-net

κρυπτογραφημένων δεδομένων και χρησιμοποιούν μηδενικής γνώσης αποδείξεις για να εντοπίσουν και να αποκλείσουν ανάρμοστη συμπεριφορά στο δίκτυο, πριν από την πρόκληση αναστάτωσης. Το συγκεκριμένο πρωτόκολλο υλοποιήθηκε το 2013 από τους τρεις δημιουργούς του Dissent, D.I. Wolinsky, H. Corrigan-Gibbs και B. Ford και στοχεύει στην αντιμετώπιση των προβλημάτων που υπήρχαν στον προκάτοχό του. Ο προληπτικός αποκλεισμός των κακόβουλων επιθέσεων ανακουφίζει το σύστημα από την ανάγκη ανίχνευσης μιας διαταραχής μετά την επίθεση.

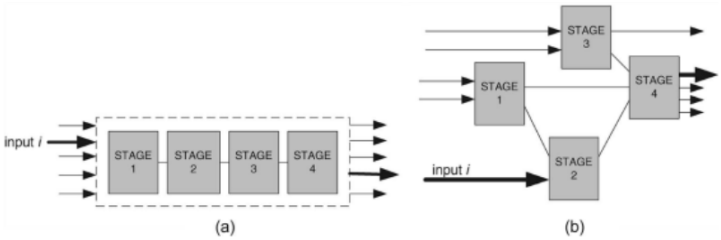
Το Verdict αποτελεί το καταλληλότερο DC-net πρωτόκολλο για την αποστολή μηνυμάτων σε μικρές ομάδες χρηστών και τη διασφάλιση της επικοινωνίας και της ανωνυμίας τους. Όμως, λόγω της χρήσης της κρυπτογραφίας δημόσιου κλειδιού για την κρυπτογράφηση του μηνύματος, αυξάνεται πάρα πολύ το κόστος υπολογισμού, με αποτέλεσμα την ύπαρξη καθυστερήσεων στο δίκτυο όταν υπάρχει μεγάλος αριθμός χρηστών.

2.2 Mix-net συστήματα

Όλα τα *mix-net* συστήματα ασφαλείας βασίζονται στην εργασία του David Chaum [3] το 1981. Ένα mix-net είναι ένα πολυσταδιακό σύστημα που δέχεται κάποιες εισόδους και παράγει ως έξοδο τα κρυπτογραφήματά τους μετατιθέμενα. Για την επίτευξη αυτού, χρησιμοποιείται μία σειρά από εξυπηρέτες (mix servers) που συνδυάζουν τα λαμβανόμενα πακέτα για να δημιουργήσουν ασαφή μονοπάτια για την επικοινωνία του αποστολέα και του παραλήπτη.

Το κύριο συστατικό της αρχιτεκτονικής mixnet είναι το *στάδιο (stage)*, γνωστό και ως *μείγμα (mix)*, το οποίο αναμειγνύει τις εισόδους που λαμβάνει. Η λειτουργία της ανάμιξης περιλαμβάνει έναν κρυπτογραφικό μετασχηματισμό, που αλλάζει τη μορφή των δεδομένων εισόδου. Στη συνέχεια, τα κρυπτογραφημένα δεδομένα θα προωθηθούν παράλληλα από το mix στον επόμενο προορισμό τους.

Ένα mixnet δίκτυο αποτελείται από πολλά διασυνδεδεμένα mixes, ανάλογα με την ευρωστία της ανωνυμίας που απαιτείται. Η διασύνδεση των mixes καθορίζει την τοπολογία mixnet και με βάση την τοπολογία μπορεί να υπάρχει *mixnet καταρράκτη* (εικόνα 2.1-a) ή *mixnet ελεύθερης διαδρομής* (εικόνα 2.1-b).



Εικόνα 2.1 Τοπολογίες mixnet δικτύων

Ένα mixnet καταρράκτη αποτελείται από mixes που συνδέονται σε μία σταθερή, διαδοχική σειρά. Το πρώτο mix στο mixnet καταρράκτη λαμβάνει ένα πλήθος από δεδομένα εισόδου, τα αναμειγνύει και στη συνέχεια τα μεταφέρει στο δεύτερο συνδεδεμένο mix. Το δεύτερο mix επαναλαμβάνει τη ανάμιξη και προώθηση, και η διαδικασία συνεχίζεται μέχρι το τελικό mix να εξάγει τις μη ανιχνεύσιμες εισόδους. Έτσι, σε ένα mixnet καταρράκτη όλα τα μηνύματα διασχίζουν την ίδια διαδρομή. Αντίθετα, σε ένα mixnet ελεύθερης διαδρομής μπορούν να υπάρχουν πολλές διαθέσιμες διαδρομές για κάθε ένα από τα δεδομένα εισόδου.

Η ασφάλεια των mixnet συστημάτων βασίζεται στη σχέση εμπιστοσύνης μεταξύ των διαμεσολαβητών, με αποτέλεσμα να μην μπορούν να παρέχουν άνευ όρων ανωνυμία. Οι επιθέσεις που μπορούν να λάβουν χώρα σε ένα mixnet δίκτυο είναι η ανάλυση της κυκλοφορίας, στην οποία ο εισβολέας παρατηρεί την κίνηση που εισέρχεται και εξέρχεται από τα mixes, με στόχο τη συσχέτιση των δεδομένων εισόδου με αυτά της εξόδου. Επίσης, μία επίθεση ενεργού τύπου, που μπορεί να πραγματοποιηθεί σε ένα mixnet δίκτυο είναι η χειραγώγηση της κυκλοφορίας. Ο επιτιθέμενος εισάγει αλλοιωμένα δεδομένα εισόδου στα mixes, με σκοπό την παραβίαση της ακεραιότητας του συστήματος και την ανίχνευση των αλλοιωμένων δεδομένων στην έξοδο.

Παρακάτω, θα παρουσιαστούν τα πρωτόκολλα που έχουν αναπτυχθεί κατά καιρούς και στηρίζονται στον μηχανισμό των Mix net συστημάτων.

2.2.1 Chaum's mix network

Η έρευνα του Chaum έχει εφαρμοστεί ως κεντρικό εργαλείο για πολλές εφαρμογές που έχουν αναπτυχθεί κατά καιρούς. Σε σύγκριση με τα DC-net δίκτυα, το βασικότερο πλεονέκτημα των δικτύων Mix-net είναι το χαμηλό κόστος επικοινωνίας.

Για εφαρμογές ανώνυμου ηλεκτρονικού ταχυδρομείου, το Mix net του Chaum μπορεί να θεωρηθεί ως θεμελιώδες στοιχείο κρυπτογράφησης δημόσιου κλειδιού, που παίρνει ως είσοδο μία σειρά από δεδομένα κρυπτογραφημένα με το δημόσιο κλειδί των διακομιστών αναμετάδοσης, που ονομάζονται mixes, τα αποκρυπτογραφεί και τα ανακατεύει ώστε να εξάγει μία τυχαία μετάθεση των αποκρυπτογραφημένων στοιχείων. Για να εξασφαλιστεί η ανωνυμία, γίνονται πολλαπλές μίξεις με μυστικές μεταθέσεις, προκειμένου η αντιστοιχία των δεδομένων στην είσοδο και εκείνων στην έξοδο να είναι κρυμμένη, ακόμα και αν ένα ή περισσότερα από τα μείγματα έχουν καταστραφεί. Ως αποτέλεσμα, το σύστημα των mix-nets παρέχει ασφάλεια των διευθύνσεων αποστολής και προορισμού για τα μέρη που επικοινωνούν.

Για πολλαπλές αλληλουχίες, η δομή των μιγμάτων είναι ανάλογη της μετάδοσης του μηνύματος με πολλαπλούς φακέλους που κατασκευάζονται από τον αποστολέα, με τη διεύθυνση του πρώτος μείγματος να είναι ο εξωτερικός φάκελος και η διεύθυνση του τελευταίου να είναι ο ενδότατος φάκελος. Η ιδέα των κλιμακούμενου φακέλου ονομάζεται κρεμμύδι (onion) στη δρομολόγηση κρεμμυδιού (onion routing).

Το 1989, οι B. Pfitzmann και A. Pfitzmann [9] απέδειξαν ότι το πρωτόκολλο που αναπτύχθηκε από τον Chaum δεν παρέχει την απαραίτητη μη συνδεσιμότητα (unlinkability) του αποστολέα και του παραλήπτη. Με άλλα λόγια, κάποιος τρίτος μπορεί να πάρει ένα μήνυμα εξόδου, να το κρυπτογραφήσει ξανά και να ελέγξει τα μηνύματα εισόδου που λαμβάνονται, αντιστρέφοντας με αυτό τον τρόπο το μείγμα. Για να αποφευχθεί κάτι τέτοιο, όλα τα πακέτα θα πρέπει να έχουν το ίδιο μέγεθος με τυχαίες ακολουθίες χαρακτήρων. Έτσι, τα μηνύματα εξόδου θα είναι δυσδιάκριτα από τους κακόβουλους χρήστες και θα αποτρέπεται η ανάλυση της κυκλοφορίας του δικτύου.

Ο Chaum στο κείμενό του πρότεινε επίσης τη χρήση ψηφιακών ψευδωνύμων που παρέχουν μη ιχνηλασιμότητα των διευθύνσεων επιστροφής. Ωστόσο, αποδείχτηκε μετέπειτα ότι στην περίπτωση που χρησιμοποιηθεί ο αλγόριθμος RSA για την κρυπτογράφηση δημόσιου κλειδιού, τα μείγματα που θα προκύψουν μπορούν να σπάσουν από κάποια επίθεση.

2.2.2 Anonymous remailers

Με τον όρο *ανώνυμος remailer* καλείται ένας εξυπηρέτης που λαμβάνει μηνύματα με ενσωματωμένες οδηγίες για το πού θα τα στείλει μετά, χωρίς να αποκαλύψει από πού αρχικά προήλθαν. Έχουν αναπτυχθεί τριών τύπων remailers, οι *Cypherpunk* ανώνυμοι remailers (τύπου I), οι *Mixmaster* ανώνυμοι remailers (τύπου II) και οι *διακομιστές nym* (*Nym Servers* - τύπου III). Οι τρεις τύποι διαφέρουν μεταξύ τους ως προς τον τρόπο λειτουργίας τους, τις πολιτικές που υιοθετούν και το είδος της επίθεσης κατά της ανωνυμίας του ηλεκτρονικού ταχυδρομείου που καλούνται να αντισταθούν.

- *Cypherpunk remailers*

Οι Cypherpunk remailers είναι η πρώτη ευρεία δημόσια εκτέλεση της mix net αρχιτεκτονικής. Καλούνται επίσης remailers τύπου I. Οι τύπου I remailers επιχειρούν να περιορίσουν την ανάλυσης της κυκλοφορίας, παρέχοντας μία αρχιτεκτονική ανώνυμης αποθήκευσης και προώθησης. Οι Cypherpunk remailers προωθούν τα μηνύματα σε διάφορα συστήματα πριν καταλήξουν στον παραλήπτη, απογυμνώνοντας κάθε φορά την ταυτότητα του κάθε συνδέσμου. Επιπλέον, προκειμένου να αποφευχθούν επιθέσεις επανάληψης (replay attacks), κάθε Cypherpunk remailer κρατάει μόνο ένα αρχείο καταγραφής των απεσταλμένων μηνυμάτων, χωρίς την ταυτότητα των εμπλεκόμενων μελών.

- *Mixmaster remailers*

Οι Mixmaster remailers (ή remailer τύπου II) αναπτύχθηκαν αρχικά από τον Lance Cottrell και στηρίζουν την αρχιτεκτονική τους στην υπόθεση ότι κάθε σύνδεση με το δίκτυο είναι υπό παρακολούθηση. Προκειμένου να προστατέψουν τα μηνύματα από την παρακολούθησης της κίνησης του δικτύου, οι

Mixmaster remailers περιλαμβάνουν padding, pool μηνυμάτων και άλλους τρόπους μίξης. Οι remailers τύπου II είναι πιο ανθεκτικοί από τους Cypherpunk remailers στην ανάλυση της κίνησης, στους αναξιόπιστους κόμβους και σε άλλες επιθέσεις. Ωστόσο, και αυτή η κατηγορία εξακολουθεί να εφαρμόζει επαναχρησιμοποιήσιμα *μπλοκ απαντήσεων (reply blocks)*, θέτοντας έτσι σε κίνδυνο την ασφάλεια. Ένας εισβολέας μπορεί να χρησιμοποιήσει αυτή την ιδιότητα για να εντοπίσει τη διαδρομή του παραλήπτη: εάν δύο εισερχόμενες συνδέσεις περιλαμβάνουν ένα μήνυμα στο ίδιο reply block, τότε το επόμενο βήμα είναι το σημείο τομής των δύο συνδέσεων.

- *Mixminion remailers*

Η υλοποίηση Mixminion [10] αποτελεί την εφαρμογή του ανώνυμου πρωτοκόλλου του τύπου III remailer. Χρησιμοποιεί την αρχιτεκτονική της ελεύθερης διαδρομής σε ένα Mix-net δίκτυο, όπως και οι remailers τύπου II, προκειμένου να παρέχει ισχυρή ανωνυμία, αλλά και να εμποδίσει τους κακόβουλους χρήστες να ανακαλύψουν τους χρήστες που επικοινωνούν. Κάθε e-mail περνά μέσα από διάφορα μείγματα, τα οποία γνωρίζουν μόνο τη διαδρομή στα γειτονικά τους μείγματα. Έτσι, κανένα μείγμα δεν μπορεί να συνδέσει τον αποστολέα του μηνύματος με τον παραλήπτη.

Τα μηνύματα στο πρωτόκολλο Mixminion αποτελούνται από ένα τμήμα επικεφαλίδας και το ωφέλιμο φορτίο (payload). Κάθε επικεφαλίδα χωρίζεται στην κύρια και σε μία δευτερεύουσα, κρυπτογραφημένη με τα δημόσια κλειδιά των ενδιάμεσων κόμβων. Το Mixminion επιτρέπει την αποστολή μηνυμάτων με τρεις διαφορετικούς τρόπους: *προώθηση (forward)*, *άμεση απάντηση (direct reply)* και *ανώνυμη απάντηση (anonymized reply)*. Στην προώθηση του μηνύματος, μόνο ο αποστολέας θα είναι ανώνυμος, ενώ αντίθετα στην άμεση απάντηση, ο παραλήπτης θα είναι ανώνυμος. Τέλος, στην ανώνυμη απάντηση τόσο ο παραλήπτης όσο και ο αποστολέας είναι ανώνυμοι.

Το πρωτόκολλο Mixminion διασπά κάθε μήνυμα σε ίδιου μεγέθους πακέτα και επιλέγει μία διαδρομή μέσω του mix-net δικτύου για το κάθε πακέτο. Το Mixminion, σε αντίθεση με τους άλλους δύο τύπους remailers, δεν εφαρμόζει επαναχρησιμοποιήσιμα reply blocks. Αντ' αυτού, χρησιμοποιεί ένα *reply block μιας χρήσης (SURBs)* για να επιτρέψει την ανωνυμία των παραληπτών. Επίσης, το συγκεκριμένο πρωτόκολλο κάνει δυσδιάκριτα τα προωθημένα μηνύματα, καθώς και τις απαντήσεις προκειμένου να αποφευχθεί μία *επίθεση ετικετών (tagging attack)*. Σε μία επίθεση ετικετών, ένας εισβολέας τροποποιεί μέρος του μηνύματος προκειμένου να αναγνωρίζει στη συνέχεια την ετικέτα του και να το εντοπίζει στη διαδρομή. Έτσι, χρησιμοποιούνται κρυπτογραφικά συστήματα ελέγχου για την προστασία των κεφαλίδων, προκειμένου να εξασφαλίσουν ότι οι πληροφορίες διεύθυνσης που περιέχονται στις κεφαλίδες καταστρέφονται στην περίπτωση που το payload έχει τροποποιηθεί.

Σε αντίθεση με τους άλλους δύο τύπους remailers που χρησιμοποιούν ως μηχανισμό μεταφορά το SMTP, οι πελάτες και οι κόμβοι του Mixminion επικοινωνούν χρησιμοποιώντας ένα TLS κρυπτογραφημένο κανάλι. Το πρωτόκολλο TLS επιτρέπει τη δημιουργία ενός κοινού κλειδιού συνεδρίας και την αποστολή του σε ένα κρυπτογραφημένο κανάλι βασισμένο στον αλγόριθμο του Diffie-Hellman για την ανταλλαγή κλειδιών. Για να διασφαλιστεί η εγκυρότητα του κοινού μυστικού κλειδιού, σε σχέση με αυτό που έστειλε ο αποστολέας, ο παραλήπτης οφείλει να το υπογράψει. Από τη στιγμή που έχει γίνει και στα δύο μέρη γνωστό το κλειδί συνόδου, διαγράφονται τα προσωρινά κλειδιά Diffie-Hellman και ξεκινά η επικοινωνία μέσα από το ασφαλές κανάλι.

Η κρυπτογραφημένη σύνδεση παρέχει απόρρητη επικοινωνία μεταξύ των χρηστών, καθώς ακόμα και οι κόμβοι που συμμετέχουν στην ανταλλαγή δεν μπορούν να αποκρυπτογραφήσουν ή να αναγνωρίσουν τα μηνύματα. Επιπλέον, η κρυπτογραφημένη σύνδεση κάνει δύσκολη την ύπαρξη ενεργών και παθητικών επιθέσεων, δεδομένου ότι κάθε μήνυμα προσδιορίζει την ταυτότητα του κόμβου από τον οποίο προήλθε.

Ωστόσο, η κρυπτογράφηση της σύνδεσης παρέχει περιορισμένη προστασία κατά της ανάλυσης της κυκλοφορίας. Παρόλο που ο εισβολέας δεν μπορεί να αναγνωρίσει ακόμη και τα δικά του μηνύματα, μπορεί ακόμα να μετρήσει την κίνηση που μεταδίδεται. Έτσι, η εφαρμογή του Mixminion έφερε στην επιφάνεια κάποια πρακτικά ζητήματα. Δεδομένου ότι η μετάδοση του μηνύματος είναι αναξιόπιστη, είναι σημαντικό να εφαρμοστούν μηχανισμοί για την αναμετάδοση και τη διόρθωση λάθος προώθησης (*Forward Error Correction – FEC*). Όμως, τα συγκεκριμένα συστήματα μπορούν να οδηγήσουν στην ανάλυση της κίνησης. Η ασφάλεια αυτού του συστήματα απαιτεί από όλους τους πελάτες αν έχουν πλήρη γνώση του δικτύου και μία κατανεμημένα υπηρεσία καταλόγου πρέπει να προσδιοριστεί για την διανομή αυτών των πληροφοριών.

2.2.3 Anonymizer

Η εφαρμογή *Anonymizer* [11] δημιουργήθηκε από την εταιρία Anonymizer Inc., η οποία ιδρύθηκε το 1995 από τον Lance Cottrell, ιδρυτή του Mixmaster remailer. Η συγκεκριμένη εφαρμογή αποτελεί μία απλή λύση για την ανωνυμία του αποστολέα (sender anonymity), καθώς όλη η δραστηριότητα πραγματοποιείται διαμέσου ενός διακομιστή διαμεσολάβησης (proxy server). Με άλλα λόγια, ο proxy server καλύπτει την πραγματική IP διεύθυνση του χρήστη και τον προστατεύει από επιθέσεις του Διαδικτύου, διασφαλίζοντας συνέχεια την πλήρη ανωνυμία του.

Το μειονέκτημα της συγκεκριμένης αρχιτεκτονικής είναι ότι οι χρήστες θα πρέπει να εμπιστεύονται τους proxy servers ότι δεν θα παρακολουθούν όλες τις δραστηριότητές τους.

2.2.4 Web MIXes

Το σύστημα των *Web Mixes* [12] προτάθηκε από τους Berthold, Federrath και Köpsell το 2001. Είχε σχεδιαστεί για ανώνυμη και μη ανιχνεύσιμη πραγματικού χρόνου πρόσβαση στο Διαδίκτυο και στόχευε στην αποτροπή της ανάλυσης της κυκλοφορίας, αλλά και των ***επιθέσεων πλημμύρας (flooding attacks)***.

Το ολοκληρωμένο σύστημα των Web Mixes αποτελείται από τρία λογικά μέρη: το *Java Anon Proxy (JAP)* από την πλευρά του πελάτη, τα *MIXes* και το *Cache-proxy* που βρίσκεται στον διακομιστή. Το JAP είναι ένα πρόγραμμα που εγκαθίσταται τοπικά στον υπολογιστή κάθε χρήστη. Το MIX χρησιμεύει για να αναμείξει τη σειρά των ρευμάτων δεδομένων και να αλλάξει την κωδικοποίησή τους χρησιμοποιώντας μεθόδους κρυπτογράφησης. Τα MIXes είναι υπολογιστές που συνδέονται μέσω του Διαδικτύου και στοχεύουν στην προστασία των χρηστών από ***επιθέσεις συσχέτισης κυκλοφορίας (traffic correlation attacks)***. Σχηματίζουν μία λογική αλυσίδα, που ονομάζεται *MIX-cascade*. Το πρώτο MIX λαμβάνει τα δεδομένα από το JAPs, ενώ το τελευταίο MIX στέλνει τα δεδομένα στο Cache-proxy του διακομιστή. Το Cache-proxy του διακομιστή αναλαμβάνει να στείλει τα δεδομένα στο Διαδίκτυο και να λάβει τις απαντήσεις από τους διακομιστές. Αυτά τα τρία μέρη συνενώνονται σε μία αλυσίδα για να χτίσουν ένα ανώνυμο κανάλι, μέσω του οποίου θα περνάει ανώνυμα όλη η κυκλοφορία.

Έχουν προταθεί πολλές επιθέσεις εναντίον των συστημάτων MIXes για να λάβουν πληροφορίες για τον χρήστη ή να σταματήσουν την υπηρεσία. Για την αποτροπή των παθητικών επιθέσεων, στα συγκεκριμένα συστήματα, ο πελάτης στέλνει και ψεύτικα μηνύματα στο MIX, προκειμένου να κάνει την ανάλυση της κυκλοφορίας δυσκολότερη. Η αποστολή ψεύτικων μηνυμάτων εγγυάται ότι όλοι οι χρήστες στέλνουν ίδια ποσότητα δεδομένων σε κάθε χρονική στιγμή. Έτσι, ένας κακόβουλος χρήστης του συστήματος δεν μπορεί να αναγνωρίσει ποια μηνύματα περιέχουν πραγματικά δεδομένα. Επίσης, κάθε MIX στέλνει παραπλανητικά μηνύματα προς το χρήστη, όταν δεν λαμβάνει πραγματικά. Αυτό διασφαλίζει ότι ο κάθε πελάτης του συστήματος λαμβάνει την ίδια ποσότητα δεδομένων κάθε χρονική στιγμή.

Τέλος, στο πρωτόκολλο MIXes προτάθηκε ένας *αλγόριθμος chop-and-slice*, έτσι ώστε τα μεγάλα μηνύματα να τεμαχίζονται σε μικρότερα κομμάτια συγκεκριμένου μεγέθους. Κάθε κομμάτι μεταδίδεται μέσω ενός ανώνυμου καναλιού MIX.

2.2.5 Onion routing

Η *δρομολόγηση κρεμμύδι (Onion routing)* [13,14] αναπτύχθηκε το 1996 από τους Reed, Syverson και Goldschlag. Πρόκειται για ένα κατανεμημένο δίκτυο επικάλυψης που σχεδιάστηκε για να την ανωνυμοποίηση των TCP επικοινωνιών. Η onion δομή δεδομένων (ή απλά onion) αποτελείται από στρώματα κρυπτογράφησης τυλιγμένα γύρω από το payload. Κάθε *onion δρομολογητής (onion router – OR)* είναι μία συσκευή αποθήκευσης και προώθησης (store-and-forward), η οποία δέχεται μηνύματα σταθερού μεγέθους, εκτελεί κρυπτογραφικές λειτουργίες και στη συνέχεια τα προωθεί στον επόμενο κόμβο της διαδρομής δρομολόγησης. Όταν ο OR λαμβάνει ένα μήνυμα, γνωρίζει τον αμέσως προηγούμενο κόμβο που του το έστειλε. Αφαιρώντας ένα στρώμα κρυπτογράφησης χρησιμοποιώντας το ιδιωτικό του κλειδί λαμβάνει οδηγίες για τη δρομολόγηση του μηνύματος στον επόμενο OR. Η διαδικασία επαναλαμβάνεται έως ότου το μήνυμα να παραδοθεί στον τελικό onion router.

Αν και το onion routing μπορεί να χρησιμοποιηθεί για ανώνυμες επικοινωνίες, διαφέρει από τους remailers στο γεγονός ότι γίνεται σε πραγματικό χρόνο αμφίδρομη επικοινωνία, χωρίς να απαιτείται οι

κόμβοι του δικτύου να γνωρίζουν την ταυτότητα προέλευσης του μηνύματος ή την τοποθεσία. Επίσης, παρόλο που το Mix-net του Chaum αποθηκεύει τα μηνύματα για αόριστο χρονικό διάστημα περιμένοντας να λάβει επαρκή αριθμό των μηνυμάτων που αναμειγνύονται μαζί, το onion routing έχει σχεδιαστεί να αποστέλλει τις πληροφορίες σε πραγματικό χρόνο, γεγονός που περιορίζει την μίξη κα ενδεχομένως αποδυναμώνει την προστασία.

Αυτό που θα πρέπει να παρατηρηθεί είναι ότι σε κάθε βήμα, όταν αποκολλάται ένα στρώμα, το "κρεμμύδι" συρρικνώνεται. Επομένως, για την αποφυγή εύρεσης των πληροφοριών δρομολόγησης λόγω της συνεχούς μείωσης του πακέτου, μία τυχαία σειρά δυαδικών ψηφίων μεγέθους όσο το κομμάτι που αποκολλήθηκε, προσαρτάται στο τέλος του payload πριν την προώθηση. Στην πραγματικότητα, ακόμα και ένα σταθερού μεγέθους onion μπορεί να εντοπιστεί στο δίκτυο από κάποιον κακόβουλο, εκτός και αν όλα τα onion έχουν το ίδιο μέγεθος. Για να λυθεί το συγκεκριμένο πρόβλημα, στο onion routing το μέγεθος του onion είναι σταθερό. Αυτό απαιτεί ο αρχικός δρομολογητής να γεμίσει το φορτίο με βάση το προκαθορισμένο μέγεθος του onion. Όταν το πακέτο φτάσει στο proxy server του παραλήπτη θα πρέπει να έχει το ίδιο μέγεθος με το αρχικό.

Το κόστος για την εγκαθίδρυση ενός δικτύου onion routing είναι σχετικά μικρό. Η εγκατάσταση της σύνδεσης είναι το ίδιο αισθητή όσο άλλες καθυστερήσεις, όπως η ρύθμιση της κανονικής Web σύνδεσης. Η λειτουργία κρυπτογράφησης δημόσιου κλειδιού που χρειάζεται υπολογιστικούς πόρους, απαιτείται μόνο κατά τη φάση της εγκατάστασης της σύνδεσης.

Το σύστημα του onion routing παρέχει τη λειτουργία της μη συνδεσιμότητας σε μεγάλο βαθμό. Με άλλα λόγια, ένας εισβολέας δεν μπορεί να προσδιορίσει εύκολα τον αποστολέα και τον παραλήπτη ενός συγκεκριμένου μηνύματος. Παρόλα αυτά ένας τοπικός κακόβουλος χρήστης μπορεί να παρακολουθεί τις δραστηριότητες ενός κόμβου αποστολής και λήψης μηνυμάτων.

2.2.6 Tor

Το *Tor* [15,16] αναπτύχθηκε το 2004 από τους αρχικούς δημιουργούς του onion routing, ως η δεύτερη γενιά του onion δρομολογητή. Αναπτύχθηκε προκειμένου να αντιμετωπίσει τους περιορισμούς του αρχικού onion routing. Το Tor παρέχει ασφάλεια ως προς την προώθηση των μηνυμάτων, έτσι ώστε οι χρήστες να μπορούν να συνδεθούν σε ιστοσελίδες του Διαδικτύου χωρίς να αποκαλύπτεται η τοποθεσία τους στις ίδιες τις ιστοσελίδες ή σε κάποιον παρατηρητή. Παρέχει επίσης έλεγχο συμφόρησης, εξυπηρέτες καταλόγου, έλεγχο της ακεραιότητας, καθώς και διαμορφωμένες πολιτικές εξόδου.

Το Tor διαφέρει από τα άλλα ανεπτυγμένα συστήματα, που στοχεύουν στην αποφυγή ανάλυσης της κυκλοφορίας, ως προς την ασφάλεια και την ευελιξία. Τα Mix-nets, όπως το Mixmaster και Mixminion, παρέχουν ισχυρή ανωνυμία αλλά προκύπτουν μεγάλες καθυστερήσεις, με αποτέλεσμα τα συγκεκριμένα πρωτόκολλα να είναι ακατάλληλα για περιήγηση στο Διαδίκτυο. Το Tor αντίθετα, έχει σχεδιαστεί με σκοπό να παρέχει χαμηλής καθυστέρησης επικοινωνία και διαδραστική κίνηση στο δίκτυο.

Το Tor αποτελεί ένα δίκτυο επικάλυψης. Χρησιμοποιεί μία παραδοσιακή αρχιτεκτονική δικτύου μέσα από μία λίστα εθελοντών διακομιστών που βρίσκονται σε μία υπηρεσία καταλόγου. Κάθε onion διακομιστής (OR) δουλεύει σαν μία κανονική διεργασία χρήστη, χωρίς ειδικά προνόμια, διατηρώντας επίσης μία TLS σύνδεση με κάθε άλλο OR.

Το Tor, σε αντίθεση με τα πιο απλά και χαμηλού κόστους συστήματα Anonymizer και Web MIXes, είναι ένα σύστημα κατανεμημένης εμπιστοσύνης. Με άλλα λόγια, βασίζεται σε ένα σύνολο από γνωστούς διακομιστές καταλόγου, οι οποίοι διοικούνται από ανεξάρτητα μέρη και οι οποίοι αποφασίζουν ποιοι κόμβοι μπορούν να ενταχθούν στο δίκτυο.

Η εγκαθίδρυση κυκλωμάτων είναι μία υπολογιστικά δαπανηρή διαδικασία και κατά κανόνα απαιτεί κρυπτογραφία δημόσιου κλειδιού. Ωστόσο, επειδή ένα κύκλωμα διασχίζει πολλούς διακομιστές και κάθε διακομιστής γνωρίζει μόνο τους παρακείμενους στο κύκλωμα, κανένας διακομιστής δε χρειάζεται να συνδέσει ένα χρήστη με τους συνεργάτες της επικοινωνίας του.

Οι onion δρομολογητές επικοινωνούν μεταξύ τους και με τους onion διακομιστές διαμεσολάβησης (onion proxies) των χρηστών μέσω συνδέσεων TLS. Ένας onion proxy ενός χρήστη χρησιμοποιεί έναν επαναληπτικό μηχανισμό για την κατασκευή κυκλωμάτων και διαπραγματεύεται το συμμετρικό κλειδί που θα έχει με τον κάθε onion router του κυκλώματος. Το αμφίδρομο κανάλι χρησιμοποιείται σε κάθε βήμα για να γίνει μία επικυρωμένη ανταλλαγή κλειδιού βασισμένη στον αλγόριθμο του Diffie-Hellman. Το

πρωτόκολλο TLS αποκρύπτει τα στοιχεία της σύνδεσης και αποτρέπει έναν εισβολέα να τροποποιήσει τα δεδομένα ή να υποδυθεί έναν OR.

Κάθε onion router διατηρεί δύο κλειδιά: ένα μακροπρόθεσμο κλειδί που αποτελεί την ταυτότητά του, και ένα μεσοπρόθεσμο onion κλειδί. Το κλειδί-ταυτότητα χρησιμοποιείται για την υπογραφή των TLS πιστοποιητικών και για τους καταλόγους. Αντίστοιχα, το onion κλειδί χρησιμοποιείται για την αποκρυπτογράφηση των αιτήσεων των χρηστών για δημιουργία κυκλώματος και ανταλλαγή κλειδιών συνεδρίας. Τα onion κλειδιά αλλάζουν σε τακτά χρονικά διαστήματα προκειμένου να περιοριστεί ο κίνδυνος έκθεσής τους.

Ένα από τα σημαντικότερα θέματα ευπάθειας για μία κρυμμένη υπηρεσία του Tor είναι η επιλογή του διακομιστή του πρώτου και του τελευταίου κόμβου στη διαδρομή επικοινωνίας. Στην περίπτωση που ένας εισβολέας μπορεί να παρακολουθήσει τα άκρα ενός κυκλώματος Tor, μπορεί να αντιληφθεί ποιοι χρήστες επικοινωνούν. Έτσι, τον Μάρτιο του 2011, ερευνητές μαζί με ανθρώπους από το Rocquencourt, το εθνικό ινστιτούτο έρευνας στην επιστήμη της πληροφορικής και του ελέγχου (Institut national de recherche en informatique et en automatique - INRIA), κατέγραψαν μια επίθεση ικανή να αποκαλύψει τη διεύθυνση IP των χρηστών του BitTorrent στο δίκτυο Tor. Η επίθεση bad apple χρησιμοποιεί τον σχεδιασμό του Tor και εκμεταλλεύεται κάθε μη ασφαλή χρήση της εφαρμογής για να τη συσχετίσει με την διεύθυνση IP του συγκεκριμένου χρήστη Tor. Μία μέθοδος επίθεσης εξαρτάται από τον έλεγχο ενός κόμβο εξόδου ή την υποκλοπή της απάντησης ενός ανιχνευτή, ενώ μία δεύτερη μέθοδος επίθεσης βασίζεται εν μέρει στην στατιστική εκμετάλλευση της ανίχνευσης του κατανεμημένου πίνακα κατακερματισμού. Επίσης, τον Οκτώβριο του 2011 ερευνητική ομάδα από την Esiea, Γαλλική σχολή μηχανολόγων, δήλωσε ότι ανακάλυψε έναν τρόπο να υπονομεύσει το δίκτυο Tor με το να αποκρυπτογραφήσει επικοινωνίες που το διαπερνούν. Η τεχνική που περιέγραψαν απαιτεί τη δημιουργία ενός χάρτη των κόμβων του δικτύου Tor, τον έλεγχο του από κάποιον τρίτο και στην συνέχεια την απόκτηση των κλειδιών κρυπτογράφησης και των πηγών του αλγόριθμου. Ύστερα, χρησιμοποιώντας τα γνωστά πλέον κλειδιά και τις πηγές θεωρούν ότι έχουν την ικανότητα να αποκρυπτογραφούν δύο από τα τρία στρώματα κρυπτογράφησης. Ισχυρίζονται ότι μπορούν να σπάσουν το τρίτο κλειδί με μια επίθεση που βασίζεται στην στατιστική ανάλυση. Για να επανακατευθύνουν την κίνηση του Tor στους κόμβους που ελέγχουν, χρησιμοποίησαν μεθόδους επίθεσης άρνησης εξυπηρέτησης και επίθεσης packet spinning.

Τέλος, αξίζει να σημειωθεί ότι το Tor, όπως και όλα τα πρωτόκολλα που βασίζονται σε mix-nets, δεν παρέχουν προστασία από επιθέσεις παρακολούθησης και ανάλυσης της κίνησης. Για παράδειγμα, ένας παθητικός καθολικός παρατηρητής (global passive observer) θα μπορούσε να παραβιάσει όλες τις ιδιότητες της ανωνυμίας που παρέχει το Tor με πολύ μεγάλη πιθανότητα, χρησιμοποιώντας πληροφορίες όπως το χρόνο αποστολής των μηνυμάτων, το χρόνο λήψης των μηνυμάτων, το φορτίο των διακομιστών OR κ.ο.κ

2.2.7 Torsk

Το πρωτόκολλο *Torsk* [17] υλοποιήθηκε το 2009 από τους J. McLachlan, A. Tran και N. Hopper. Πρόκειται για ένα δομημένο *peer-to-peer (P2P)* πρωτόκολλο μικρής καθυστέρησης. Το Torsk έχει σχεδιαστεί ως ένας λειτουργικός αντικαταστάτης της επιλογής αναμετάδοσης και της υπηρεσίας καταλόγου του δικτύου Tor, καθώς μειώνει το κόστος αναμετάδοσης και συντήρησης, χωρίς ωστόσο να προκύπτουν νέες επιθέσεις και επιπλέον καθυστέρηση στις συνδέσεις.

Σε αντίθεση με τα άλλα P2P συστήματα ανωνυμίας, το Torsk δεν απαιτεί από όλους τους χρήστες να μεταδίδουν κίνηση για άλλους και χρησιμοποιεί έναν μηχανισμό αναζήτησης P2P με συμπληρωματικές δυνατότητες, προκειμένου να αποφύγει επιθέσεις σχετικά με την εμπιστευτικότητα και την ακεραιότητα των αναζητήσεων.

Το Torsk με την υλοποίησή του προσπαθεί να καλύψει κάποια αδύναμα σημεία του Tor, αλλά και των P2P προγραμμάτων ανωνυμίας. Όμως, δεν καταφέρνει να διασφαλίσει την ανωνυμία στην επικοινωνία του αποστολέα με τον παραλήπτη, που εμφανίζει στα onion routing πρωτόκολλα, καθώς και την προστασία των χρηστών από τους παθητικούς παρατηρητές που μπορούν να ανακαλύπτουν την προέλευση των πακέτων.

2.2.8 Hordes

Το *Hordes* [18] είναι ένα *πρωτόκολλο πολλαπλής διανομής (multicast)* που παρέχει ανωνυμία του αποστολέα στο Διαδίκτυο. Προτάθηκε το 2002 από τους B. Levine και C. Shields, και χρησιμοποιεί μηχανισμούς παρόμοιους με εκείνους που εφαρμόζονταν στα προηγούμενα πρωτόκολλα για την αποστολή δεδομένων. Όμως, στο Hordes εμφανίζεται για πρώτη φορά η χρήση της πολλαπλής δρομολόγησης για τη λήψη ανώνυμων δεδομένων.

Το πρωτόκολλο Hordes χρησιμοποιεί πολλαπλούς διακομιστές διαμεσολάβησης (proxies), παρόμοια με το πρωτόκολλο Crowds, ώστε να δρομολογήσει ένα πακέτο προς τον παραλήπτη. Ωστόσο, γίνεται χρήση της multicast επικοινωνίας για την αντίστροφη πορεία των ανώνυμων συνδέσεων, με αποτέλεσμα την παροχή πρόσθετης προστασίας του αποστολέα. Η ασυμμετρία των εμπρός και των ανάστροφων μονοπατιών του πρωτοκόλλου δημιουργεί μία πρόκληση για την παροχή των TCP υπηρεσιών, δεδομένου ότι η προωθημένη διαδρομή έχει μεγαλύτερη καθυστέρηση και πιθανότητα απώλειας κάποιου πακέτου, σε σχέση με την ανάστροφη.

Σε γενικές γραμμές, παρατηρήθηκε ότι ο σχεδιασμός του πρωτοκόλλου Hordes οδήγησε σε μικρότερες καθυστερήσεις μετάδοσης και απαιτεί λιγότερη εργασία στην επεξεργασία των μηνυμάτων από τους συμμετέχοντες, παρέχοντας ταυτόχρονα την ανωνυμία σε παρόμοιο βαθμό με τα πρωτόκολλα Crowds και Onion Routing. Όμως, η συγκεκριμένη υλοποίηση δεν διασφαλίζει την ανωνυμία του παραλήπτη και την μη συνδεσιμότητα των συμμετέχοντων στην επικοινωνία χρηστών.

2.2.9 HORNET

Το πρωτόκολλο *HORNET* [19] παρουσιάστηκε το 2015 από τους Chen, Asoni, Barrera, Danezi και Perrig. Πρόκειται για ένα σύστημα που επιτρέπει υψηλής ταχύτητας end-to-end ανώνυμα κανάλια. Το HORNET έχει υλοποιηθεί ως ένα onion σύστημα δρομολόγησης χαμηλής καθυστέρησης που λειτουργεί στο επίπεδο δικτύου, επιτρέποντας έτσι ένα ευρύ φάσμα εφαρμογών. Το συγκεκριμένο σύστημα χρησιμοποιεί συμμετρική κρυπτογραφία μόνο για τη διαβίβαση των δεδομένων, χωρίς να απαιτείται καμία αναφορά ανά ροή στους ενδιάμεσους κόμβους. Αυτό έχει ως αποτέλεσμα, οι κόμβοι του πρωτοκόλλου να επεξεργάζονται ανώνυμη κίνηση πάνω από 93 Gb/s.

Συγκρίνοντας το πρωτόκολλο HORNET με το πιο διαδεδομένο εργαλείο onion routing, το Tor, παρατηρείται ότι η κατασκευή των υποδομών του HORNET είναι ικανή να παρέχει μεγαλύτερες ταχύτητες στους κόμβους και είναι πιο ανθεκτικό στις απόπειρες υποκλοπής. Βασική διαφορά τους είναι ότι το HORNET χρησιμοποιεί δύο διαφορετικά onion πρωτόκολλα, ένα για την προστασία της ανωνυμίας των αιτημάτων των χρηστών στο Διαδίκτυο και το δεύτερο αποτελεί την τροποποιημένη έκδοση του Tor ώστε να επικοινωνεί με επιτυχία με μία τοποθεσία του δικτύου HORNET.

Το HORNET αποτελεί ένα από τα σπουδαιότερα πρωτόκολλα σήμερα, διότι παρέχει τη δυνατότητα γρήγορης ανώνυμης επικοινωνίας. Όμως, και αυτό όπως και οι προκάτοχοί του, δεν μπορούν να προστατέψουν τους χρήστες από επιθέσεις ετικετών (tagging attacks).

2.2.10 Crowds

Το πρωτόκολλο *Crowds* [20] σχεδιάστηκε από τους Reiter και Rubin το 1998, με σκοπό την προστασία των χρηστών του συστήματος από τους εσωτερικούς επιτιθεμένους και ένα διεφθαρμένο παραλήπτη και την παροχή ενός μηχανισμού για ανώνυμη περιήγηση στο Web. Το Crowds εισήγαγε για πρώτη φορά την έννοια της ανάμειξης (blending) των χρηστών σε ένα πλήθος υπολογιστών, κάτι που χρησιμοποιήθηκε σε μετέπειτα πρωτόκολλα. Η βασική ιδέα του πρωτοκόλλου Crowds είναι η απόκρυψη των επικοινωνιών του κάθε χρήστη, δρομολογώντας τις τυχαία σε μία ομάδα παρόμοιων χρηστών. Το πρωτόκολλο Crowds έχει υλοποιηθεί με τέτοιο τρόπο που καθιστά δύσκολο για κάποιο κακόβουλο μέλος της ομάδας να γνωρίζει αν ένας χρήστης είναι ο πραγματικός αποστολέας ή απλά δρομολογεί το μήνυμα κάποιου άλλου χρήστη. Επιπλέον, το σύστημα Crowds σχεδιάστηκε επίσης με σκοπό να προσφέρει ασφάλεια από τα συνεργαζόμενα μέλη του πρωτοκόλλου, καθώς και τον τελικό διακομιστή.

Κατά την πρόσβαση στο Διαδίκτυο, ένας χρήστης αρχικά εισέρχεται σε ένα σύστημα crowd που περιλαμβάνει και άλλους χρήστες. Το αίτημα του πελάτη προς ένα διακομιστή Web λαμβάνεται αρχικά από

ένα τυχαίο μέλος της ομάδας. Το συγκεκριμένο μέλος μπορεί είτε να υποβάλλει το αίτημα απευθείας στον τελικό διακομιστή, είτε να το προωθήσει σε κάποιο άλλο μέλος. Η ίδια διαδικασία επαναλαμβάνεται και στα επόμενα μέλη, με αποτέλεσμα το αρχικό αίτημα του πελάτη να υποβάλλεται από κάποιο τυχαίο μέλος του συστήματος στο διακομιστή. Αυτός ο σχεδιασμός αποτρέπει έναν εισβολέα να αντιληφθεί τον αρχικό αποστολέα, καθώς ο τελευταίος δε διαφέρει από ένα άλλο μέλος που αναλαμβάνει να διαβιβάσει ένα αίτημα.

Συγκρίνοντας το πρωτόκολλο Crowds με τα συστήματα που στηρίζονται στη χρήση κάποιου ενδιάμεσου διαμεσολαβητή, όπως το Anonymizer, παρατηρείται ότι το πρώτο παρέχει προστασία ενάντια σε ένα ευρύτερο φάσμα επιθέσεων. Ειδικότερα, τα συστήματα διαμεσολάβησης είναι ευάλωτα σε παθητικές επιθέσεις με σκοπό την απόκτηση του ελέγχου του proxy.

Ένα μήνυμα αρχικοποίησης δρομολογείται αρχικά από τον αποστολέα σε μία σειρά κόμβων στο πλήθος, προκειμένου να σχηματιστεί μία διαδρομή για όλα τα μελλοντικά μηνύματα που θα σταλθούν από τον συγκεκριμένο αποστολέα. Κάθε φορά που ένα μέλος του πλήθους λαμβάνει ένα αίτημα από κάποιον άλλο κόμβο επιλέγει τυχαία είτε να το διαβιβάσει σε άλλο τυχαίο μέλος του συστήματος είτε να το υποβάλλει στον διακομιστή. Τις περισσότερες φορές, η τυχαία επιλογή τείνει υπέρ της προώθησης των μηνυμάτων. Δηλαδή, υπάρχει μία παράμετρος στο σύστημα η οποία ορίζει μεγαλύτερη πιθανότητα στη διαβίβαση και η οποία καθορίζει το επίπεδο της ανωνυμίας στο σύστημα. Η διαδρομή που έχει οριστεί διατηρείται για περιορισμένο χρονικό διάστημα, μετά το πέρας του οποίου πρέπει να αναμορφωθεί όπως και όλες οι υπόλοιπες.

Όπως περιγράφηκε και παραπάνω, το πρωτόκολλο Crowds παρέχει ανωνυμία της επικοινωνίας, όταν ο εισβολέας είναι εσωτερικός χρήστης του συστήματος (ο τελικό διακομιστής ή κάποιος από τους κόμβους). Επίσης, τα συγκεκριμένο πρωτόκολλο προσφέρει ανωνυμία του παραλήπτη εναντίον ωτακουστών του συστήματος που καταφέρνουν να παρατηρήσουν όλη την κίνηση του μηχανήματος του χρήστη. Αντίθετα, το Crowds δεν μπορεί να προσφέρει ανωνυμία απέναντι σε εξωτερικούς κακόβουλους χρήστες. Η ανωνυμία του αποστολέα βασίζεται στην παραδοχή ότι τα μέλη του συστήματος δεν γνωρίζουν αν ο προηγούμενος κόμβος που τους απέστειλε το μήνυμα είναι ο αρχικός ή κάποιος ενδιάμεσος. Επίσης, είναι αποδεδειγμένο ότι τα συστήματα Crowds είναι ευάλωτα σε *επιθέσεις predecessor*, καθώς αν ένας κόμβος ζητά επανειλημμένα ένα συγκεκριμένο πόρο, τελικά μπορεί να συνδεθούν.

2.2.11 Buses – Taxis

Η ιδέα της περιγραφής της κυκλοφορίας των μηνυμάτων στο δίκτυο ως ένα λεωφορείο αναπτύχθηκε το 2003 από τους Beimel και Dolev [21]. Κάθε κομμάτι πληροφορίας καταλαμβάνει ένα κάθισμα στο λεωφορείο. Τα δρομολόγια έχουν επιλεχθεί και τα λεωφορεία διασχίζουν αυτές τις διαδρομές είτε σε συγκεκριμένα είτε σε τυχαία χρονοδιαγράμματα. Δεδομένου ότι τα λεωφορεία διασχίζουν το δίκτυο σε συγκεκριμένες διαδρομές, ο εισβολέας δεν μπορεί να ανακαλύψει ποιοι κόμβοι επικοινωνούν.

Λόγω του ότι ο χρόνος και η πολυπλοκότητα της επικοινωνίας είναι δύο αντικρουόμενα ζητήματα σχεδιασμού, οι Beimel και Dolev πρότειναν δύο πρωτόκολλα: ένα για τη βελτιστοποίηση της επικοινωνίας και ένα για αυτή του χρόνου. Στο πρωτόκολλο βέλτιστης επικοινωνίας, μόνο ένας κόμβος στέλνει μήνυμα σε ένα άλλο σε κάθε μονάδα χρόνου. Αυτό σημαίνει ότι σε κάθε χρονική στιγμή μόνο ένα λεωφορείο διασχίζει το γράφημα επικοινωνίας. Το κάθε κάθισμα περιέχει ένα μήνυμα που στέλνει ένας κόμβος σε κάποιον άλλο, κρυπτογραφημένο με το δημόσιο κλειδί του δεύτερου ή με ένα συμμετρικό κλειδί που έχουν ανταλλάξει.

Κάθε φορά που το λεωφορείο σταματά σε κάποιον κόμβο, αυτός ελέγχει ποια μηνύματα προορίζονται για αυτόν, αποκρυπτογραφώντας τα και αγνοώντας αυτά που περιέχουν ψεύτικη πληροφορία. Στη συνέχεια, τοποθετεί το μήνυμα που θέλει να αποστείλει στο κάθισμά του ή ψεύτικη πληροφορία αν δε θέλει να επικοινωνήσει με άλλον κόμβο. Με αυτό τον τρόπο, ένας επιτιθέμενος στο σύστημα δεν μπορεί να προσδιορίσει αν η πληροφορία που υπάρχει σε κάθε κάθισμα είναι πραγματική και εάν δύο κόμβοι επικοινωνούν.

Στο πρωτόκολλο βέλτιστου χρόνου, η χρονική πολυπλοκότητα είναι η απόσταση μεταξύ των δύο κόμβων. Στον συγκεκριμένο σχεδιασμό, δύο λεωφορεία ταξιδεύουν από κόμβο σε κόμβο σε διάφορες κατευθύνσεις. Οι κόμβοι μεταφέρουν τις θέσεις από το ένα λεωφορείο στο άλλο, με βάση το κριτήριο της συντομότερης διαδρομής. Και στην παρούσα υλοποίηση προστατεύεται η ασφάλεια της επικοινωνίας,

καθώς τα μηνύματα είναι κρυπτογραφημένα και αποστέλλονται εικονικά μηνύματα αν δεν υπάρχει πραγματική επικοινωνία μεταξύ δύο κόμβων.

Στο αρχικό πρωτόκολλο Buses, το μέγεθος του λεωφορείου, καθώς και η ανάγκη να γεμίσουν όλα τα καθίσματα για να ξεκινήσει η δρομολόγηση, κάνουν το σύστημα ακατάλληλο για επικοινωνίες πραγματικού χρόνου. Έτσι, έχουν αναπτυχθεί κατά καιρούς διάφορες παραλλαγές του αρχικού πρωτοκόλλου. Το 2005, οι Hirt, Jacobson και Williamson [22] παρουσιάζουν μία πρακτική εφαρμογή της αρχιτεκτονικής Buses, στην οποία κάθε χρήστης έχει συγκεκριμένο αριθμό θέσεων στο λεωφορείο. Όμως, η συγκεκριμένη υλοποίηση δεν καταφέρνει να διορθώσει τις καθυστερήσεις στην επικοινωνία που είχε και η αρχική. Το 2008, οι ίδιοι δημιουργοί παρουσιάζουν το πρωτόκολλο Taxis [23], μία ακόμα παραλλαγή του Buses. Στο συγκεκριμένο, η δρομολόγηση γίνεται με ταξί αντί για λεωφορεία. Το σύστημα αυτό παρουσιάζει λιγότερη καθυστέρηση σε σχέση με τους προκατόχους του, χάνοντας όμως σημαντικό έδαφος στην ανωνυμία των χρηστών λόγω της μείωσης του αριθμού των καθισμάτων.

2.2.12 Drunk Motorcyclist

Το πρωτόκολλο *Drunk Motorcyclist (DM)* [24] βασίζεται στα πρωτόκολλα Buses (και Taxis) και επιτρέπει την ανώνυμη επικοινωνία χωρίς σύνδεση. Παρουσιάστηκε το 2014 από τους A. Young και M. Yung. Η διαφορά του DM πρωτοκόλλου από τα προηγούμενα είναι πως αντί να χρησιμοποιούνται λεωφορεία και ταξί που έχουν πολλαπλά καθίσματα, η μονάδα μεταφοράς είναι μία μοτοσυκλέτα που έχει μόνο ένα κάθισμα. Το κάθισμα περιέχει μόνο ένα κρυπτογράφημα και ένα χρονικό διάστημα που παραμένει ενεργό (*time-to-live – TTL*). Η αναλογία είναι η εξής: ο μοτοσικλετιστής είναι μεθυσμένος και ανεβαίνει στη μοτοσυκλέτα του. Σε κάθε κόμβο του γραφήματος, ο οποίος ονομάζεται σταθμός, ο αναβάτης διαλέγει τυχαία έναν γειτονικό κόμβο και κάνει βόλτες σε αυτόν. Μετά από έναν καθορισμένο αριθμό σταθμών, η μοτοσυκλέτα διαλύεται. Με άλλα λόγια, το κρυπτογράφημα εκτελεί μία τυχαία διαδρομή συγκεκριμένου μήκους στο γράφημα, πριν αφαιρεθεί λόγω λήξης του TTL. Ο κόμβος, ο οποίος σταματά να λειτουργεί, χάνει τις επόμενες μοτοσυκλέτες αλλά δεν διαταράσσει τα ταξίδια των άλλων μοτοσικλετών στο δίκτυο. Έτσι, το πρωτόκολλο των μεθυσμένων μοτοσικλετιστών επιδεικνύει ανοχή σε σφάλματα, σε αντίθεση με την τοπολογία των λεωφορείων. Σε αυτή διατίθεται ένα λεωφορείο με συγκεκριμένες θέσεις, οπότε στην περίπτωση σφάλματος, η επικοινωνία διακόπτεται.

Το DM πρωτόκολλο απαιτεί υπολογιστική ασφάλεια της εμπιστευτικότητας και της ανωνυμίας. Παράλληλα, αποδεικνύεται ότι η συγκεκριμένη υλοποίηση μετριάζει τις εξωτερικές ενεργές επιθέσεις. Όμως, το Drunk Motorcyclist πρωτόκολλο προσφέρει ανωνυμία στην επικοινωνία χωρίς σύνδεση, με αποτέλεσμα να μην μπορεί να χρησιμοποιηθεί σε συστήματα πραγματικού χρόνου.

2.2.13 Tarzan

Το πρωτόκολλο *Tarzan* [25] σχεδιάστηκε το 2002 από τους Freedman και Morris. Πρόκειται για ένα peer-to-peer (P2P) ανώνυμο δίκτυο επικάλυψης της IP διεύθυνσης. Ο αποστολέας ενός μηνύματος επιλέγει ψευδο-τυχαία ένα μονοπάτι από κόμβους, με τέτοιο τρόπο που κάποιος εισβολέας δεν μπορεί να επιδράσει εύκολα. Επιτυγχάνει την ανωνυμία του με τη εγκαθίδρυση onion πολυεπίπεδης κρυπτογραφημένης σύνδεσης που αναπαράγεται μέσω μιας αλληλουχίας των ενδιάμεσων κόμβων. Κάθε ενδιάμεσος κόμβος λειτουργεί σαν ένα mix για τους υπόλοιπους.

Το πρωτόκολλο Tarzan χρησιμοποιεί μία περιορισμένη τοπολογία δικτύου για τη δρομολόγηση των πακέτων. Κάθε κόμβος διατηρεί μόνιμες συνδέσεις με ένα μικρό σύνολο από άλλους κόμβους, που ονομάζεται mimics. Οι διαδρομές των ανώνυμων μηνυμάτων δημιουργούνται μόνο μέσω και μεταξύ των mimics, προκειμένου να αποφευχθεί η ανάλυση της κυκλοφορίας από κάποιον κακόβουλο χρήστη. Έτσι, υπάρχει αναλογική σχέση μεταξύ της αποτελεσματικότητας και τις ασφάλειας.

Επιπλέον, το Tarzan επιτρέπει στους χρήστες που συμμετέχουν στο δίκτυο να επικοινωνούν με διακομιστές Internet που βρίσκονται εκτός του πρωτοκόλλου διαμέσου ειδικών IP σηράγγων. Τα δύο άκρα της σήραγγας είναι ένας κόμβος Tarzan που τρέχει μία εφαρμογή-πελάτη και ένας κόμβος που τρέχει έναν μεταφραστή διεύθυνσης δικτύου (*Network Address Translator – NAT*), ο οποίος είναι υπεύθυνος για την προώθηση της κυκλοφορίας του πελάτη στον τελικό προορισμό του Διαδικτύου.

Το πρωτόκολλο Tarzan λειτουργεί σε τρία στάδια. Αρχικά, ένας κόμβος που τρέχει μία εφαρμογή-πελάτη επιλέγει ένα σύνολο κόμβων και θεσπίζει ένα μονοπάτι δρομολόγησης μέσω του δικτύου επικάλυψης. Στη συνέχεια, ο αρχικός κόμβος δρομολόγησης εγκαθιδρύει μία σήραγγα που χρησιμοποιεί τους παραπάνω κόμβους. Τέλος, καθοδηγεί τα πακέτα δεδομένων μέσω αυτής της σήραγγας. Μία από τις αδυναμίες ασφάλειας του Tarzan είναι ότι η επιλογή των γειτονικών κόμβων για την υλοποίηση του mimic γίνεται με βάση κάποιο αναγνωριστικό δικτύου ή τη διεύθυνση, με αποτέλεσμα να είναι ευάλωτο σε spoofs.

Σε μία πρώτη έκδοση του Tarzan, κάθε κόμβος είναι υποχρεωμένος να γνωρίζει ένα τυχαίο υποσύνολο άλλων κόμβων στο P2P δίκτυο. Δεδομένου ότι το δίκτυο είναι αρκετά μεγάλο, οι κόμβοι παρουσιάζουν μεγάλο ποσοστό αποσυνδέσεων, με αποτέλεσμα στο τέλος κάθε κόμβος να γνωρίζει μόνο ένα μικρό υποσύνολο από άλλους κόμβους. Ένας εισβολέας μπορεί να διακρίνει τον κόμβο-πηγή της σύνδεσης με πολύ μεγάλη πιθανότητα εάν διαφθείρει κάποιο κόμβο και λάβει γνώση του προηγούμενου και του επόμενού του. Αυτό το πρόβλημα έχει διορθωθεί στην τελική έκδοση του πρωτοκόλλου, απαιτώντας κάθε κόμβος να γνωρίζει όλους τους άλλους. Ωστόσο, η τελική έκδοση είναι σαφώς λιγότερο πρακτική.

2.2.14 MorphMix

Το *MorphMix* [26] αποτελεί άλλο ένα P2P σύστημα για ανώνυμη χρήση του Διαδικτύου. Αναπτύχθηκε το 2002 από τους Rennhard και Plattner. Η αρχιτεκτονική και οι απειλές του συστήματος MorphMix είναι παρόμοιες με το σύστημα Tarzan. Μια κρίσιμη διαφορά τους είναι ότι στο σύστημα Tarzan, η διαδρομή καθορίζεται από την πηγή, ενώ στο MorphMix η διαδρομή επιλέγεται από τους ενδιάμεσους κόμβους. Ο αποστολέας επιλέγει μόνο τον πρώτο ενδιάμεσο κόμβο. Κάθε κόμβος κατά μήκος της ανώνυμης σήραγγας επιλέγει τον επόμενο κόμβο. Το πλεονέκτημα αυτού του σχεδιασμού είναι ότι κάθε κόμβος έχει να διαχειριστεί μόνο το τοπικό περιβάλλον του, το οποίο είναι ανεξάρτητο του συνολικού μεγέθους του συστήματος.

Το MorphMix είναι ένα Mix-net σε επίπεδο εφαρμογής πρωτόκολλο που χρησιμοποιεί αρχιτεκτονική TCP μεταξύ των mixes. Όπως έχει υλοποιηθεί, κάθε συμμετέχων αποτελεί και ένα mix ταυτόχρονα. Με άλλα λόγια, όλοι οι συμμετέχοντες είναι peers. Το σύνολο των mixes είναι ένα δυναμικό σύστημα από αναξιόπιστους κόμβους που μπορούν να ενταχθούν και να αποχωρήσουν ανά πάσα στιγμή.

Στο πρωτόκολλο MorphMix κάθε κόμβος δεν γνωρίζει εάν ο προηγούμενος κόμβος είναι ο αποστολέας ή αν απλά έχει το ρόλο του κόμβου αναμετάδοσης. Ως εκ τούτου, το πρωτόκολλο μπορεί να παρέχει δυνατότητα άρνησης (deniability). Λόγω της δυνατότητας που παρέχεται στους ενδιάμεσους κόμβους να επιλέξουν τον αμέσως επόμενο τους στο δίκτυο, τα MorphMix συστήματα θεωρούνται ευάλωτα σε *επιθέσεις συνεργίας (colluding attacks)*, καθώς οι κακόβουλοι κόμβοι που υπάρχουν στο σύστημα θα δημιουργήσουν ένα μονοπάτι με συνεργαζόμενους κόμβους. Για την αποφυγή της συγκεκριμένης επίθεσης, το MorphMix χρησιμοποιεί έναν μηχανισμό ανίχνευσης των συμπαιγνιών για να ανιχνεύσει ανάρμοστη συμπεριφορά στο δίκτυο. Αυτό αποτρέπει τις επιθέσεις στο δίκτυο, αλλά δεν παρέχει σε καμία περίπτωση ασφάλεια.

2.3 Broadcast / Άλλα συστήματα

Τα συστήματα που βασίζονται στη μετάδοση θεωρούνται από τα πιο ασφαλή απέναντι τις ενεργές επιθέσεις ανάλυσης της κυκλοφορίας. Στα συστήματα αυτά, το μήνυμα δεν αποστέλλεται ανάμεσα στους κόμβους, αλλά μεταδίδεται στο δίκτυο και κάθε κόμβος μπορεί να το παραλάβει.

2.3.1 P^5

Το πρωτόκολλο *P^5* προτάθηκε το 2002 από τον Sherwood και άλλους [27] με σκοπό τη διασφάλιση ανώνυμων επικοινωνιών μέσω του Διαδικτύου. Αποτελεί ένα σύστημα που στοχεύει να προσφέρει ανωνυμία στον αποστολέα, στον παραλήπτη, αλλά και στη μεταξύ τους επικοινωνία. Το όνομά του προήλθε από το *Peer-to-Peer Personal Privacy Protocol (Ομότιμο Προσωπικό Πρωτόκολλο Προστασίας)*. Χρησιμοποιεί δέντρα-μεταδότες για να επιτύχει ανωνυμία. Το P^5 προσφέρει στον κάθε χρήστη του πρωτοκόλλου τη δυνατότητα να ορίσει μία ισορροπία μεταξύ του βαθμού της ανωνυμίας και της απόδοσης της επικοινωνίας, κάτι το οποίο μπορεί να χρησιμοποιηθεί για την υλοποίηση επεκτάσιμων ομάδων.

Το P^5 δημιουργεί μία ιεραρχία στη μετάδοση, έτσι ώστε διαφορετικά επίπεδα της ιεραρχίας να παρέχουν διαφορετικά επίπεδα ανωνυμίας, με γνώμονα το κόστος του εύρους της επικοινωνίας και την αξιοπιστία. Οι χρήστες του συστήματος επιλέγουν τοπικά ένα επίπεδο ανωνυμίας και αποδοτικότητας της επικοινωνίας, βασιζόμενοι στην αναμενόμενη απόδοση που θέλουν να επιτύχουν. Για το σκοπό αυτό κάθε χρήστης επιλέγει μία μάσκα (mask) η οποία ορίζει το βαθμό πολυεκπομπής που θα χρησιμοποιήσει ο συγκεκριμένος χρήστης.

Η λογική ιεραρχία μετάδοσης των P^5 συστημάτων είναι ένα δυαδικό δέντρο που κατασκευάστηκε χρησιμοποιώντας τα δημόσια κλειδιά του κάθε χρήστη. Κάθε κόμβος του δέντρου αποτελείται από μία συμβολοσειρά συγκεκριμένου μήκους, που εκπροσωπεί το επίπεδο ιεραρχίας και την ομάδα της. Όλοι οι χρήστες, για να επικοινωνήσουν, στέλνουν συγκεκριμένου μήκους μηνύματα σε συγκεκριμένη αναλογία. Τα μηνύματα μπορεί να είναι πραγματική πληροφορία ή θόρυβος. Στην περίπτωση που οι χρήστες θέλουν πραγματικά να επικοινωνήσουν, το μήνυμα που στέλνεται είναι κρυπτογραφημένο με το δημόσιο κλειδί του παραλήπτη. Κάθε χρήστης του συστήματος που θα λάβει το κρυπτογράφημα θα προσπαθήσει να το αποκρυπτογραφήσει με το ιδιωτικό του κλειδί.

Παρόλο που το πρωτόκολλο P^5 είναι αρκετά επεκτάσιμο να υποστηρίξει μεγάλο αριθμό χρηστών, χρειάζεται ένα κανάλι μετάδοσης που θα αποτελεί τη ρίζα του συστήματος. Επίσης, παρατηρείται πως όταν ο αριθμός των ενεργών χρηστών ξεπεράσει ένα συγκεκριμένο όριο, υπάρχουν πολλές απώλειες πακέτων που αποστέλλονται.

2.3.2 BitMessage

Το *BitMessage* [28] είναι ένα πρωτόκολλο P2P επικοινωνίας που χρησιμοποιείται για την ανταλλαγή κρυπτογραφημένων μηνυμάτων μεταξύ των χρηστών μέσα από ένα κανάλι μετάδοσης. Δεν στηρίζεται σε κάποιο κεντρικό διακομιστή, με αποτέλεσμα οι χρήστες του συστήματος να μην χρειάζεται να εμπιστεύονται κάποια τρίτη οντότητα για την εξασφάλιση της επικοινωνίας τους. Χρησιμοποιεί ισχυρό έλεγχο ταυτότητας, που σημαίνει ότι ο αποστολέας ενός μηνύματος δεν μπορεί να πλαστογραφηθεί, και στοχεύει να διασφάλιση της ανωνυμίας του αποστολέα και του παραλήπτη από παθητικούς ωτακουστές του συστήματος.

Ως προς την υλοποίησή του, αρχικά ένας χρήστης δημιουργεί ένα μήνυμα, το οποίο το κρυπτογραφεί με το δημόσιο κλειδί του παραλήπτη. Στη συνέχεια, το μεταδίδει στο δίκτυο, με αποτέλεσμα να το λάβουν όλοι οι χρήστες. Έπειτα, ο κάθε χρήστης χρησιμοποιεί το ιδιωτικό του κλειδί για να το αποκρυπτογραφήσει και μόνο ο σωστός παραλήπτης μπορεί να δει το πραγματικό μήνυμα που στάλθηκε.

Το πρωτόκολλο BitMessage παρέχεται για την ανταλλαγή ανώνυμων e-mails και δεν προτείνεται για συστήματα πραγματικού χρόνου που απαιτούν χαμηλής καθυστέρησης επικοινωνία.

2.3.3 Cocaine Auction Protocol

Το πρωτόκολλο *Cocaine Auction* [29] περιγράφηκε από τους Stajano F. Και Anderson R. το 1999. Η παρούσα υλοποίηση αποτελεί μία αποτελεσματική τεχνική ανώνυμης μετάδοσης και στηρίζεται στη διαδικασία της δημοπρασίας. Η επικοινωνία πραγματοποιείται μέσω Ethernet ή ασύρματα. Επομένως, το συγκεκριμένο πρωτόκολλο προορίζεται για τοπικά περιβάλλοντα δικτύωσης και δεν είναι πρακτικό για επικοινωνία μεγαλύτερης κλίμακας. Επίσης, παρόλο που παρέχει πλήρη ανωνυμία, είναι επιρρεπές σε επιθέσεις στο κανάλι.

2.3.4 ISDN-mixes

Τα *ISDN-mixes* [30] προτάθηκαν το 1991 με σκοπό την πρόληψη της ανίχνευσης της επικοινωνίας σε ISDN δίκτυα με μικρό εύρος (bandwidth). Τα ISDN-mixes συνδυάζουν την αρχιτεκτονική των mixes που παρουσίασε ο Chaum, της εικονικής κίνησης στα κανάλια επικοινωνίας των χρηστών και της μετάδοσης (broadcast) όλων των εισερχόμενων συνδέσεων στην περιοχή του χρήστη. Βασίζεται στα κανάλια επικοινωνίας (mix channels), ώστε να διασφαλίζεται ότι το κάθε μήνυμα βρίσκεται σε επεξεργασία από όλα τα mixes την ίδια χρονική στιγμή. Κάθε mix channel αποτελείται από ένα *κανάλι αποστολής (mix-sending*

channel) του αποστολέα και ένα *κανάλι λήψης (mix-receiving channel)* του παραλήπτη. Έτσι, το κάθε μισό του καναλιού αναλαμβάνει να προστατέψει την ανωνυμία του χρήστη που το χρησιμοποιεί.

Βασικό χαρακτηριστικό της συγκεκριμένης αρχιτεκτονικής είναι η διαίρεση της σύνδεσης σε μία ακολουθία χρονικών (*timeslide*) καναλιών. Σε κάθε ορισμένη χρονική στιγμή, οι χρήστες μπορούν να απελευθερώσουν κάποια σύνδεση και να δημιουργήσουν νέες. Ένας χρήστης που δεν χρειάζεται κανάλι επικοινωνία στη διάρκεια ενός timeslide, θεσπίζει ένα εικονικό. Κάτι τέτοιο δεν κοστίζει επιπλέον bandwidth γιατί το εικονικό κανάλι που εγκαθιδρύεται βρίσκεται στη γραμμή του συνδρομητή και μόνο.

2.3.5 Real-time MIXes

Τα *MIXes πραγματικού χρόνου (Real-time MIXes)* [31] αποτελούν τη γενίκευση της αρχιτεκτονικής των ISDN-mixes και χαρακτηρίζονται ως ένα αποδοτικού εύρους πρωτόκολλο ανωνυμίας με περιορισμούς πραγματικού χρόνου. Προτάθηκε από την Jerichow και άλλους το 1998 και παρουσιάζει ουσιαστικά τεχνικές για αποτελεσματική ανωνυμία στην επικοινωνία σε μικρής εμβέλειας ISDN δίκτυα.

Παρόμοια με τα ISDN-mixes, το συγκεκριμένο σύστημα βασίζεται σε mix-channels, που περιλαμβάνουν mix-sending channels που στοχεύουν στην ανωνυμία του αποστολέα και mix-receiving channels, που διασφαλίζουν τον παραλήπτη. Στο παρόν πρωτόκολλο, κάθε χρήστης είναι συνδεδεμένος σε ένα τοπικό κέντρο μέσω ενός αποκλειστικού καλωδίου, ενώ το εύρος ζώνης μοιράζεται σε μεγάλης απόστασης δίκτυο.

Και στη συγκεκριμένη κατηγορία, βασικό χαρακτηριστικό είναι η διαίρεση της σύνδεσης σε μία ακολουθία χρονικών (timeslide) καναλιών. Σε κάθε timeslide, οι χρήστες μπορούν να απελευθερώσουν κάποια σύνδεση και να δημιουργήσουν νέες, ή αν δε χρειάζονται να επικοινωνήσουν, θεσπίζεται ένα εικονικό κανάλι.

2.3.6 Riffle

Το σύστημα *Riffle* [32] παρουσιάστηκε το 2016 από τους A. Kwon, D. Lazar, S. Devadas, και B. Ford. Αποτελεί ένα πρωτόκολλο ανώνυμης επικοινωνίας που παρέχει προστασία κατά της ανάλυσης κυκλοφορίας και ισχυρή ανωνυμία, ελαχιστοποιώντας ταυτόχρονα το εύρος και τους υπολογιστικούς πόρους. Το Riffle αποτελείται από ένα μικρό σύνολο ανώνυμων διακομιστών και ένα μεγάλο αριθμό χρηστών, και εγγυάται ισχυρή ανωνυμία ανάμεσα σε ειλικρινείς πελάτες όσο υπάρχει ένας τουλάχιστον ειλικρινής διακομιστής. Η παρούσα αρχιτεκτονική χρησιμοποιεί ένα υβριδικό σύστημα ανακατέματος, το οποίο χρησιμοποιείται κατά την αποστολή των μηνυμάτων, και ένα σύστημα ανάκτησης ιδιωτικών πληροφοριών, το οποίο με τη σειρά του χρησιμοποιείται κατά την λήψη.

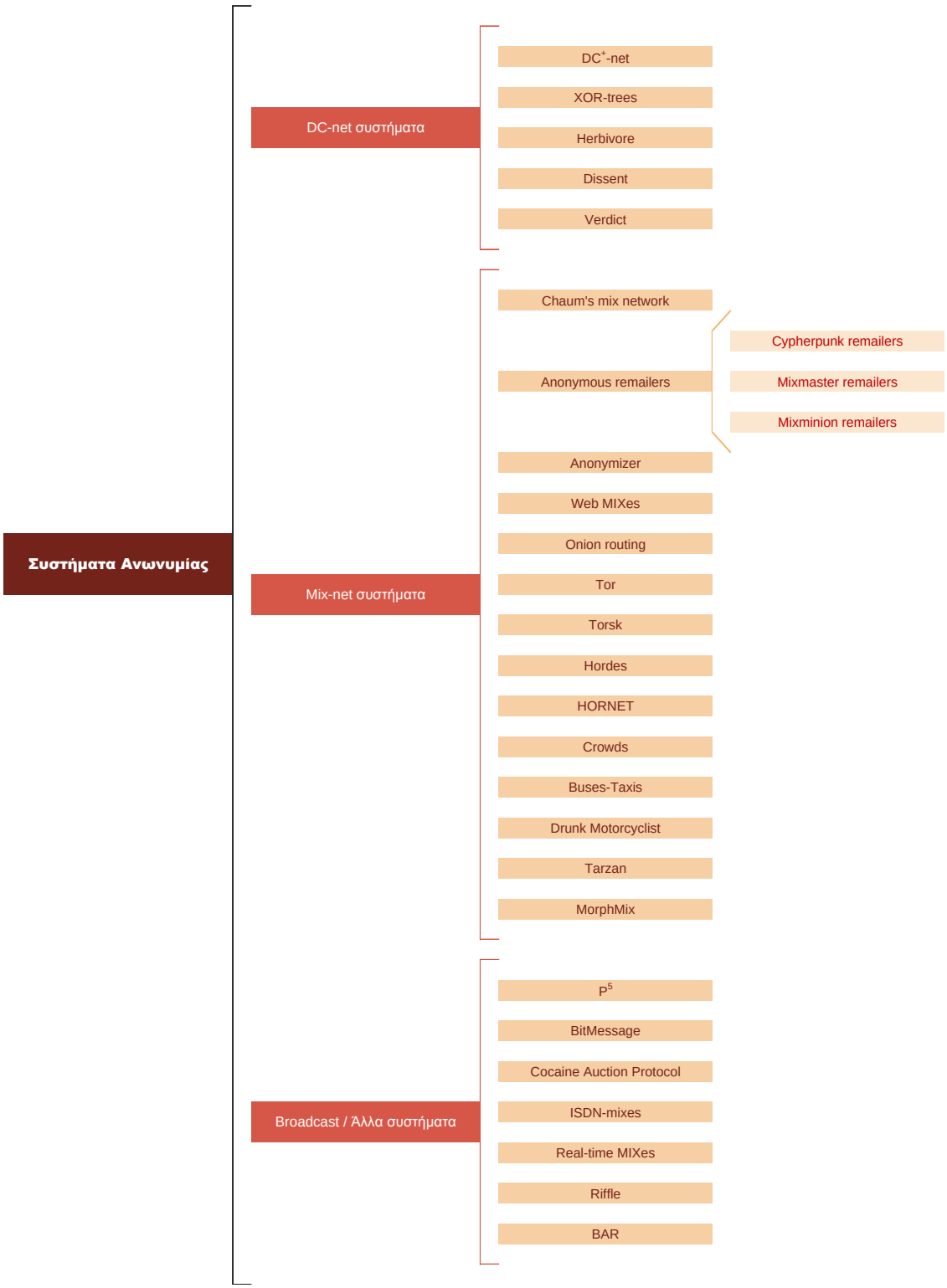
Το σύστημα Riffle χρησιμοποιείται για τον ασφαλή διαμοιρασμό αρχείων, καθώς και για υπηρεσίες *microblogging* (ένα είδος blogging με μικρότερο περιεχόμενο), και εξασφαλίζει ισχυρή ανωνυμία και καλό εύρος σε μεγάλο αριθμό χρηστών.

2.4 Σύγκριση των συστημάτων

	Anonymity properties			Real-time	Scalable	Distributed	Implemented
	Sender	Receiver	Sender-Receiver				
DC-net	✓	✓	✓	✗	✗	✓	✗
DC+-net	✓	✓	✓	✗	✗	✓	✗
XOR-trees	✓	✓	✓	✗	✗	✓	✓
Herbivore	✓	✓	✓	✗	✗	✓	✓
Dissent	✓	✓	✓	✗	✓	✓	✓
Verdict	✓	✓	✓	✓	✓	✓	✓
Mix net	✓	✓	✗	✗	✗	✓	✗
Cypherpunk	✓	✓	✓	✗	✓	✓	✓
Mixmaster	✓	✓	✓	✗	✓	✓	✓
Mixminion	✓	✓	✓	✗	✓	✓	✓
Anonymizer	✓	✗	✗	✓	✗	✗	✓
Web MIXes	✓	✓	✓	✓	✓	✓	✓
Onion routing	✓	✗	✗	✓	✓	✓	✓
Tor	✓	✓	✗	✓	✓	✓	✓
Torsk	✓	✓	✗	✓	✓	✓	✓
Hordes	✓	✗	✗	✓	✓	✓	✓
HORNET	✓	✓	✗	✓	✓	✓	✓
Crowds	✓	✗	✗	✓	✓	✗	✓
Buses & variations	✓	✓	✓	✗	✗	✓	✗
Drunk Motorcyclist	✓	✓	✓	✗	✓	✓	✗
Tarzan	✓	✓	✗	✓	✓	✓	✓
MorphMix	✓	✓	✗	✓	✓	✓	✓
P5	✓	✓	✓	✓	✓	✓	✓
BitMessage	✓	✓	✗	✗	✓	✓	✓
Cocaine auction	✓	✓	✓	✗	✗	✗	✗
ISDN-mixes	✓	✓	✗	✗	✓	✓	✗
Real-time MIXes	✓	✓	✗	✓	✓	✓	✗
Riffle	✓	✓	✗	✗	✓	✓	✓
BAR	✓	✓	✓	✓	✓	✓	✓

Εικόνα 2.2 Σύγκριση συστημάτων ανωνυμίας

2.4 Δενδροειδής απεικόνιση συστημάτων



Εικόνα 2.3 Δενδροειδής απεικόνιση συστημάτων ανωνυμίας

Κεφάλαιο 3

Περιγραφή πρωτοκόλλου BAR

Στο συγκεκριμένο κεφάλαιο θα γίνει μία αρχική παρουσίαση του πρωτοκόλλου ανώνυμης επικοινωνίας BAR [33], καθώς και της υφιστάμενης υλοποίησής του [34]. Θα περιγραφούν τα συστατικά, οι λειτουργίες του, καθώς και τα χαρακτηριστικά της ανωνυμίας που το διέπουν.

3.1 Εισαγωγή στο BAR

Το πρωτόκολλο *Broadcast Anonymous Routing (BAR)* [33] είναι ένα σύστημα που παρέχει ισχυρή ανωνυμία τόσο για τον αποστολέα όσο και για τον παραλήπτη του μηνύματος. Επίσης, καθιστά δύσκολο για κάποιον έντιμο άλλα και περίεργο αντίπαλο να διακρίνει αν δύο χρήστες BAR βρίσκονται πραγματικά σε επικοινωνία. Το BAR συνδυάζει τα χαρακτηριστικά των DC-net και Mix-net αρχιτεκτονικών. Για την διασφάλιση της ανωνυμίας του παραλήπτη, χρησιμοποιεί τα κανάλια εκπομπής (broadcast channels).

Συγκρίνοντας το BAR με παρόμοιες προσεγγίσεις Broadcast συστημάτων, παρατηρείται ότι το πρώτο έχει χαμηλότερο υπολογιστικό κόστος, λόγω ενός αποτελεσματικού μηχανισμού που επιτρέπει στους πελάτες να αποκρυπτογραφήσουν επιλεκτικά μόνο τα μηνύματα που προορίζονται για αυτούς. Επίσης, το BAR έχει σημαντικά χαμηλότερο κόστος εκπομπής, καθώς χρησιμοποιεί έναν μηχανισμό μετάδοσης που του επιτρέπει να μειώσει το εύρος ζώνης προκαλώντας μικρή αύξηση της καθυστέρησης της επικοινωνίας, αλλά διατηρώντας το επίπεδο της ανωνυμίας.

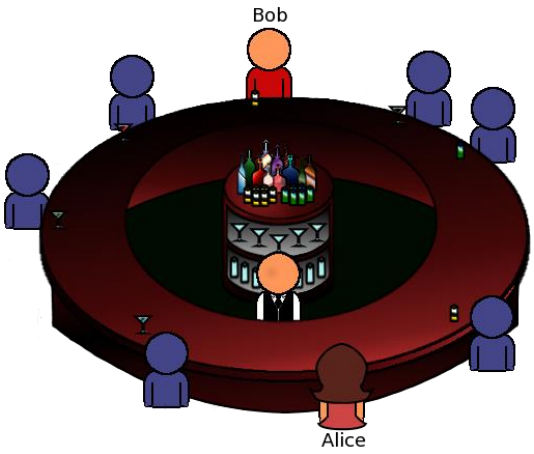
Για την ανωνυμία του αποστολέα χρησιμοποιείται η προσέγγιση των Mix-net συστημάτων. Παρόλα αυτά και σε αντίθεση με τα δίκτυα Mix-net που συνήθως απαιτούν επίπεδα κρυπτογράφησης δημόσιου κλειδιού, το BAR χρησιμοποιεί κυρίως συμμετρική κρυπτογράφηση. Η ανταλλαγή των κλειδιών ανώνυμα πραγματοποιείται με τη βοήθεια ενός ενσωματωμένου μηχανισμού διαχείρισης κλειδιών. Τέλος, για την επικοινωνία χρηστών που έχουν ανατεθεί σε διαφορετικούς εξυπηρέτες BAR, χρησιμοποιούνται για κάθε συνεδρία τυχαία σετ δημόσιου και ιδιωτικού κλειδιού.

3.2 Παράδειγμα χρήσης του πρωτοκόλλου

Στο συγκεκριμένο κεφάλαιο, θα γίνει μία αρχική επεξήγηση της ανώνυμης μετάδοσης με τη χρήση ενός απλού παραδείγματος [34].

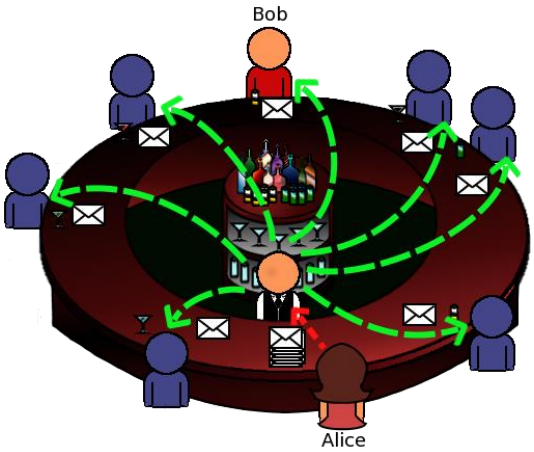
Ας πάρουμε για παράδειγμα την Alice και τον Bob να κάθονται σε ένα γεμάτο μπαρ. Εκτός από αυτούς του δύο, υπάρχουν και άλλοι άνθρωποι που στέκονται μεταξύ τους και πίνουν το ποτό τους. Ο μπάρμαν είναι πρόθυμος να εξυπηρετήσει τους πάντες γύρω. Η Alice επιθυμεί να επικοινωνήσει με τον Bob, αλλά δεν θέλει κανένας τρίτος στο μπαρ να το γνωρίζει και ασφαλώς ο τελευταίος να μην μπορεί να δει τα περιεχόμενα του μηνύματός της.

Επομένως, τι μπορεί να κάνει η Alice;



Εικόνα 3.1 Παράδειγμα BAR – Αρχική κατάσταση

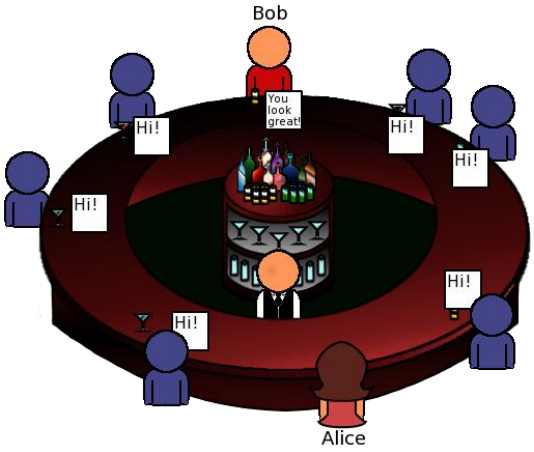
Η Alice γράφει ένα μήνυμα για κάθε πελάτη που βρίσκεται στο μπαρ και το τοποθετεί σε διαφορετικό φάκελο. Το μόνο πραγματικό μήνυμα είναι αυτό που θα αποσταλεί στον Bob και όλα τα άλλα είναι ψεύτικα. Στη συνέχεια, παραδίδει τους φακέλους στον μπάρμαν, μαζί με οδηγίες για τη δρομολόγηση του καθενός.



Εικόνα 3.2 Παράδειγμα BAR – Αποστολή μηνυμάτων

Ο κάθε πελάτης που παραλαμβάνει φάκελο από τον μπάρμαν, μπορεί να διαβάσει ένα ψεύτικο μήνυμα από την Alice, εκτός από τον Bob που είναι ο πραγματικός παραλήπτης.

Με αυτό τον τρόπο, όλοι οι πελάτες, ακόμα και ο μπάρμαν είναι σε θέση να καταλάβουν ότι η Alice επικοινώνησε με κάποιον στο μπαρ, αλλά δεν μπορούν να γνωρίζουν τον πραγματικό παραλήπτη, καθώς ο καθένας πήρε μήνυμα από την Alice. Οι πελάτες μπορούν να βρουν τον πραγματικό δέκτη του μηνύματος αν συνεργαστούν για να το ανακαλύψουν. Έτσι, έχει διασφαλιστεί η ανωνυμία του Bob.



Εικόνα 3.3 Παράδειγμα BAR – Παραλαβή μηνυμάτων

Με τον τρόπο που περιγράφηκε παραπάνω, διασφαλίζεται η ανωνυμία του Bob, αλλά όλοι γνωρίζουν ότι η Alice επικοινώνησε με κάποιον το μπαρ.

Το πρωτόκολλο BAR χρησιμοποιεί στην έκδοσή του βελτιώσεις προκειμένου να διασφαλίσει την ανωνυμία και του αποστολέα. Η Alice χρησιμοποιεί μία τυχαία διαδρομή από χρήστες δημιουργώντας έτσι μία διαδρομή δρομολόγησης, όπως στο πρωτόκολλο Tor. Όλοι οι χρήστες έχουν τυχαία κλειδιά, τα οποία ενημερώνονται δυναμικά σε κάθε σύνδεση και χρησιμοποιούνται για να κρύψουν τη διαδρομή του μηνύματος της Alice.

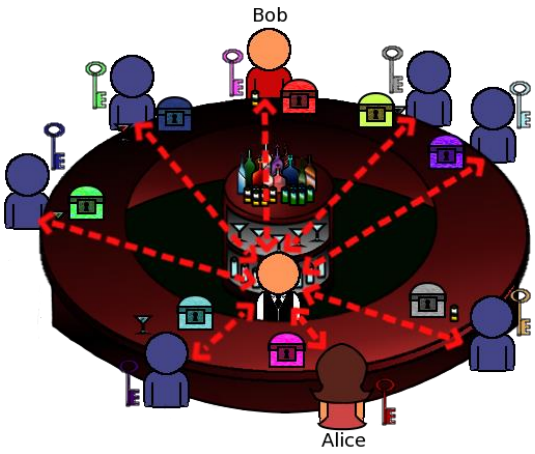
Επίσης, το πρωτόκολλο BAR παρέχει τη δυνατότητα εξασφάλισης της ανωνυμίας αποστολέα-παραλήπτη (sender-receiver anonymity). Αυτό επιτυγχάνεται μέσα από τη συνεχή αποστολή μηνυμάτων σταθερού μεγέθους και σε συγκεκριμένες χρονικές στιγμές στον μπάρμαν (τον BAR διακομιστή εκπομπής). Αν κάποιοι χρήστες δεν επιθυμούν να επικοινωνήσουν κάποια στιγμή, στέλνουν τότε ψεύτικα μηνύματα κατάλληλου μεγέθους στον μπάρμαν. Με αυτό τον τρόπο, κανένας δεν μπορεί να γνωρίζει κάποια στιγμή ποιοι χρήστες του συστήματος επικοινωνούν.

Το ερώτημα τώρα είναι πώς μπορεί να χρησιμοποιηθεί το πρωτόκολλο BAR σε ένα δίκτυο υπολογιστών:

Χρειάζεται ένας διακομιστής BAR, ο οποίος ενεργεί ως μπάρμαν και μεταδίδει τα μηνύματα των χρηστών στο δίκτυο. Η Alice δεν στέλνει διαφορετικό μήνυμα σε κάθε χρήστη, αλλά το ίδιο προωθείται στον διακομιστή BAR. Πριν την αποστολή του, το μήνυμα κρυπτογραφείται με ένα συμμετρικό κλειδί AES, έτσι ώστε μόνο ο Bob να μπορεί να το αποκρυπτογραφήσει.

Επίσης, μοιράζεται μία μυστική ετικέτα (label) ανάμεσα στους δύο χρήστες, η οποία προστίθεται και αυτή στο κρυπτογραφημένο μήνυμα. Το label χρησιμοποιείται προκειμένου οι παραλήπτες ενός μηνύματος να μην αποκρυπτογραφούν όλα τα μηνύματα που παραλαμβάνουν, αλλά μόνο αυτά που απευθύνονται σε αυτούς. Τα μηνύματα με άγνωστες ετικέτες διαγράφονται από τον παραλήπτη.

Το συμμετρικό κλειδί AES, καθώς και το label τα είχε ανταλλάξει η Alice με τον Bob στην αρχή. Επίσης, και τα δύο είναι για μία συνεδρία μόνο και μετά αλλάζονται.



Εικόνα 3.4 Παράδειγμα BAR – Βελτιωμένη επικοινωνία

3.3 Περιγραφή του πρωτοκόλλου BAR

Στο επόμενο κεφάλαιο θα γίνει αναλυτική περιγραφή του συστήματος BAR, με την παρουσίαση των χαρακτηριστικών και των πρωτοκόλλων που το αποτελούν.

Η αρχιτεκτονική του BAR αποτελείται από τρεις ομάδες συστημάτων: τους χρήστες, τους διακομιστές BAR (*BAR servers*) και έναν συντονιστή του συστήματος (*System Coordinator*), ο ρόλος του οποίου είναι να δημοσιεύει τις απαιτούμενες παραμέτρους και να υποστηρίζει τη λειτουργία του. Επίσης, ο coordinator οφείλει να διοργανώνει δυναμικά τα λογικά κανάλια μετάδοσης (*Clusters*). Τα Clusters μπορεί να αποτελούνται από έναν ή περισσότερους γειτονικούς BAR servers. Ο συντονιστής και οι διακομιστές πρέπει να είναι διαθέσιμοι σε πραγματικό χρόνο.

Πέρα από τις οντότητες που συμμετέχουν στο πρωτόκολλο BAR, υπάρχουν και κάποιες παράμετροι, που διασφαλίζουν την ομαλή λειτουργία του συστήματος και καθορίζονται από τον διαχειριστή του System Coordinator: η παράμετρος ανωνυμίας N_{min}, που περιγράφει το ελάχιστο πλήθος των χρηστών σε ένα Cluster και η παράμετρος αποτελεσματικότητα εκπομπής N_{max}, η οποία καθορίζει το μέγιστο πλήθος των ενεργών χρηστών που επιτρέπονται σε ένα Cluster.

Στον παρακάτω πίνακα παρουσιάζονται οι οντότητες, οι παράμετροι του συστήματος, καθώς και οι σημειογραφίες που χρησιμοποιούνται.

Οντότητες	**Περιγραφή**
$\mathbb{U} = \{\boldsymbol{u_1}, \boldsymbol{u_2}, \dots, \boldsymbol{u_N}\}$	Οι χρήστες του συστήματος
$\boldsymbol{BAR_1}, \boldsymbol{BAR_2}, \dots, \boldsymbol{BAR_M}$	Οι φυσικοί διακομιστές του συστήματος
$\boldsymbol{BAR_0}$	Ο συντονιστής του συστήματος
$\boldsymbol{Cluster_1}, \dots, \boldsymbol{Cluster_\mu}$	Τα λογικά κανάλια μετάδοσης
Παράμετροι συστήματος	**Περιγραφή**
$\boldsymbol{N_{min}}$	Η παράμετρος ανωνυμίας
$\boldsymbol{N_{max}}$	Η παράμετρος αποτελεσματικότητας εκπομπής
Σημειογραφίες	**Περιγραφή**
$\boldsymbol{nym_i}$	Το ψευδώνυμο που ταυτοποιεί έναν ανώνυμο χρήστη u_i

$\boldsymbol{pk_i, sk_i}$	Μόνιμο σετ δημόσιου/ιδιωτικού κλειδιού για τον χρήστη u_i
$\boldsymbol{k_{ij}, l_{ij}}$	Συμμετρικό κλειδί και label για δύο χρήστες u_i, u_j
$\boldsymbol{\overline{pk_\iota}, \overline{sk_\iota}}$	Ζεύγος δημόσιου/ιδιωτικού κλειδιού συνεδρίας για τον χρήστη u_i
$\boldsymbol{List_i}$	Η λίστα επαφών του χρήστη u_i που περλαμβάνει $[pk_j, k_{ij}, l_{ij}]$
$\boldsymbol{UsersList}$	Δημόσια λίστα με όλους τους χρήστες που περιέχει $[nym_i, pk_i, \sigma_i]$ όπου $\sigma_i = sig_{BAR_0}^{(nym_i \mid pk_i)}$
$\boldsymbol{ActiveList}$	Δημόσια λίστα με τους ενεργούς χρήστες που περιέχει $[barID_i, IP_i, \overline{pk_\iota}]$

Εικόνα 3.5 Οντότητες, παράμετροι και συντομογραφίες του BAR

Τα υπο-πρωτόκολλα που απαιτούνται για την ομαλή λειτουργία του συστήματος BAR είναι:

- Πρωτόκολλο Εγγραφής Χρήστη (User Registration Protocol)
- Πρωτόκολλο Εισόδου Χρήστη (User Login Protocol)
- Πρωτόκολλο Ανταλλαγής Κλειδιών (Key Exchange Protocol)
- Πρωτόκολλο BAR Ανώνυμης Επικοινωνίας για αξιόπιστα δίκτυα (BAR Communication Protocol - BCP)
- Πρωτόκολλο BAR Ανώνυμης Επικοινωνίας για μη αξιόπιστα δίκτυα (Extended BAR Communication Protocol – E-BCP)

Παρακάτω, θα γίνει μία αναλυτική περιγραφή των υπο-πρωτοκόλλων του συστήματος.

3.4 Αναλυτική περιγραφή των υπο-πρωτοκόλλων του συστήματος

Στη συνέχεια, θα περιγραφούν οι βασικές μέθοδοι που υποστηρίζει το πρωτόκολλο BAR για να εξασφαλίσει την ανωνυμία των χρηστών του και την ομαλή λειτουργία του συστήματος.

Ο coordinator καθορίζει στην αρχή το μέγιστο πλήθος των clusters και των BAR servers που θα αποτελούν το σύστημα. Επίσης, σε κάθε BAR server αναθέτει ένα τμήμα από διευθύνσεις χρηστών. Ανάλογα με το πλήθος των ενεργών χρηστών που υπάρχουν στους BAR servers, ο συντονιστής του συστήματος έχει τη δυνατότητα να συγχωνεύσει ή να διασπάσει κάποιους φυσικούς εξυπηρέτες σε ένα cluster, προκειμένου να διασφαλίσει ότι κάθε cluster έχει αριθμό χρηστών ανάμεσα στα όρια N_{min} και N_{max}, που έχουν οριστεί εξ αρχής. Αυτό έχει ως αποτέλεσμα να εξασφαλίζεται κάθε φορά από το πρωτόκολλο το ελάχιστο εύρος ζώνης που χρειάζεται για να λειτουργήσει ομαλά.

3.4.1 Πρωτόκολλο Εγγραφής Χρήστη (User Registration Protocol)

Ο κάθε χρήστης που επιθυμεί να χρησιμοποιήσει το πρωτόκολλο BAR, χρειάζεται την πρώτη φορά να εγγραφεί στην υπηρεσία. Οι χρήστες εγγράφονται δημοσιεύοντας ανώνυμα μέσω του Coordinator μία νέα καταχώρηση στο λίστα χρηστών ($UsersList$). Για την πρώτη επικοινωνία του χρήστη με τον Coordinator του συστήματος, δημιουργείται μία onion δρομολόγηση (onion path routing) χρησιμοποιώντας τα κλειδιά συνόδου των ενδιάμεσων κόμβων.

Παρόλο που η συγκεκριμένη διαδικασία εξασφαλίζει μόνο την ανωνυμία του αποστολέα και όχι του παραλήπτη (Coordinator), κρίνεται επαρκής για την λειτουργία της εγγραφής του χρήστη.

Σε αυτό το σημείο, θα παρουσιαστούν οι αρμοδιότητες των δύο συνεργαζόμενων μερών για την επιτυχή εγγραφή ενός νέου χρήστη.

1) Χρήστης u_i

   $\Rightarrow$ Επιλέγει ένα μοναδικό ψευδώνυμο nym_i και παράγει ένα ζευγάρι δημόσιου (pk_i) / ιδιωτικού (sk_i) κλειδιού.

⇒ Επιλέγει τυχαία τρεις ενεργούς χρήστες από την λίστα ($ActiveList$) και χρησιμοποιεί το προσωρινό δημόσιο κλειδί ($\overline{pk_j}$) του καθενός για να υλοποιήσει onion δρομολόγηση για τον Coordinator.

⇒ Χρησιμοποιεί τη συγκεκριμένη διαδρομή για να στείλει nym_i, pk_i στον Coordinator.

2) Coordinator BAR_0

⇒ Ελέγχει την εγκυρότητα των τιμών που παρέλαβε και αν δεν υπάρχει ήδη η εγγραφή.

⇒ Υπολογίζει το $\sigma_i = sig_{sk_{BAR_0}}^{(nym_i|pk_i)}$, το οποίο αποτελεί την υπογραφή του συντονιστή για το χρήστη $user_i$ που επιβεβαιώνει την εγκυρότητά του.

⇒ Προσθέτει την εγγραφή $[nym_i, pk_i, \sigma_i]$ στη λίστα όλων των χρηστών $UsersList$.

3.4.2 Πρωτόκολλο Εισόδου Χρήστη (User Login Protocol)

Ένας ήδη εγγεγραμμένος χρήστης μπορεί να εισέλθει στο σύστημα BAR ακολουθώντας την παρακάτω διαδικασία.

1) Χρήστης u_i

⇒ Υπολογίζει το $h_i = hash(nym_i|pk_i)$. Η τιμή h_i καθορίζει σε ποιο BAR server ανήκει ο κάθε χρήστης.

⇒ Χρησιμοποιεί τις παραμέτρους του συστήματος, που λαμβάνει από τον Coordinator για να βρει σε ποιο Cluster ανήκει, με βάση την τιμή h_i.

⇒ Δημιουργεί ένα νέο ζευγάρι δημόσιου / ιδιωτικού κλειδιού ($\overline{pk_i}$ / $\overline{sk_i}$) για την τρέχουσα συνεδρία.

⇒ Στέλνει στον Coordinator τα στοιχεία εισόδου του: $IP_i, \overline{pk_i}, BAR_i, Cluster_x$.

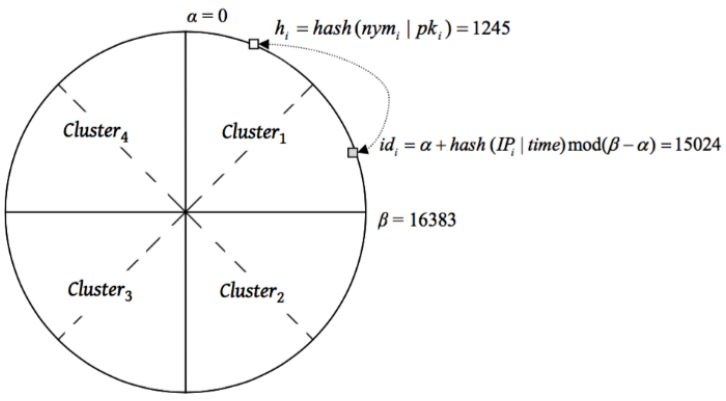
2) Coordinator BAR_0

⇒ Επιλέγει ένα τυχαίο $id_i \in BAR_i$, το οποίο δεν χρησιμοποιείται. Το id_i καθορίζει το BAR id που θα έχει ο συγκεκριμένος χρήστης για την τρέχουσα συνεδρία. Μετά την αποσύνδεσή του, το BAR id αποδεσμεύεται και μπορεί να χρησιμοποιηθεί από άλλον χρήστη.

⇒ Προσθέτει την εγγραφή $[id_i, IP_i, \overline{pk_i}]$ στη λίστα με τους ενεργούς χρήστες $ActiveList$.

Σε αυτό το σημείο πρέπει να διευκρινιστεί ότι ο Coordinator θα πρέπει να είναι ειλικρινής όσον αφορά τις παραμέτρους του συστήματος. Επίσης, είναι υπεύθυνος να παρακολουθεί συνεχώς τις δυναμικές αλλαγές στον αριθμό των χρηστών ανά Cluster προκειμένου να αποτρέψει την παραβίαση των ορίων N_{min} και N_{max}.

Η παρακάτω εικόνα παρουσιάζει ένα παράδειγμα εισόδου ενός χρήστη στο σύστημα BAR. Στο συγκεκριμένο παράδειγμα χρησιμοποιούνται 8 BAR servers και 4 λογικά Clusters. Ο χρήστης υπολογίζει το h_i, το οποίο προκύπτει ότι ανήκει στο BAR_2 server και στο $Cluster_1$. Στη συνέχεια, αποστέλλει τα απαραίτητα στοιχεία στον Coordinator, ο οποίος επιλέγει ένα τυχαίο $id_i \in BAR_2$.



Εικόνα 3.6 Παράδειγμα διαδικασίας εισόδου στο σύστημα BAR

3.4.3 Πρωτόκολλο Ανταλλαγής Κλειδιών (Key Exchange Protocol)

Για να είναι δυνατή η επικοινωνία μέσω του πρωτοκόλλου BAR, οι χρήστες θα πρέπει αρχικά να ανταλλάξουν ανώνυμα τα ζεύγη κλειδιών τους.

1) Χρήστης u_i
   ⇒ Για κάθε εγγραφή της $UsersList$:
      i) Επιλέγει ένα τυχαίο label l_{ij} και κλειδί k_{ij}.
      ii) Δημιουργεί την υπογραφή $\sigma_{ij} = sig_{sk_i}(nym_i, pk_i, pk_j, k_{ij}, l_{ij})$.
      iii) Δημιουργεί το κρυπτογραφημένο μήνυμα που θα αποστείλει $c_{ij} = enc_{pk_j}\{nym_i, pk_i, k_{ij}, l_{ij}, \sigma_{ij}\}$,
   ⇒ Επιλέγει 3 τυχαίους ενεργούς χρήστες (εγγραφές της $ActiveList$) και χρησιμοποιεί το κλειδί συνόδου $\overline{pk_j}$ του καθενός για να δημιουργήσει μία onion δρομολόγηση προς τον Coordinator.
   ⇒ Χρησιμοποιεί την συγκεκριμένη διαδρομή για να στείλει ανώνυμα στον Coordinator τα ζευγάρια $[nym_j, c_{ij}]$.
2) Coordinator BAR_0
   ⇒ Έχει ορίσει μία δημόσια λίστα με όλα τα ζευγάρια $[nym_j, c_{ij}]$.
   ⇒ Καταχωρεί στη λίστα όλες τις νέες εγγραφές που του αποστέλλονται μέσω των onion διαδρομών.
3) Οι υπόλοιποι χρήστες του BAR u_j, $j \neq i$
   ⇒ Κατά την είσοδό τους στο σύστημα BAR, ελέγχουν την δημόσια λίστα για νέες εγγραφές που τους αφορούν.
   ⇒ Για κάθε νέα εγγραφή, χρησιμοποιούν το ιδιωτικό τους κλειδί sk_j για να αποκρυπτογραφήσουν το c_{ij} και να λάβουν το $\{nym_i, pk_i, k_{ij}, l_{ij}, \sigma_{ij}\}$.
   ⇒ Επαληθεύουν την υπογραφή σ_{ij}. Αν είναι επιτυχής, προσθέτει τα στοιχεία $[pk_i, l_{ij}, k_{ij}]$ στην ιδιωτική του λίστα επαφών $List_j$.

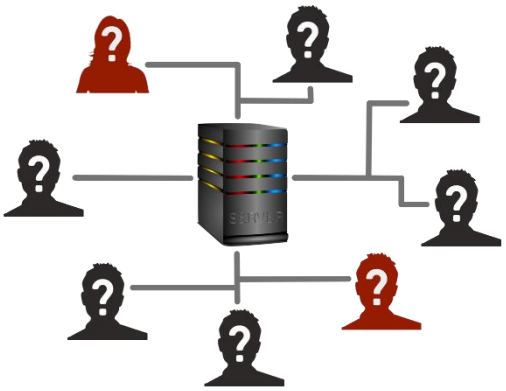
3.4.4 Πρωτόκολλο BAR για αξιόπιστα δίκτυα (BCP)

Στην συγκεκριμένη ενότητα θα γίνει μία παρουσίαση του τρόπου λειτουργίας του πρωτοκόλλου επικοινωνίας BAR (*BAR Communication Protocol – BCP*) υποθέτοντας ότι το δίκτυο είναι αξιόπιστο, δηλαδή ότι τα μηνύματα ανάμεσα στους χρήστες λαμβάνονται σε εύλογο χρονικό διάστημα. Στην επόμενη ενότητα θα επεκταθεί το πρωτόκολλο προκειμένου να μπορεί να συλλάβει την αποτυχία του δικτύου.

Αρχικά, θα περιγραφεί η επικοινωνία των χρηστών που ανήκουν στον ίδιο BAR server. Όπως αναφέρθηκε και παραπάνω, κάθε BAR server έχει συγκεκριμένο αριθμό χρηστών και σε κάθε χρήστη έχει δοθεί ένα BAR id που ανήκει στα όρια του εξυπηρέτη που ανήκει.

Ένας χρήστης, αφού έχει εισέλθει στο σύστημα BAR, επιλέγει κάποιον άλλο χρήστη, που ανήκει στον ίδιο BAR server και είναι ενεργός την συγκεκριμένη χρονική στιγμή, για να επικοινωνήσουν. Έτσι, δημιουργεί το μήνυμα που θέλει, το κρυπτογραφεί με το δημόσιο κλειδί του άλλου χρήστη και το αποστέλλει στον BAR server. Με τη σειρά του ο εξυπηρέτης λαμβάνει τα μηνύματα των χρηστών και τα μεταδίδει στο δίκτυο. Στη συνέχεια, κάθε χρήστης λαμβάνει όλα τα μηνύματα που υπάρχουν στο δίκτυο και ελέγχει ποια τον αφορούν, με βάση το label και το κλειδί του καθενός. Τέλος, χρησιμοποιεί το ιδιωτικό του κλειδί, προκειμένου να αποκρυπτογραφήσει τα μηνύματα με τα γνωστά labels και να διαβάσει τα μηνύματα.

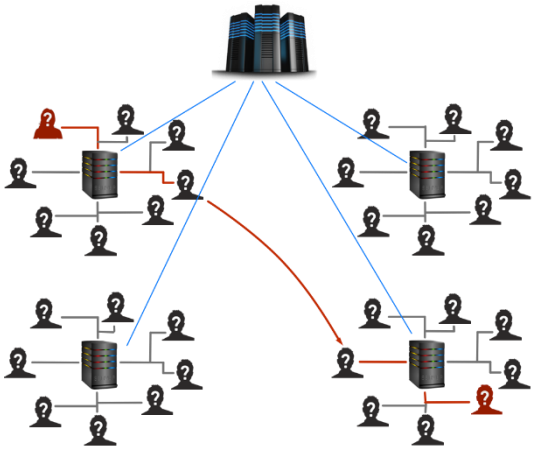
Σε αυτό το σημείο, αξίζει να αναφερθεί ότι οι ετικέτες (labels) και τα κλειδιά (keys) που ανταλλάσσουν δύο χρήστες για την επικοινωνία τους πρέπει να αλλάζουν πριν την αποστολή ενός νέου μηνύματος.



Εικόνα 3.7 BAR server

Έχοντας εξηγήσει τη λειτουργία ενός BAR server, στη συνέχεια θα περιγραφεί η επικοινωνία των χρηστών σε ένα δίκτυο BAR, το οποίο αποτελείται από Clusters με BAR servers. Σε ένα τέτοιο δίκτυο BAR, χρειάζεται και ένας συντονιστής (coordinator), ο οποίος θα συγχωνεύει ή θα διασπά δυναμικά τους φυσικούς εξυπηρέτες (BAR servers) σε λογικά συμπλέγματα (clusters), σύμφωνα με το πλήθος των χρηστών.

Σε ένα εκτεταμένο δίκτυο BAR, ένας χρήστης μπορεί να επικοινωνεί με όλους τους ενεργούς χρήστες, ανεξαρτήτως του server και cluster που ανήκουν. Για την μετάδοση σε ένα άλλο BAR χρησιμοποιείται το πρωτόκολλο γεφύρωσης (bridging protocol). Ένα εκτεταμένο δίκτυο BAR παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 3.8 Σύστημα BAR

Και σε αυτή την περίπτωση, ένας χρήστης, αφού εισέλθει στο σύστημα BAR, επιλέγει κάποιον χρήστη από τη λίστα επαφών του για να επικοινωνήσει. Με βάση το id του άλλου, ο πρώτος μπορεί να προσδιορίσει σε ποιο Cluster ανήκει. Έτσι, ακολουθείται η παρακάτω διαδικασία:

⇒ Ο αποστολέας u_i

i) Υπολογίζει το $h_j = hash(nym_j|pk_j)$, για να ταυτοποιήσει το Cluster που ανήκει ο παραλήπτης u_j.

ii) Επιλέγει από την λίστα των ενεργών χρηστών ($ActiveList$) έναν τυχαίο χρήστη "εισόδου" u_y, ο οποίος ανήκει στο ίδιο Cluster με τον χρήστη u_j.

iii) Επιλέγει από την λίστα επαφών του ($List_i$) έναν χρήστη "εξόδου" u_x, ο οποίος ανήκει στο ίδιο Cluster με τον αποστολέα και με τον οποίο έχουν ήδη οριστεί τα κοινά κλειδιά k_{ix} και labels l_{ix}.

iv) Δημιουργεί το μήνυμά του και το κρυπτογραφεί με το κοινό κλειδί k_{ij} που έχει ανταλλάξει με τον παραλήπτη u_j και προκύπτει το κρυπτογράφημα c_j.

v) Το κρυπτογράφημα c_j που προκύπτει παραπάνω, μαζί με το l_{ij}, στη συνέχεια το κρυπτογραφεί με το δημόσιο κλειδί συνόδου $\overline{pk_y}$ του χρήστη "εισόδου" u_y και προκύπτει ένα νέο κρυπτογραφημένο κείμενο c_y.

vi) Έπειτα, χρησιμοποιείται το κοινό κλειδί k_{ix} του αποστολέα u_i με τον χρήστη u_x για την εκ νέου κρυπτογράφηση του c_y, μαζί με την IP_y και προκύπτει το νέο κρυπτογράφημα c_x.

vii) Το μήνυμα $[l_{ix}, c_x]$ μεταδίδεται στο Cluster που ανήκει ο αποστολέας και ο χρήστης u_x.

⇒ Οι χρήστες του κοινού Cluster με τον αποστολέα

i) Από τα μηνύματα που παραλαμβάνουν ελέγχουν ποια απευθύνονται σε αυτούς με βάση τα labels.

⇒ Ο χρήστης "εξόδου" u_x

i) Αποκρυπτογραφεί με το κοινό κλειδί k_{ix} το c_x για να προκύψει η IP διεύθυνση του επόμενου χρήστη και το c_y.

ii) Στέλνει το c_y στον χρήστη «εισόδου» u_y.

⇒ Ο χρήστης "εισόδου" u_y

i) Αποκρυπτογραφεί το c_y χρησιμοποιώντας το ιδιωτικό κλειδί συνεδρίας $\overline{sk_y}$.

ii) Μεταδίδει στο Cluster που ανήκει το μήνυμα $[l_{ij}, c_j]$.

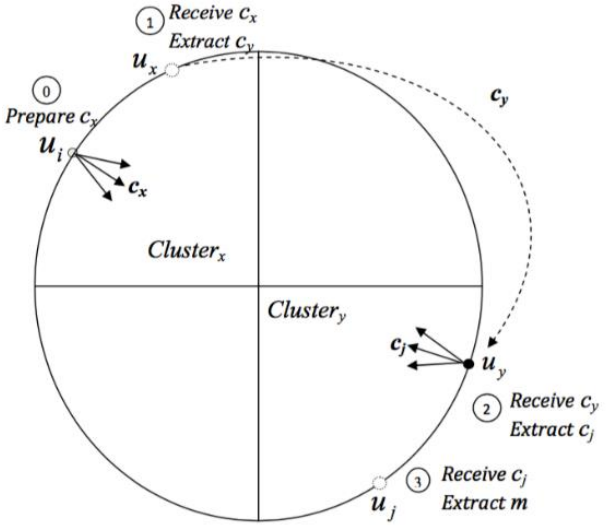
⇒ Οι χρήστες του κοινού Cluster με τον παραλήπτη

i) Από τα μηνύματα που παραλαμβάνουν ελέγχουν ποια απευθύνονται σε αυτούς με βάση τα labels.

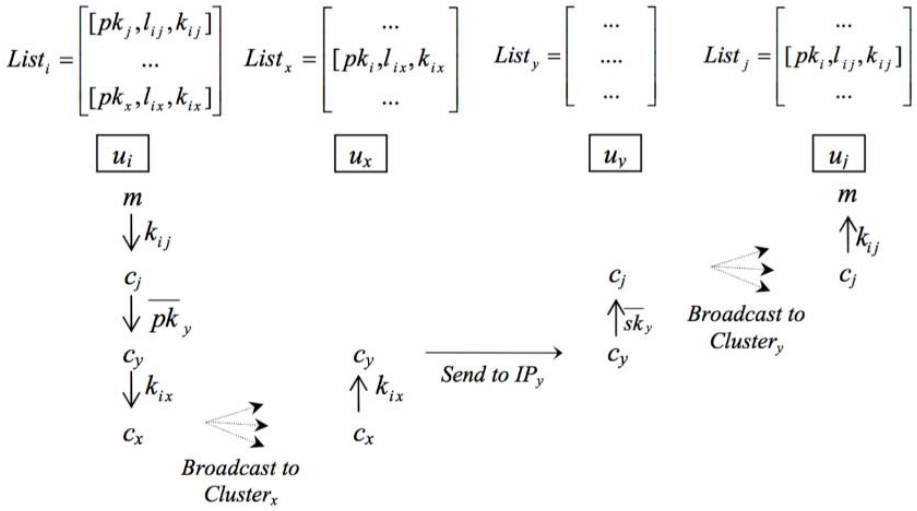
⇒ Ο τελικός παραλήπτης u_j

i) Αποκρυπτογραφεί το κρυπτογράφημα c_j με το κοινό κλειδί k_{ij} για να αποκαλύψει το αρχικό μήνυμα.

Παρακάτω, στην πρώτη εικόνα παρουσιάζεται το πρωτόκολλο BCP όπως περιγράφηκε. Στη δεύτερη εικόνα επεξηγούνται γραφικά οι ροές επικοινωνίας και οι κρυπτογραφήσεις που πραγματοποιούνται για την επικοινωνία ανάμεσα σε χρήστες διαφορετικού Cluster.



Εικόνα 3.9 Περιγραφή συστήματος BCP



Εικόνα 3.10 Ροές επικοινωνίας και κρυπτογραφήσεις

3.3.5 Πρωτόκολλο BAR για μη αξιόπιστα δίκτυα (E-BCP)

Στην προηγούμενη ενότητα παρουσιάστηκε το πρωτόκολλο BCP, το οποίο λειτουργεί πλήρως στην περίπτωση που το δίκτυο είναι αξιόπιστο. Στη συγκεκριμένη ενότητα, θα επεκταθεί το πρωτόκολλο BCP προκειμένου να μπορεί να αντιμετωπίζει αποτυχία της επικοινωνίας, που απορρέει από απώλεια, πτώση ή καθυστέρηση των πακέτων που ανταλλάσσονται. Κάτι τέτοιο μπορεί να επιτευχθεί με περιορισμένη και ελεγχόμενη επαναχρησιμοποίηση των ετικετών (labels), όπου υπάρχει αποτυχία.

Η επέκταση του πρωτοκόλλου BCP ονομάστηκε *Extended-BCP (E-BCP)*. Στο πρωτόκολλο E-BCP, κάθε καταχώρηση της λίστας επαφών του κάθε χρήστη περιέχει δύο τιμές. Εκτός από το κοινό κλειδί και label που θα χρησιμοποιηθεί για το συγκεκριμένο μήνυμα, κρατούνται και οι προηγούμενες τιμές τους. Έτσι, η λίστα επαφών ενός χρήστη u_i παίρνει πλέον τη μορφή: $[pk_j, k_{ij}, l_{ij}, k'_{ij}, l'_{ij}]$, όπου (k_{ij}, l_{ij}) οι τρέχουσες τιμές και (k'_{ij}, l'_{ij}) οι τιμές που θα χρησιμοποιηθούν για το επόμενο μήνυμα.

Όταν υπάρχει αδυναμία επικοινωνίας, τότε και μόνο τότε, ο αποστολέας θα χρησιμοποιήσει την τρέχουσα ετικέτα l_{ij}, ώστε να αποφευχθεί αποσυγχρονισμός της ετικέτας που έχει ανταλλαχθεί με έναν χρήστη. Οι ετικέτες και τα κλειδιά ανταλλάσσονται με τέτοιο τρόπο ώστε ανά πάσα στιγμή να υπάρχει τουλάχιστον ένα κοινό ζεύγος (κοινό κλειδί, ετικέτα) για δύο χρήστες. Οι χρήστες δοκιμάζουν να φιλτράρουν και να αποκρυπτογραφήσουν τα μηνύματα χρησιμοποιώντας και τα δύο ζεύγη.

3.5 Ανάλυση της ασφάλειας

Το πρωτόκολλο BCP χρησιμοποιεί ένα τυπικό TCP δίκτυο επικάλυψης. Τα κανάλια μετάδοσης έχουν υλοποιηθεί μέσω διακομιστών, οι οποίοι διατηρούν μόνιμες συνδέσεις με τους πελάτες και των οποίων ο ρόλος είναι να μεταδώσουν όλη την κίνηση που λαμβάνουν από τους χρήστες. Η IP διεύθυνση του κάθε ενεργού χρήστη βρίσκεται στην λίστα $ActiveList$ και το ανώνυμο δημόσιο κλειδί του βρίσκεται στην λίστα $UsersList$. Όμως, η συσχέτιση των καταχωρήσεων των δύο λιστών μπορεί να γίνει μόνο από τον ίδιο τον χρήστη.

Επίσης, θα πρέπει να αποσαφηνιστεί το είδος των χρηστών που μπορούν να βρεθούν σε ένα σύστημα BAR. Υπάρχουν οι ειλικρινείς χρήστες, οι οποίοι τηρούν το πρωτόκολλο και εκείνοι που το εκθέτουν σε κίνδυνο, καθώς συνεργάζονται με τον αντίπαλο παρέχοντάς του τα κλειδιά και τις ετικέτες που διαθέτουν για να επικοινωνούν με άλλους. Αντίπαλος του συστήματος μπορεί να θεωρηθεί ένας ειλικρινής-αλλά-περίεργος χρήστης, ο οποίος παρακολουθεί όλα τα κανάλια επικοινωνίας, έχει πρόσβαση στα ιδιωτικά κλειδιά των συνεργαζόμενων με αυτόν χρηστών και μπορεί να εκτελέσει ενεργές επιθέσεις στην

επικοινωνία. Βασικός στόχος του πρωτοκόλλου BAR είναι να παρέχει ανωνυμία στο σύνολο των καλών χρηστών του συστήματος.

Τέλος, θα πρέπει να διευκρινιστεί η εμπιστοσύνη που υπάρχει στα μέρη του συστήματος από τους χρήστες του. Οι διακομιστές BAR, συμπεριλαμβανομένου και του συντονιστή (coordinator), παρέχουν σε πραγματικό χρόνο αξιόπιστες υπηρεσίες. Ειδικότερα, θεωρείται δεδομένη η ακεραιότητα και η διαθεσιμότητα σε πραγματικό χρόνο των δημοσίων καταλόγων ($UsersList$, $ActiveList$) και πληροφοριών του πρωτοκόλλου. Όσον αφορά τους χρήστες του συστήματος, θεωρούμε ότι κάθε BAR server μπορεί να περιέχει ένα μεγάλο αριθμό από κακόβουλους χρήστες, αλλά υπάρχει και ένας εξίσου μεγάλος αριθμός έντιμων χρηστών, οι οποίοι διασφαλίζουν την ομαλή λειτουργία του πρωτοκόλλου BAR.

3.5.1 Ανωνυμία αποστολέα (Sender anonymity)

Βασικό γνώρισμα του πρωτοκόλλου επικοινωνίας BAR είναι ότι διατηρεί κρυφή την ταυτότητα του αποστολέα. Υποθέτοντας ότι τα κανάλια επικοινωνίας και οι BAR servers είναι αξιόπιστοι, μπορεί να αποδειχτεί ότι ο αποστολέας ενός μηνύματος παραμένει ανώνυμος, αν τουλάχιστον ένας από τους ενδιάμεσους χρήστες, που συμμετέχουν στο πρωτόκολλο bridging, είναι ειλικρινής.

Παρακάτω, θα εξεταστούν τρεις περιπτώσεις διαφορετικών ειδών χρηστών που συμμετέχουν στην επικοινωνία, προκειμένου να αποδειχτεί η ανωνυμία του αποστολέα. Για την διασφάλισή της, θεωρείται σίγουρο ότι ο αποστολέας u_i είναι ειλικρινής χρήστης του συστήματος, ενώ το είδος χρήστη του παραλήπτη u_j δεν την επηρεάζει. Για άλλη μία φορά, θα θεωρήσουμε έναν χρήστη “εξόδου” u_x, ο οποίος ανήκει στο ίδιο Cluster με τον αποστολέα και έναν χρήστη “εισόδου” u_y, ο οποίος ανήκει στο Cluster του παραλήπτη.

- *Σενάριο 1: Ο χρήστης u_x συνεργάζεται με τον αντίπαλο*

  Σε αυτή την περίπτωση, ο εισβολέας του συστήματος δεν γνωρίζει τα κλειδιά του u_y. Μπορεί μόνο να χρησιμοποιήσει τη λίστα επαφών του χρήστη u_x με τον οποίο συνεργάζεται, ώστε να συνδέσει το κρυπτογράφημα που λαμβάνει ο u_x (το οποίο έχει μεταδώσει ο αποστολέας u_i) με το κρυπτογράφημα που έχει λάβει ο u_y. Όμως, δεν μπορεί να μάθει τον αρχικό αποστολέα που μπορεί να είναι οποιοσδήποτε χρήστης του Cluster του u_x.

- *Σενάριο 2: Ο χρήστης u_y συνεργάζεται με τον αντίπαλο*

  Σε αυτή την περίπτωση, ο εισβολέας δεν μπορεί να γνωρίζει τα κλειδιά του u_x. Μπορεί να αποκαλύψει μόνο από ποιον χρήστη παρέλαβε ο u_y το κρυπτογραφημένο μήνυμα και σε ποιον χρήστη πρέπει να το παραδώσει. Επομένως, αποτυγχάνει για άλλη μια φορά να μάθει τον αρχικό αποστολέα.

- *Σενάριο 3: Και οι δύο χρήστες u_x και u_y συνεργάζονται με τον αντίπαλο*

  Σε αυτή την περίπτωση είναι πιο εύκολα τα πράγματα για τον εισβολέα. Γνωρίζει τα κλειδιά όλων των ενδιάμεσων χρηστών της επικοινωνίας και του τελικού παραλήπτη. Έτσι, μπορεί να χαρτογραφήσει τη διαδρομή του μηνύματος και χρησιμοποιώντας τη λίστα επαφών του χρήστη u_x να ανακαλύψει τον αρχικό αποστολέα.

3.5.2 Ανωνυμία παραλήπτη (Receiver anonymity)

Ένα ακόμα γνώρισμα του πρωτοκόλλου BAR είναι η εξασφάλιση της ανωνυμίας του παραλήπτη. Αυτή τη φορά, θεωρούμε ότι ο παραλήπτης του μηνύματος είναι ειλικρινής χρήστης του συστήματος. Ο αντίπαλος το συστήματος θα πρέπει να χαρτογραφήσει την πορεία του μηνύματος $c_x \rightarrow c_y \rightarrow c_j \rightarrow m$. Κάτι τέτοιο είναι εφικτό, καθώς γνωρίζει τα κλειδιά του αποστολέα και των ενδιάμεσων χρηστών. Όμως, λόγω της μετάδοσης του μηνύματος στο Cluster του παραλήπτη, όλοι οι χρήστες παραλαμβάνουν το μήνυμα. Οπότε, δε θα μπορέσει να αντιληφθεί ποιος χρήστης είναι αυτός που το αποκρυπτογράφησε.

3.5.3 Ανωνυμία αποστολέα-παραλήπτη (Sender-Receiver anonymity)

Το πρωτόκολλο BAR στοχεύει επίσης στη διασφάλιση της ανωνυμίας της επικοινωνίας δύο χρηστών. Για τη συγκεκριμένη περίπτωση θεωρείται ότι οι δύο χρήστες που ανταλλάσσουν μηνύματα (ο αποστολέας u_i και ο παραλήπτης u_j) είναι ειλικρινείς χρήστες του συστήματος.

Δεδομένου ότι όλοι οι χρήστες μεταδίδουν κρυπτογραφημένα μηνύματα σταθερού μεγέθους σε σταθερό ρυθμό, δεν είναι εφικτό για τον εισβολέα να διακρίνει τα ψεύτικα μηνύματα από τα πραγματικά. Παράλληλα, ο αντίπαλος δεν μπορεί να προσδιορίσει τα μηνύματα του bridging πρωτοκόλλου, καθώς όλοι οι χρήστες στέλνουν σε τυχαία χρονικά διαστήματα και δέκτες προσομοιωμένα bridging μηνύματα.

Έτσι, ακόμα και αν οι δύο ενδιάμεσοι χρήστες της bridging επικοινωνίας για την αποστολή ενός στοχευμένου μηνύματος συνεργάζονται με τον κακόβουλο, το μόνο που μπορεί να αντιληφθεί ο τελευταίος είναι ότι κάποιος χρήστης του ενός Cluster επικοινωνεί με έναν χρήστη του άλλου Cluster.

3.5.4 Ανωνυμία συνεδρίας (Session anonymity)

Η ανωνυμία συνόδου στοχεύει στην πλήρη ασφάλεια μίας συνεδρίας στο σύστημα BAR, δηλαδή από τη στιγμή που αναγνωρίζεται το μήνυμα και ενημερώνονται τα κλειδιά και labels των συμμετεχόντων χρηστών.

Στο πρωτόκολλο E-BCP, που αποτελεί την επέκταση του αρχικού πρωτοκόλλου BCP, τα κλειδιά των χρηστών ενημερώνονται στην αρχή της επικοινωνίας, προκειμένου τα μηνύματα που έχουν σταλθεί σε προηγούμενες συνεδρίες να μην συνδεθούν με αυτούς. Ο κάθε χρήστης θα λάβει ένα μήνυμα επιβεβαίωσης στην περίπτωση που ενημερώθηκαν τα κλειδιά με επιτυχία. Τότε, το πρωτόκολλο E-BCP λειτουργεί όπως το BCP. Σε αντίθετη περίπτωση, χρησιμοποιούνται τα προηγούμενα labels με κίνδυνο τα μηνύματα να συνδεθούν με τους χρήστες που επικοινωνούν.

Αν υπάρχει αδυναμία επικοινωνίας σε ένα σύστημα E-BCP, ο αποστολέας θα χρησιμοποιήσει ξανά την προηγούμενη ετικέτα, με αποτέλεσμα τον κίνδυνο συσχέτισης με προηγούμενα μηνύματα. Ωστόσο, μόλις ολοκληρωθεί μία συνεδρία (ο αποστολέας λάβει επιβεβαίωση), οι τιμές των κλειδιών ενημερώνονται με τυχαίες τιμές. Έτσι, ο εισβολέας δεν μπορεί να ανακαλύψει περισσότερες πληροφορίες για τα μυστικά κλειδιά του αποστολέα και του παραλήπτη σε προηγούμενη συνεδρία, διασφαλίζοντας έτσι την ανωνυμία τους και αυτή της επικοινωνίας τους.

3.5.5 Άριστη μελλοντική εμπιστευτικότητα (Perfect forward secrecy)

Ως μελλοντική εμπιστευτικότητα (forward secrecy) ορίζεται η ιδιότητα εκείνη η οποία εξασφαλίζει την προστασία της εμπιστευτικότητας όλων των μηνυμάτων τα οποία έχουν ανταλλαγεί στο παρελθόν, *πριν από την αποκάλυψη του κλειδιού κρυπτογράφησης*. Δηλαδή, σε περίπτωση που ένα κλειδί αποκαλυφθεί σε κάποια στιγμή στο μέλλον, να μην επηρεάζεται η εμπιστευτικότητα όλων των ανταλλαγών που πραγματοποιήθηκαν πριν την παραβίαση της ασφάλειας του κλειδιού. Κάτι τέτοιο διασφαλίζεται από το πρωτόκολλο BCP, καθώς σε κάθε ανταλλαγή μηνύματος δημιουργείται νέο ζεύγος κλειδιού και ετικέτας. Παρόμοια, στην επέκταση του πρωτοκόλλου BAR, E-BCP, διασφαλίζεται επίσης η εμπιστευτικότητα, καθώς μία ετικέτα επαναχρησιμοποιείται μόνο στην περίπτωση που μία συνεδρία δεν έχει ολοκληρωθεί.

3.5.6 Άλλα χαρακτηριστικά ασφάλειας

Εκτός από την ανάλυση των ιδιοτήτων της ανωνυμίας, παρακάτω θα εξεταστούν και άλλα χαρακτηριστικά ασφάλειας που παρέχονται από το πρωτόκολλο BAR.

1. *Αντοχή σε επιθέσεις επιλεγμένης τοποθέτησης (positioning attacks)*

   Δεδομένου ότι οι παράμετροι ομαδοποίησης είναι δημόσιες, ένας κακόβουλος χρήστης μπορεί να ενταχθεί σε ένα διαφορετικό Cluster από αυτό που πραγματικά ανήκει, προκειμένου να βρίσκεται στο ίδιο Cluster-στόχο με τον αποστολέα ή τον παραλήπτη.

Positioning attacks εναντίον του αποστολέα. Ένας εισβολέας, τοποθετώντας τον εαυτό του στο ίδιο Cluster με τον αποστολέα, ποτέ δε θα επιλεγεί ως χρήστης "εξόδου" από οποιονδήποτε χρήστη. Αυτό συμβαίνει γιατί ένας χρήστης επιλέγει τον χρήστη "εξόδου" από τη λίστα επαφών του.

Positioning attacks εναντίον του παραλήπτη. Ένας εισβολέας, τοποθετώντας τον εαυτό του στο Cluster του παραλήπτη, μπορεί να επιλεγεί ως χρήστης "εισόδου" της επικοινωνίας, δεδομένου ότι ο συγκεκριμένος χρήστης επιλέγεται τυχαία από τη λίστα με τους ενεργούς χρήστες. Ωστόσο, ένας κακόβουλος χρήστης "εισόδου" δεν μπορεί να διακρίνει ποιος είναι ο πραγματικός δέκτης, λόγω της μετάδοσης των μηνυμάτων που γίνεται στο Cluster.

2. *Αντοχή σε χρήστες με πολλαπλές ταυτότητες*

   Ένας κακόβουλος χρήστης μπορεί αν δημιουργήσει πολλαπλά ζεύγη κλειδιών και ψευδώνυμα που ανήκουν σε ένα Cluster. Στη συνέχεια, θα πλαστογραφήσει πολλαπλές IP διευθύνσεις, με σκοπό να ενταχθεί πολλές φορές στο Cluster-στόχο και έτσι να μειώσει το επίπεδο της ανωνυμίας του. Παρά το γεγονός ότι αυτές οι περιπτώσεις δεν μπορούν να προληφθούν πλήρως, το πρωτόκολλο είναι εγγενώς ανθεκτικό σε επιθέσεις πολλαπλής ταυτότητας, λόγω της μετάδοσης των μηνυμάτων.

   Επίσης, θα πρέπει να σημειωθεί ότι ένας τέτοιος εισβολέας λαμβάνει την κίνηση της μετάδοσης του Cluster τόσες φορές όσες είναι και οι διαφορετικές ταυτότητές του. Αυτό περιορίζει τις δυνατότητες του χρήστη να παρουσιάσει πολλαπλούς λογαριασμούς, δεδομένου ότι θα προκαλέσει άρνηση της υπηρεσίας (Denial-of-service) στον εαυτό του, λόγω του μεγάλου όγκου της λαμβανόμενης κυκλοφορίας.

3. *Αντοχή σε επιθέσεις ανάλυσης κυκλοφορίας (traffic analysis attacks)*

   Λόγω της μετάδοσης των μηνυμάτων, το πρωτόκολλο BAR δεν είναι ευάλωτο σε επιθέσεις ανάλυσης της κυκλοφορίας. Για παράδειγμα, ένας εισβολέας δεν μπορεί να μαρκάρει μία ροή στην είσοδο προκειμένου να την παρακολουθήσει στην έξοδό της.

4. *Χαλάρωση της εμπιστοσύνης*

   Η ανάλυση της ασφάλειας του πρωτοκόλλου BAR στηρίζεται στην υπόθεση εμπιστοσύνης μίας αξιόπιστης υπηρεσίας παράδοσης και έμπιστων διακομιστών BAR. Η υπόθεση της αξιόπιστης παράδοσης όλων των μηνυμάτων από τους servers μπορεί να μειωθεί σημαντικά, αν χρησιμοποιείται το πρωτόκολλο εκτεταμένης επικοινωνίας E-BCP.

   Η υπόθεση των αξιόπιστων διακομιστών μπορεί να χαλαρώσει συνδυάζοντας τοπική επαλήθευση και μηχανισμούς παγκόσμιας φήμης. Επειδή κάθε BAR server αναμένεται να μεταδίδει όλα τα μηνύματα σε όλους τους χρήστες, είναι εύκολο για κάθε χρήστη να εξακριβώσει σε τοπικό επίπεδο εάν ένας διακομιστής μεταδίδει σωστά τα πακέτα του σε άλλους, επιθεωρώντας την μεταδιδόμενη κίνηση. Οι χρήστες μπορούν επίσης να δοκιμάσουν τη συμπεριφορά ενός BAR server με την κρυπτογράφηση μηνυμάτων που προορίζονται για τους ίδιους.

   Στη συνέχεια, μπορεί να κατασκευαστεί ένας μηχανισμός φήμης με τον οποίο οι χρήστες μπορούν να βαθμολογήσουν τους διακομιστές BAR, με βάση την τοπική επαλήθευσή τους. Οι servers με χαμηλή φήμη θα εξαιρούνται από το σύστημα για ένα ορισμένο χρονικό διάστημα ή μόνιμα.

5. *Διανομή του ρόλου των BAR servers*

   Μια πιθανή επέκταση θα ήταν η διανομή του ρόλου των BAR servers μεταξύ των χρηστών, οι οποίοι θα μπορούσαν να μεταδίδουν συνεργατικά τα πακέτα στα Clusters. Ένας τέτοιος σχεδιασμός εγείρει ζητήματα ασφάλειας που σχετίζονται με κακόβουλους χρήστες, οι οποίοι δε θα διαβιβάζουν σωστά τα πακέτα. Ένας τρόπος να αντιμετωπιστεί μη αξιόπιστη συμπεριφορά είναι οι μηχανισμοί φήμης που αναφέρθηκαν παραπάνω.

3.6 Ανάλυση της αποδοτικότητας

3.6.1 Ανάλυση της μετάδοσης σε ένα Cluster

Προκειμένου να εξεταστεί το εύρος ζώνης μετάδοσης, καθώς και ο χρόνος παράδοσης μέσα σε ένα Cluster που περιλαμβάνει ένα BAR server, χρησιμοποιήθηκε η υφιστάμενη υλοποίηση του πρωτόκολλου BAR και έγιναν κάποιες μετρήσεις.

Αρχικά, και για τον υπολογισμό του απαιτούμενου χρόνου παράδοσης από το ένα άκρο της επικοινωνίας στο άλλο, προσμετρήθηκαν όλες οι καθυστερήσεις που επιβάλλονται από το πρωτόκολλο, οι οποίες εμφανίζονται από την προετοιμασία του μηνύματος μέχρι και την παραλαβή του. Σε αυτές συμπεριλαμβάνεται ο χρόνος που απαιτείται για την κρυπτογράφηση ενός μηνύματος με τη δημόσια κλείδα του παραλήπτη, την αποστολή του μηνύματος από τον αποστολέα στον BAR server, την παραλαβή και μετάδοση του μηνύματος από τον διακομιστή σε όλους τους χρήστες του δικτύου του, την παραλαβή και το φιλτράρισμα της μεταδιδόμενης κυκλοφορίας από τον παραλήπτη και την αποκρυπτογράφηση του επιτυχώς φιλτραρισμένου μηνύματος.

Έτσι, χρησιμοποιώντας την υφιστάμενη υλοποίηση, έγιναν κάποια πειράματα και υπολογίστηκε ο μέσος χρόνος παράδοσης. Ο παρακάτω πίνακας παρουσιάζει τα τέσσερα διαφορετικά σενάρια που χρησιμοποιήθηκαν για το παραπάνω πείραμα. Σε αυτά έχει χρησιμοποιηθεί διαφορετικός αριθμός χρηστών που συμμετέχουν στο πρωτόκολλο BAR, θεωρώντας ότι τα ζευγάρια επικοινωνίας είναι $N/2$ (κάθε χρήστης μπορεί να είναι αποστολέας ή παραλήπτης). Για κάθε σενάριο έχουν γίνει 50 δοκιμές και υπολογίστηκε ο μέσος χρόνος. Σε όλες τις δοκιμές, οι χρήστες στέλνουν συνεχώς μηνύματα, τα οποία μπορεί να είναι πραγματικά ή θόρυβος, σε σταθερό χρονικό διάστημα ίσο με $290\,msec$. Επίσης, το μήκος του κάθε μηνύματος είναι $0.6\,KB$, τα οποία περιλαμβάνουν $0.5\,KB$ πραγματικού μηνύματος και $0.1\,KB$ που χρησιμοποιείται για τα κλειδιά, τα labels και τα δεσμευμένα bits. Τα συγκεκριμένα στοιχεία οδηγούν σε έναν εκτιμώμενο ρυθμό αποστολής της τάξης των $16.5\,Kbps$ ($2.6\,KBps$).

Με βάση τον παρακάτω πίνακα, παρατηρείται ότι ο μέσος χρόνος παράδοσης σε ένα Cluster κυμαίνεται ανάμεσα στα 500 και $1400\,msec$. Με άλλα λόγια, ο χρόνος εξαρτάται αποκλειστικά από το εύρος ζώνης μετάδοσης, ενώ τα κόστη υπολογισμού συνεισφέρουν ελάχιστα. Αυτό οφείλεται στον *μηχανισμό φιλτραρίσματος* που περιλαμβάνεται στο πρωτόκολλο BAR, ο οποίος επιτρέπει στους χρήστες να φιλτράρουν αρκετή από τη μεταδιδόμενη κίνηση με ελάχιστες υπολογιστικές απώλειες. Για την κυκλοφορία σε ένα Cluster και σε κάθε περίοδο μετάδοσης, ο κάθε χρήστης εκτελεί μόνο μία αποκρυπτογράφηση, καθώς και ψευδή πακέτα φιλτράρονται και διαγράφονται. Από την άλλη πλευρά, το κόστος του εύρους μετάδοσης αυξάνεται ανάλογα σε σχέση με τον αριθμό των χρηστών.

Test #	Αριθμός χρηστών (N)	Ταυτόχρονα ζεύγη	Εύρος μετάδοσης (Mbps/MBps)	Χρόνος παράδοσης (msec)
1	100	50	1.62/0.2	550
2	200	100	3.23/0.4	720
3	300	150	4.85/0.6	1320
4	400	200	6.46/0.8	1400

Εικόνα 3.11 Χρόνοι παράδοσης σε ένα Cluster για διαφορετικό αριθμό χρηστών

3.6.2 Τεχνικές βελτιστοποίησης του εύρους: Επιλεκτική Μετάδοση

Στη συνέχεια και για τον περιορισμό του κόστους του εύρους ζώνης, προτάθηκαν κάποιες βελτιστοποιήσεις, οι οποίες στοχεύουν στη μείωση της κυκλοφορίας μετάδοσης μέσα από μία μικρή αύξηση του χρόνου παράδοσης, διατηρώντας σταθερή την παράμετρο της ανωνυμίας.

Έτσι, χρησιμοποιήθηκε η χρηστικότητα του ελεγχόμενου label, που περιγράφεται στο πρωτόκολλο E-BCP. Υποθέτοντας ότι κάθε BAR server χρησιμοποιεί μία παράμετρο μετάδοσης q που κυμαίνεται στις τιμές 0 και 1 ($0 < q < 1$), θεωρείται ότι σε κάθε περίοδο μετάδοσης, ο διακομιστής επιλέγει τυχαία και

μεταδίδει qN από τα N ληφθέντα μηνύματα, διαγράφοντας τα υπόλοιπα $(1-q)N$ μηνύματα. Έτσι, οι αποστολείς των μη επιλεγμένων μηνυμάτων θα αντιμετωπίσουν το συγκεκριμένο συμβάν σαν αποτυχία του δικτύου και θα επαναποστείλουν τα μηνύματά τους χρησιμοποιώντας το προηγούμενο label. Όπως προκύπτει λοιπόν, κάθε μήνυμα απαιτεί κατά μέσο όρο q^{-1} μεταδόσεις.

Στη συνέχεια, υποθέτοντας ότι η πιθανότητα p ένας χρήστης να αποστέλλει πραγματικό μήνυμα κυμαίνεται από 0 μέχρι 1 $(0 < p < 1)$, θεωρούνται δύο περιπτώσεις για τα μη επιλεγμένα μηνύματα: πραγματικά ή θόρυβος. Η πιθανότητα ένα πραγματικό μήνυμα να διαγράφηκε είναι:

$$\Pr[real\ message\ \in \{dropped\ message\}] = p(1-q)$$

Έτσι, κάθε πραγματικό μήνυμα απαιτεί κατά μέσο όρο μία επιπλέον καθυστέρηση από $pq^{-1}(1-q)$ μεταδόσεις. Αν $t_c(N)$ θεωρείται ο αναμενόμενος χρόνος παράδοσης σε ένα Cluster με N χρήστες και t_b ο χρόνος που απαιτείται για τις διαδικασίες του πρωτοκόλλου γεφύρωσης (bridging protocol) στο σύστημα E-BCP, τότε προκύπτει ότι ο χρόνος παράδοσης ενός πραγματικού μηνύματος είναι:

$$t_{end} = 2[1 + pq^{-1}(1-q)]t_c(qN) + t_b$$

Με άλλα λόγια, το εύρος μετάδοσης μειώνεται σημαντικά, καθώς για ρυθμό αποστολής $s\ Kbps$ ο κάθε χρήστης αφιερώνει μόνο $(qsN)\ Kbps$ αντί για $(sN)\ Kbps$ του εύρους ζώνης τους, για την απόκτηση N-ανωνυμίας.

Κεφάλαιο 4

Επέκταση της υλοποίησης: Ανάλυση και Σχεδίαση

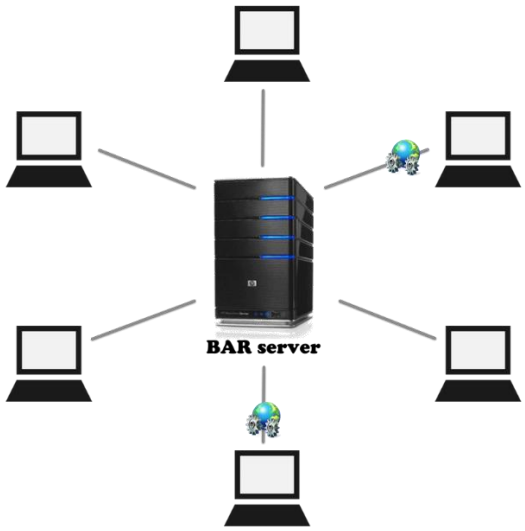
Σε αυτό το κεφάλαιο γίνεται η ανάλυση και ο αρχικός σχεδιασμός του συστήματος, ώστε να αποσαφηνιστούν τα βήματα που έγιναν για να επιτευχθεί το τελικό αποτέλεσμα. Συλλέγονται οι απαιτήσεις που προκύπτουν με βάση τις ανάγκες των χρηστών και τις διαθέσιμες λειτουργίες του συστήματος. Στη συνέχεια και λαμβάνοντας υπόψη τις πληροφορίες που συλλέχθηκαν κατά τη φάση της ανάλυσης, οδηγούμαστε στην ανάπτυξη του βασικού αρχιτεκτονικού σχεδιασμού για το σύστημα, ο οποίος περιγράφει το υλικό, λογισμικό και τη δικτυακή υποδομή που θα χρησιμοποιηθούν.

4.1 Αρχική υλοποίηση του πρωτοκόλλου

Για την αναπαράσταση του πρωτοκόλλου BAR, έχει δημιουργηθεί ένας πρωτότυπος διακομιστής BAR. Κάθε πελάτης συνδέεται και διατηρεί μία μόνιμη TCP σύνδεση με τον BAR server προκειμένου να στέλνει και να λαμβάνει όλη την κίνηση το δικτύου. Κάθε πελάτης στέλνει σε σταθερό ρυθμό κρυπτογραφημένα μηνύματα (για πραγματική επικοινωνία ή απλά θόρυβο) στον διακομιστή BAR. Ο διακομιστής με τη σειρά του μεταδίδει όλη την κίνηση που έλαβε στους χρήστες του δικτύου του, μέσω των εδραιωμένων συνδέσεων. Χρησιμοποιήθηκε το TCP πρωτόκολλο στο επίπεδο μεταφοράς, προκειμένου να προσομοιωθεί η χρήση του BAR σε πρωτόκολλα επιπέδου εφαρμογής βασισμένα στο TCP, όπως είναι το HTTP. Επίσης, η χρήση του TCP έχει ως αποτέλεσμα την ελαχιστοποίηση των πιθανοτήτων απώλειας πακέτων.

Η εφαρμογή [34] που υλοποιήθηκε, είναι γραμμένη σε γλώσσα Python και χρησιμοποιεί τη βιβλιοθήκη δικτύωσης Twisted [52]. Επίσης, χρησιμοποιείται SQLite για την αποθήκευση τοπικά των καταλόγων επαφών των χρηστών. Στην συγκεκριμένη εφαρμογή, τα κλειδιά που χρησιμοποιήθηκαν, καθώς και τα labels είναι τυχαία δημιουργημένα και θεωρείται ότι έχουν ανταλλαχθεί ήδη από τους χρήστες. Για την κρυπτογράφηση των κλειδιών χρησιμοποιήθηκε ο αλγόριθμος AES με μέγεθος 128-bit, ενώ τα labels δημιουργήθηκαν τυχαία και έχουν μέγεθος 32-bit.

Η αρχική εφαρμογή που υλοποιήθηκε, στοχεύει στην παρουσίαση της επικοινωνίας σε ένα σύστημα BAR, χωρίς να δίνει ιδιαίτερη έμφαση στην προετοιμασία των χρηστών, ώστε να χρησιμοποιήσουν το πρωτόκολλο. Επιπλέον, δεν καλύπτει τη δυνατότητα επικοινωνίας χρηστών που ανήκουν σε διαφορετικούς εξυπηρέτες. Στη συνέχεια, λοιπόν, θα αναλυθεί η εφαρμογή που υλοποιήθηκε στα πλαίσια της παρούσας διατριβής, η οποία επιτρέπει την εγγραφή των χρηστών στο σύστημα και την ανταλλαγή των απαραίτητων κλειδιών και labels για τη διασφάλισης της ανωνυμίας τους και της επικοινωνίας με άλλους χρήστες.



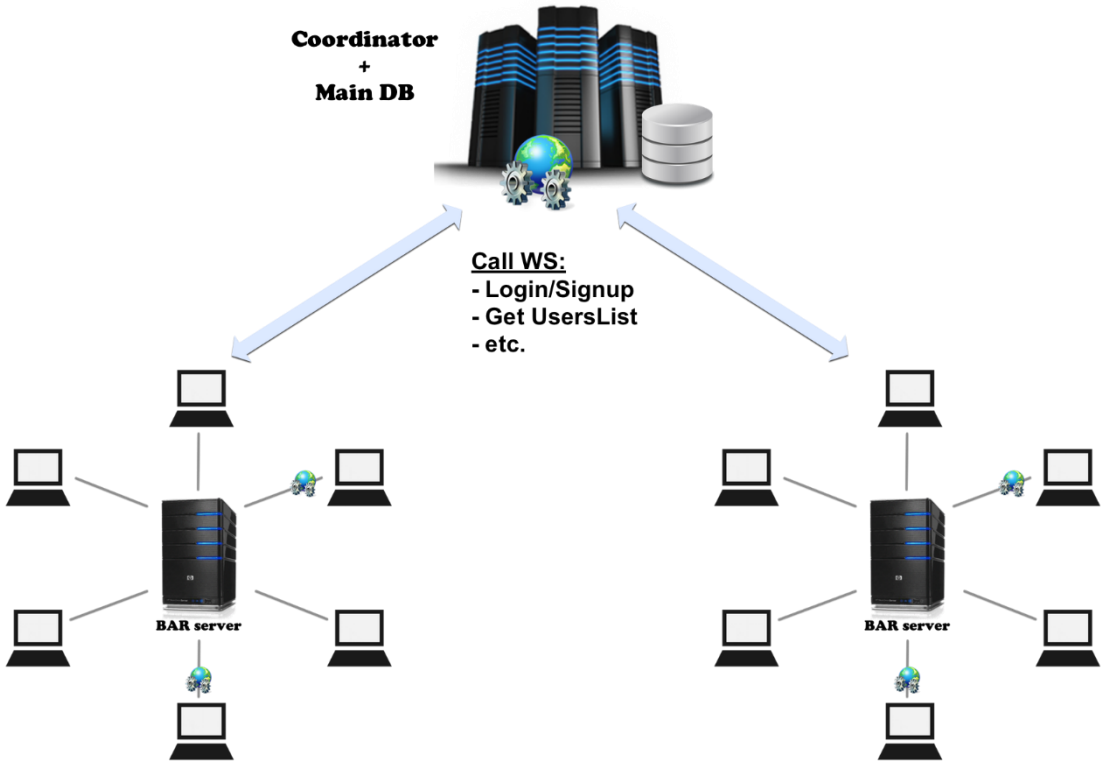
Εικόνα 4.1 Αρχιτεκτονική συστήματος στην προηγούμενη υλοποίηση

4.2 Αρχιτεκτονική

Η εφαρμογή [54] έχει υλοποιηθεί για να τρέχει μέσα από ένα πρόγραμμα περιήγησης (browser). Ο χρήστης ανοίγει με έναν browser την εφαρμογή, κάνει εγγραφή (register) ή σύνδεση (login), και στη συνέχεια, επιλέγει την υπηρεσία BAR (BAR service) την οποία επιθυμεί να χρησιμοποιήσει. Κατά τις διαδικασίες του login και register, η εφαρμογή έχει δημιουργήσει τα απαραίτητα κλειδιά και labels που χρειάζεται το πρωτόκολλο BAR για να διασφαλίσει την ανωνυμία του χρήστη.

Ο χρήστης της εφαρμογής στέλνει τα δεδομένα εισόδου μέσω Διαδικτύου στον Συντονιστή (BAR Coordinator). Τα δεδομένα αυτά μπορεί να είναι είτε κρυπτογραφημένες πληροφορίες για την είσοδό του στο σύστημα είτε αιτήσεις για την αποστολή των δημόσιων πληροφοριών που κρατάει ο διακομιστής και διασφαλίζουν την ομαλή λειτουργία του συστήματος BAR, όπως τον κατάλογο με τις παραμέτρους ή τις νέες επαφές. Ο Coordinator είναι ένας εξυπηρέτης (server) που περιλαμβάνει τη βάση δεδομένων και αναλαμβάνει την καταχώρηση και ανάκτηση δεδομένων από αυτή.

Παράλληλα, η εφαρμογή BarApp στέλνει δεδομένα μέσω Διαδικτύου και στον BAR server που ανήκει, προκειμένου στη συνέχεια, ο τελευταίος να τα μεταδώσει σε όλους τους υπόλοιπους χρήστες του συστήματος. Τα δεδομένα αυτά είναι κρυπτογραφημένες πληροφορίες που αποτελούν ένα μήνυμα για κάποιον παραλήπτη ή ψεύτικα μηνύματα που θα διασφαλίσουν την ανωνυμία της επικοινωνίας του. Ο BAR server αποτελεί έναν εξυπηρέτη (server), ο οποίος αναλαμβάνει να παραλάβει όλα τα κρυπτογραφημένα μηνύματα των χρηστών του και να τα μεταδώσει στο δίκτυο.



Εικόνα 4.2 Αρχιτεκτονική συστήματος

4.3 Ρόλοι του συστήματος

Κατά τη διάρκεια της ανάλυσης και του σχεδιασμού, καταγράφονται όλοι οι χρήστες που θα αλληλεπιδράσουν με το σύστημα, αλλά και οι ρόλοι τους σε αυτό. Έτσι, καταλήγουμε στους τέσσερις χρήστες που ακολουθούν.

- Διαχειριστής / administrator του Coordinator

- Διαχειριστής / administrator ενός BAR server
- Εγγεγραμμένος χρήστης της εφαρμογής BarApp
- Επισκέπτης του συστήματος / guest

Κάθε μία από τις οντότητες του συστήματος αναλύεται ξεχωριστά παρακάτω.

Ο διαχειριστής του Coordinator είναι η οντότητα που έχει τις περισσότερες δυνατότητες στο σύστημα. Μπορεί να διαχειρίζεται τη βάση δεδομένων και να ελέγχει τους χρήστες. Βέβαια, λόγω της δομής της βάσης και της κρυπτογράφησης που έχουν τα βασικότερα δεδομένα που στέλνονται από την εφαρμογή, ο διαχειριστής δεν έχει την ικανότητα να γνωρίζει τους κωδικούς των χρηστών ή αν κάποιος χρήστης έχει εισέλθει κάποια χρονική στιγμή στο σύστημα BAR και επικοινωνεί. Τέλος, αρμοδιότητά του είναι να απαντάει στα ερωτήματα των χρηστών και να επιλύει τα προβλήματα που αντιμετωπίζουν. Γενικά, είναι αυτός που ελέγχει τη σωστή λειτουργία του συστήματος. Όμως, η αλληλεπίδραση του πρώτου με αυτό είναι λιγότερη συχνή σε σχέση με τις άλλες οντότητες.

Ο διαχειριστής ενός BAR server είναι εκείνος που είναι υπεύθυνος για τη σωστή λειτουργία του εξυπηρέτη. Μπορεί να ελέγχει την κατάσταση του server (ενεργός ή μη) και να την τροποποιεί ανάλογα με τις ανάγκες που προκύπτουν. Επίσης, είναι αρμόδιος για την επίλυση προβλημάτων του BAR server που μπορεί να προκύψουν.

Ο εγγεγραμμένος χρήστης της BarApp εφαρμογής έχει το βασικότερο ρόλο στο σύστημα. Εξ ορισμού είναι ο χρήστης που χρησιμοποιεί την εφαρμογή, προκειμένου να αποστείλει μηνύματα ανώνυμα. Έχοντας εισέλθει στο σύστημα, ο συγκεκριμένος τύπος χρήστη μπορεί να επιλέξει κάποια από τις διαθέσιμες υπηρεσίες (BAR services) για να επικοινωνήσει. Επίσης, μπορεί να έρθει σε επικοινωνία με τον διαχειριστή του συστήματος μέσω της αντίστοιχης φόρμας. Επομένως, είναι ο χρήστης που αλληλεπιδρά πιο συχνά με το σύστημα και αντλεί τις περισσότερες πληροφορίες από αυτό.

Ο επισκέπτης του συστήματος είναι ο χρήστης με τη λιγότερη γνώση των ικανοτήτων της εφαρμογής, καθώς δεν μπορεί να αλληλεπιδράσει με αυτή. Ο συγκεκριμένος τύπος χρήστη δεν είναι εγγεγραμμένος στην εφαρμογή, με αποτέλεσμα οι δυνατότητές του να είναι περιορισμένες. Μπορεί να επιλέξει την σύνδεση ή εγγραφή του στην υπηρεσία BAR και να επικοινωνήσει με τον διαχειριστή του συστήματος.

4.4 Απαιτήσεις

Γνωρίζοντας πλέον ποιοι μπορούν να χρησιμοποιήσουν το σύστημα και να αλληλεπιδράσουν με αυτό, μπορούμε να καθορίσουμε τις απαιτήσεις που έχει κάθε μία από τις οντότητες του συστήματος.

Ο διαχειριστής του Coordinator μπορεί:

- Να διαχειρίζεται τη βάση δεδομένων.
- Να ελέγχει τη σωστή λειτουργία του Coordinator.
- Να ελέγχει τη σωστή λειτουργία της εφαρμογής από τους χρήστες.
- Να απαντάει στα ερωτήματα των χρηστών.
- Να επιλύει τα προβλήματά τους.

Ο διαχειριστής ενός BAR server μπορεί:

- Να εκκινεί / σταματά τον BAR server.
- Να ελέγχει τη σωστή λειτουργία του BAR server.
- Να επιλύει τυχόν προβλήματα που μπορεί να προκύψουν.

Ο εγγεγραμμένος χρήστης της BarApp εφαρμογής μπορεί:

- Να επικοινωνήσει ανώνυμα με κάποιο από τα διαθέσιμα BAR services.
- Να απενεργοποιήσει τον λογαριασμό του, σβήνοντας από τη βάση δεδομένων όσα στοιχεία τον αφορούν.
- Να επικοινωνήσει με τον διαχειριστή του συστήματος για βοήθεια.

Ο επισκέπτης της BarApp εφαρμογής μπορεί:

- Να εγγραφεί / εισέλθει στην υπηρεσία.
- Να επικοινωνήσει με τον διαχειριστή του συστήματος για βοήθεια.

4.5 Βάση δεδομένων του Coordinator

Μία σωστά δομημένη βάση δεδομένων παρέχει πρόσβαση σε ενημερωμένες και ακριβείς πληροφορίες. Επομένως, η σωστή σχεδίασή της είναι ουσιαστικής σημασίας για την υλοποίηση της εφαρμογής και την επίτευξη των στόχων. Έτσι, όταν δουλεύουμε με μία βάση δεδομένων, η επένδυση του χρόνου που απαιτείται για την εκμάθηση των αρχών της καλής σχεδίασης είναι σημαντική.

Υπάρχουν ορισμένες αρχές που καθοδηγούν τη διαδικασία της σχεδίασης βάσης δεδομένων. Η πρώτη αρχή είναι ότι οι διπλότυπες πληροφορίες (που επίσης ονομάζονται πλεονάζοντα δεδομένα) είναι κακές, διότι σπαταλούν χώρο και αυξάνουν την πιθανότητα σφαλμάτων και ασυνεπειών. Η δεύτερη αρχή είναι ότι η ορθότητα και η πληρότητα των πληροφοριών είναι σημαντικές. Εάν η βάση δεδομένων περιέχει λανθασμένες πληροφορίες, οι εκθέσεις που αντλούν πληροφορίες από τη βάση δεδομένων επίσης θα περιέχουν λανθασμένες πληροφορίες. Ως αποτέλεσμα, οι όποιες αποφάσεις στηρίζονται σε αυτές τις εκθέσεις θα είναι βασισμένες σε παραπληροφόρηση.

Συνεπώς, υπάρχουν κάποια κομβικά σημεία κατά τη διαδικασία σχεδίασης μιας βάσης δεδομένων:

- Ο καθορισμός του σκοπού που θα εξυπηρετεί η βάση δεδομένων.
- Η εύρεση και οργάνωση των απαιτούμενων πληροφοριών.
- Ο χωρισμός των πληροφοριών σε πίνακες.
- Η μετατροπή των στοιχείων πληροφοριών σε στήλες.
- Ο καθορισμός των πρωτευόντων κλειδιών.
- Η δημιουργία σχέσεων μεταξύ των πινάκων.
- Η βελτίωση της σχεδίασης από τυχόν σφάλματα.
- Η εφαρμογή των κανόνων κανονικοποίησης των δεδομένων.

Με βάση όλα τα παραπάνω και μετά το σχεδιασμό σε χαρτί των πινάκων που θα χρησιμοποιηθούν, το επόμενο βήμα είναι ο σχεδιασμός τους. Για τη σχεδίαση της βάσης χρησιμοποιήθηκε το phpMyAdmin. Για τη δημιουργία του διαγράμματος Οντοτήτων-Συσχετίσεων (ER diagram) χρησιμοποιήθηκε το Microsoft Visio 2013.

4.5.1 Πίνακας BAR_activeUsers

Ο πίνακας BAR_activeUsers δημιουργήθηκε για την καταχώρηση των ενεργών χρηστών του πρωτοκόλλου BAR.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
	userBarID	int(11)	Όχι	Καμία	
	ip	varchar(15)	Όχι	Καμία	
	bridgedPk	BLOB	Όχι	Καμία	

Εικόνα 4.3 Πίνακας BAR_activeUsers

userBarID

Πρωτεύον κλειδί του πίνακα, αποτελεί μία αριθμητική τιμή που προσδιορίζει έναν εγγεγραμμένο χρήστη της εφαρμογής BarApp, ο οποίος έχει εισέλθει στην εφαρμογή επιτυχώς και έχει λάβει ένα τυχαίο

αναγνωριστικό για τη χρήση του πρωτοκόλλου BAR. Ο τύπος δεδομένων είναι Integer(11), καθώς είναι ακέραιος αριθμός.

ip

Στο συγκεκριμένο πεδίο αποθηκεύεται η IP διεύθυνση του χρήστη. Η συγκεκριμένη εγγραφή χρησιμοποιείται για την επικοινωνία του χρήστη στο πρωτόκολλο BAR. Ο τύπος δεδομένων είναι Varchar(15).

bridgedPk

Το συγκεκριμένο πεδίο περιλαμβάνει το προσωρινό δημόσιο κλειδί που έχει δημιουργήσει ο χρήστης του BAR πρωτοκόλλου για την γεφύρωση των ανώνυμων επικοινωνιών μεταξύ των Clusters. Ο τύπος δεδομένων είναι BLOB.

4.5.2 Πίνακας BAR_captcha

Ο πίνακας BAR_captcha περιλαμβάνει κάποιες τυχαίες τιμές, οι οποίες χρησιμοποιούνται κατά τη διαδικασία της εγγραφής του χρήστη στην εφαρμογή BarApp ως επιπλέον μέτρο ασφάλειας για να μη γίνονται αυτοματοποιημένες επιθέσεις (bots), με σκοπό τη διακοπή της λειτουργίας της εφαρμογής.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
🔑	cID	int(11)	Όχι	Καμία	AUTO_INCREMENT
	captcha	varchar(6)	Όχι	Καμία	

Εικόνα 4.4 Πίνακας BAR_captcha

cID

Πρωτεύον κλειδί του πίνακα BAR_captcha, αποτελεί μία αριθμητική τιμή που υποδεικνύει το αναγνωριστικό μιας τιμής captcha στη βάση δεδομένων. Η τιμή της αυξάνεται κατά μία με την εισαγωγή νέου captcha. Ο τύπος δεδομένων είναι Integer(11), καθώς είναι ακέραιος αριθμός.

captcha

Το συγκεκριμένο πεδίο αποτελεί μία τυχαία τιμή captcha. Ο τύπος δεδομένων είναι Varchar(10) γιατί δύναται να πάρει μέχρι 10 χαρακτήρες, οι οποίοι μπορούν να είναι αριθμοί και γράμματα.

4.5.3 Πίνακας BAR_contacts

Ο πίνακας BAR_contacts δημιουργήθηκε για την καταχώρηση των νέων επαφών για κάθε χρήστη. Αποτελεί την δημόσιο κατάλογο, στον οποίο κάθε νέο εγγεγραμμένο μέλος του πρωτοκόλλου καταχωρεί τα κλειδιά και labels που θα ανταλλάξει με τους υπόλοιπους χρήστες του πρωτοκόλλου.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
🔑	pseudonym	varchar(10)	Όχι	Καμία	
🔑	data	BLOB	Όχι	Καμία	

Εικόνα 4.5 Πίνακας BAR_contacts

pseudonym

Το συγκεκριμένο πεδίο υποδεικνύει το ψευδώνυμο του χρήστη u_j που απευθύνεται. Ο τύπος δεδομένων είναι Varchar(10). Σε συνδυασμό με το πεδίο data αποτελούν τα πρωταρχικά κλειδιά του πίνακα.

data

Στο συγκεκριμένο πεδίο αποθηκεύονται κρυπτογραφημένα τα δεδομένα που ανταλλάσσουν ένας χρήστης u_i με κάποιον άλλο u_j κατά τη διαδικασία του Key Exchange. Τα δεδομένα που περιλαμβάνονται είναι: το ψευδώνυμο του χρήστη u_i, το δημόσιο κλειδί του pk_i, το κοινό κλειδί τους k_{ij}, το label l_{ij} και την υπογραφή του σ_{ij}. Ο τύπος δεδομένων είναι BLOB.

4.5.4 Πίνακας BAR_loginAttempts

Ο πίνακας BAR_loginAttempts χρησιμοποιείται για την καταχώρηση των λανθασμένων προσπαθειών εισόδου του χρήστη στην υπηρεσία. Στην περίπτωση που οι λανθασμένες προσπάθειες πρόσβασης ξεπεράσουν τις τρεις, ο λογαριασμός με το συγκεκριμένο userID λαμβάνει την ένδειξη locked.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
🔑	userID	int(11)	Όχι	Καμία	
	attempts	int(11)	Όχι	Καμία	

Εικόνα 4.6 Πίνακας BAR_loginAttempts

userID

Πρωτεύον κλειδί του πίνακα, αποτελεί μία αριθμητική τιμή που υποδεικνύει τον αριθμό του χρήστη στη βάση δεδομένων. Το συγκεκριμένο userID υπάρχει ήδη στον πίνακα BAR_users και αποτελεί δευτερεύον κλειδί (foreign key). Ο τύπος δεδομένων είναι Integer(11), καθώς είναι ακέραιος αριθμός.

attempts

Το συγκεκριμένο πεδίο περιλαμβάνει το πλήθος των αποτυχημένων προσπαθειών εισόδου ενός χρήστη. Ο τύπος δεδομένων του πεδίου είναι Integer(11), καθώς δύναται να πάει μόνο αριθμητικές τιμές.

4.5.5 Πίνακας BAR_nymUsers

Ο πίνακας BAR_nymUsers περιλαμβάνει όλους τους εγγεγραμμένους χρήστες του BAR πρωτοκόλλου. Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία που χρησιμοποιεί ένας χρήστης στο σύστημα BAR για τη διασφάλιση της ανωνυμίας του και δεν σχετίζεται με την πραγματική του ταυτότητα.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
🔑	userID	int(11)	Όχι	Καμία	
	pseudonym	varchar(20)	Όχι	Καμία	UNIQUE
	pk	blob	Όχι	Καμία	UNIQUE
	sig	blob	Όχι	Καμία	

Εικόνα 4.7 Πίνακας BAR_nymUsers

userID

Πρωτεύον κλειδί του πίνακα, αποτελεί μία αριθμητική τιμή που υποδεικνύει τον αριθμό του χρήστη στη βάση δεδομένων. Το συγκεκριμένο userID υπάρχει ήδη στον πίνακα BAR_users και αποτελεί δευτερεύον κλειδί (foreign key) του παρόντος πίνακα. Ο τύπος δεδομένων είναι Integer(11), καθώς είναι ακέραιος αριθμός.

pseudonym

Το συγκεκριμένο πεδίο περιλαμβάνει το ψευδώνυμο που χρησιμοποιεί κάποιος χρήστης για την ανώνυμη επικοινωνία στο πρωτόκολλο BAR. Το ψευδώνυμο δεν αποκαλύπτει στοιχεία της πραγματικής ταυτότητας του χρήστη. Ο τύπος δεδομένων είναι Varchar(20).

pk

Στο συγκεκριμένο πεδίο αποθηκεύεται το δημόσιο κλειδί του χρήστη, το οποίο έχει υποστεί προηγουμένως κωδικοποίηση. Το pk αποθηκεύεται στη βάση με μορφή BLOB (Binary Large Object). Ο συγκεκριμένος τύπος χρησιμοποιείται για την αποθήκευση των δυαδικών δεδομένων ως μία ενιαία οντότητα.

sig

Στο συγκεκριμένο πεδίο κρατάται η υπογραφή του Coordinator για τα δεδομένα του χρήστη, προκειμένου να μπορεί ο οποιοσδήποτε να επαληθεύσει την εγκυρότητά τους. Το sig αποθηκεύεται στη βάση με μορφή BLOB (Binary Large Object).

4.5.6 Πίνακας BAR_systemParams

Ο πίνακας BAR_systemParams αποτελεί τον κατάλογο με τις παραμέτρους του συστήματος, οι οποίες πρέπει να γνωστοποιούνται σε όλους τους χρήστες του πρωτοκόλλου BAR.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
🔑	type	varchar(20)	Όχι	Καμία	
	param	varchar(128)	Όχι	Καμία	

Εικόνα 4.8 Πίνακας BAR_systemParams

type

Πρωτεύον κλειδί του πίνακα, περιλαμβάνει το όνομα του τύπου της παραμέτρου. Ο τύπος δεδομένων είναι Varchar(20).

param

Στο συγκεκριμένο πεδίο καταχωρείται η τιμή της παραμέτρου, που καθορίζει το πεδίο type. Ο τύπος δεδομένων που χρησιμοποιήθηκε είναι Varchar(128).

4.5.7 Πίνακας BAR_users

Ο πίνακας BAR_users δημιουργήθηκε για την καταχώρηση των χρηστών της BarApp εφαρμογής στο σύστημα.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
🔑	userID	int(11)	Όχι	Καμία	AUTO_INCREMENT
	username	varchar(10)	Όχι	Καμία	
	password	varchar(255)	Όχι	Καμία	
	encryptSalt	varchar(128)	Όχι	Καμία	
	email	varchar(100)	Όχι	Καμία	
	birthdate	date	Όχι	Καμία	
	locked	tinyint(1)	Όχι	Καμία	

Εικόνα 4.9 Πίνακας BAR_users

userID

Πρωτεύον κλειδί του πίνακα, αποτελεί μία αριθμητική τιμή που υποδεικνύει τον αριθμό του χρήστη στη βάση δεδομένων. Η τιμή της αυξάνεται κατά μία με την εισαγωγή νέου χρήστη. Ο τύπος δεδομένων είναι Integer(11), καθώς είναι ακέραιος αριθμός.

username

Το συγκεκριμένο πεδίο αποτελεί το αναγνωριστικό που χρησιμοποιεί ο εγγεγραμμένος χρήστης της εφαρμογής κατά την είσοδό του. Ο τύπος δεδομένων είναι Varchar(10) γιατί δύναται να πάρει μέχρι 10 χαρακτήρες, οι οποίοι μπορούν να είναι αριθμοί και γράμματα.

password

Στο συγκεκριμένο πεδίο αποθηκεύεται ο κωδικός του κάθε χρήστη, ο οποίος είναι προσωπικός για τον καθένα και τον γνωρίζει μόνο αυτός. Για λόγους ασφαλείας, δεν αποθηκεύεται ο κωδικός στη βάση δεδομένων, αλλά το ψηφιακό αποτύπωμα που δημιουργείται από τη συνάρτηση σύνοψης SHA-256. Ο τύπος δεδομένων είναι Varchar(255).

encryptSalt

Το συγκεκριμένο πεδίο αποτελεί την επιπλέον πληροφορία που χρησιμοποιείται από τη μονόδρομη συνάρτηση σύνοψης SHA-256. Χρειάζεται ώστε να προστατεύσει τον κωδικό του χρήστη και από “επιθέσεις λεξικών” (dictionary attacks), πέρα από τις υπόλοιπες επιθέσεις που προστατεύει η ίδια η συνάρτηση σύνοψης. Το salt έχει δυαδική μορφή αλλά έχει οριστεί να αποθηκεύεται στη βάση με μορφή αλφαριθμητικού. Έτσι, ο τύπος δεδομένων του συγκεκριμένου πεδίου είναι Varchar(128).

email

Αποθηκεύει το email του εγγεγραμμένου χρήστη που θα χρησιμοποιηθεί για την επικοινωνία του με τον διαχειριστή του συστήματος και μελλοντικά, για την ανάκτηση του κωδικού πρόσβασης. Είναι ξεχωριστό για κάθε χρήστη και επειδή ο τύπος δεδομένων είναι αλφαριθμητικό, χρησιμοποιήθηκε Varchar(100).

birthdate

Στο συγκεκριμένο πεδίο κρατάται η ημερομηνία γέννησης του χρήστη. Το συγκεκριμένο πεδίο θα χρησιμοποιηθεί μελλοντικά για τη λειτουργία ανάκτησης του κωδικού πρόσβασης. Περιλαμβάνει δεδομένα τύπου Date γιατί αποθηκεύεται ημερομηνία.

locked

Στο συγκεκριμένο πεδίο αποθηκεύεται μία τιμή true-false, η οποία χρησιμοποιείται για να προσδιορίσει αν ο λογαριασμός του χρήστη είναι κλειδωμένος. Ο λογαριασμός κλειδώνεται μετά από τρεις αποτυχημένες προσπάθειες εισόδου του χρήστη στο σύστημα. Για το ξεκλείδωμα, θα χρησιμοποιηθεί η διαδικασία ανάκτησης κωδικού που θα υλοποιηθεί μελλοντικά. Ο τύπος δεδομένων του συγκεκριμένου πεδίου είναι Λογικός (Boolean) και αρχικά ορίζεται ως false.

4.6 Βάση δεδομένων της εφαρμογής BarApp

Πέρα από τη βάση δεδομένων που χρησιμοποιεί ο Coordinator και περιλαμβάνει δεδομένα και παραμέτρους, απαραίτητα για την ομαλή λειτουργία του πρωτοκόλλου BAR, είναι σημαντική η υλοποίηση και μίας επιπλέον βάσης δεδομένων. Η συγκεκριμένη βάση αρχικοποιείται κατά την πρώτη λειτουργία της εφαρμογής BarApp και αποθηκεύει τοπικά κάποια στοιχεία που χρησιμοποιεί η εφαρμογή, αλλά και δεδομένα του χρήστη που διασφαλίζουν την ανωνυμία του στο πρωτόκολλο BAR.

Η σχεδίαση της βάσης γίνεται μέσα από την εφαρμογή χρησιμοποιώντας εντολές της βιβλιοθήκης SQLite. Για τη δημιουργία του διαγράμματος Οντοτήτων-Συσχετίσεων (ER diagram) και για αυτή τη βάση χρησιμοποιήθηκε το Microsoft Visio 2013.

4.6.1 Πίνακας BAR_UserContacts

Ο συγκεκριμένος πίνακας δημιουργήθηκε κατά την πρώτη υλοποίηση του πρωτοκόλλου BAR και χρησιμοποιήθηκε και για την εφαρμογή BarApp. Στον πίνακα BAR_UserContacts γίνεται η καταχώρηση των επαφών που έχει ο κάθε χρήστης. Η ενημέρωση των νέων επαφών του χρήστη πραγματοποιείται κατά την επιτυχή είσοδό του στο σύστημα.

	Όνομα	Τύπος	Επιπλέον
🔑	id	integer	AUTOINCREMENT
	name	text	
	label	text	
	publickey	text	
	sharedkey	text	

Εικόνα 4.10 Πίνακας BAR_UserContacts

id

Πρωτεύον κλειδί του πίνακα, αποτελεί μία αριθμητική τιμή που υποδεικνύει το αναγνωριστικό μίας επαφής στη βάση δεδομένων. Η τιμή της αυξάνεται κατά μία με την εισαγωγή νέας επαφής. Ο τύπος δεδομένων είναι Integer, καθώς είναι ακέραιος αριθμός.

name

Στο συγκεκριμένο πεδίο αποθηκεύεται το ψευδώνυμο του χρήστη u_j. Ο τύπος δεδομένων είναι Text.

label

Στο συγκεκριμένο πεδίο καταχωρείται το label που έχουν ανταλλάξει οι δύο χρήστες, το οποίο ανανεώνεται πριν την νέα επικοινωνία τους. Ο τύπος δεδομένων είναι Text.

publickey

Το πεδίο χρησιμοποιείται για να προσδιορίσει το κωδικοποιημένο δημόσιο κλειδί του χρήστη u_j. Ο τύπος δεδομένων είναι Text.

sharedkey

Το συγκεκριμένο πεδίο περιλαμβάνει το κλειδί που έχουν ανταλλάξει οι δύο χρήστες, το οποίο ανανεώνεται πριν την επικοινωνία τους μαζί με το label. Ο τύπος δεδομένων είναι Text.

4.6.2 Πίνακας BAR_SiteMenu

Στην εφαρμογή BarApp το μενού παρουσιάζεται δυναμικά και ανάλογα με την κατάσταση εισόδου του χρήστη (συνδεδεμένος ή μη). Έτσι, ο πίνακας BAR_SiteMenu περιέχει τα δεδομένα του μενού που χρησιμοποιεί η εφαρμογή.

	Όνομα	Τύπος
🔑	entry	text
	position	integer
	link	text
	not_connected	integer

Εικόνα 4.11 Πίνακας BAR_SiteMenu

entry

Πρωτεύον κλειδί του πίνακα, περιλαμβάνει το όνομα της εγγραφής, όπως παρουσιάζεται στο μενού της εφαρμογής. Ο τύπος δεδομένων είναι Text, καθώς μπορεί να λάβει αλφαριθμητικά δεδομένα.

position

Στο συγκεκριμένο πεδίο αποθηκεύεται η θέση που θα παρουσιάζεται η κάθε εγγραφή στο μενού της εφαρμογής. Ο τύπος δεδομένων είναι Integer.

link

Στο συγκεκριμένο πεδίο καταχωρείται ο υπερσύνδεσμος (link) που ενεργοποιείται από την κάθε εγγραφή του μενού. Ο τύπος δεδομένων είναι Text.

not_connected

Το πεδίο χρησιμοποιείται για να προσδιορίσει ποιες εγγραφές του μενού θα εμφανιστούν, ανάλογα με το αν ο χρήστης είναι συνδεδεμένος ή μη. Έτσι, τα στοιχεία του μενού που εμφανίζονται σε όλες τις καταστάσεις , έχουν λάβει την τιμή 0. Οι εγγραφές του μενού που εμφανίζονται σε μη εγγεγραμμένους χρήστες (πχ. Login) έχουν την τιμή 1, ενώ αυτές που ισχύουν για συνδεδεμένους χρήστες (πχ. Services) έχουν λάβει την τιμή 2. Ο τύπος δεδομένων είναι Integer.

4.6.3 Πίνακας BAR_TempBarID

Στον συγκεκριμένο πίνακα αποθηκεύεται η τιμή του BarID που έχει λάβει προσωρινά ένας εγγεγραμμένος χρήστης του πρωτοκόλλου. Η συγκεκριμένη εγγραφή δημιουργείται με την είσοδο του χρήστη στην εφαρμογή και διαγράφεται με την αποσύνδεσή του.

	Όνομα	Τύπος
	userBarID	integer

Εικόνα 4.12 Πίνακας BAR_TempBarID

userBarID

Πρωτεύον κλειδί του πίνακα BAR_captcha, αποτελεί μία αριθμητική τιμή που υποδεικνύει το αναγνωριστικό το προσωρινό αναγνωριστικό που έχει λάβει ένας χρήστης για την ανώνυμη περιήγησή του στο πρωτόκολλο BAR. Ο τύπος δεδομένων είναι Integer(11), καθώς είναι ακέραιος αριθμός.

4.6.4 Πίνακας BAR_TempKeys

Ο πίνακας BAR_TempKeys δημιουργήθηκε για την καταχώρηση του προσωρινού ζεύγους δημόσιου-ιδιωτικού κλειδιού για κάθε χρήστη. Τα συγκεκριμένα κλειδιά αποθηκεύονται προσωρινά με την σύνδεση του χρήστη στην εφαρμογή, ισχύουν για μία συνεδρία και στη συνέχεια αντικαθίστανται.

	Όνομα	Τύπος
	sessionPk	text
	sessionSk	text

Εικόνα 4.13 Πίνακας BAR_TempKeys

sessionPk

Το συγκεκριμένο πεδίο υποδεικνύει το δημόσιο κλειδί συνόδου που κατέχει ένας χρήστης. Ο τύπος δεδομένων είναι Text, καθώς περιλαμβάνει την τιμή του κλειδιού κωδικοποιημένη.

sessionSk

Αντίστοιχα, στο συγκεκριμένο πεδίο περιέχεται η τιμή του προσωρινού ιδιωτικού κλειδιού του χρήστη κωδικοποιημένη. Και στο παρόν πεδίο, ο τύπος δεδομένων είναι Text.

4.6.5 Πίνακας BAR_UserInfo

Ο πίνακας BAR_UserInfo χρησιμοποιείται για την καταχώρηση των στοιχείων ενός χρήστη, απαραίτητων για τη χρήση του πρωτοκόλλου BAR. Τα στοιχεία αυτά δημιουργούνται κατά την εγγραφή του χρήστη στην εφαρμογή και δε δύναται να αλλάξουν.

	Όνομα	Τύπος
	nym	text
	pk	text
	sk	text

Εικόνα 4.14 Πίνακας BAR_UserInfo

nym

Το συγκεκριμένο πεδίο περιλαμβάνει το ψευδώνυμο που χρησιμοποιεί ο χρήστης στο πρωτόκολλο BAR για την ανώνυμη επικοινωνία του. Ο τύπος δεδομένων είναι Text, καθώς μπορεί να λάβει αλφαριθμητικές τιμές.

pk

Στο πεδίο αποθηκεύεται κωδικοποιημένο το δημόσιο κλειδί του χρήστη, το οποίο έχει δημιουργηθεί κατά την εγγραφή του. Ο τύπος δεδομένων είναι Text.

sk

Αντίστοιχα, στο συγκεκριμένο πεδίο αποθηκεύεται κωδικοποιημένο το ιδιωτικό κλειδί του χρήστη, το οποίο έχει δημιουργηθεί κατά την εγγραφή του. Ο τύπος δεδομένων είναι Text.

4.7 Πλατφόρμα ανάπτυξης βάσης δεδομένων

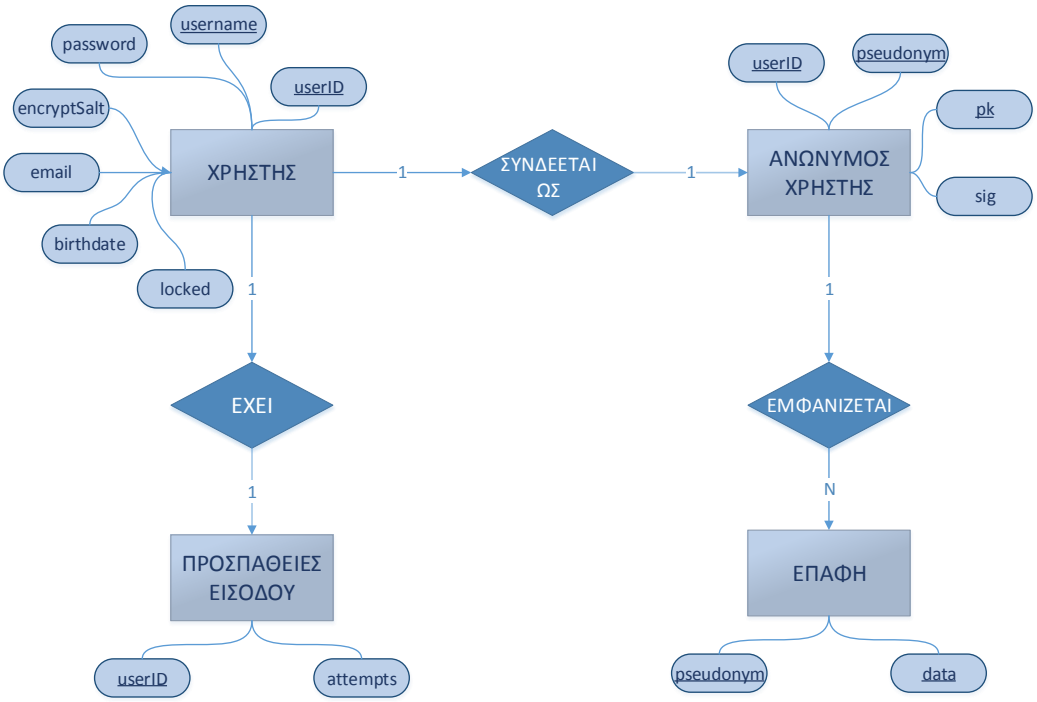
Για τη σχεδίαση της βάσης δεδομένων του Coordinator, λόγω ευκολίας στη χρήση και των δυνατοτήτων του χρησιμοποιήθηκε το phpMyAdmin, το οποίο παρείχε και έλεγχο στη λειτουργικότητα της βάσης. Επίσης, εφαρμόστηκε σε κάποια σημεία η γλώσσα προγραμματισμού sql που παρέχεται ως δυνατότητα του phpMyAdmin.

Αντίστοιχα, για τη σχεδίασης της τοπικής βάσης δεδομένων που χρησιμοποιεί η εφαρμογή BarApp, η υλοποίηση έγινε στο πρόγραμμα χρησιμοποιώντας τη βιβλιοθήκη της SQLite. Ο έλεγχος της σχεδίασης και της λειτουργικότητας της βάσης πραγματοποιήθηκε χρησιμοποιώντας το πρόγραμμα DbSchema της εταιρίας Wise Coders Solutions.

4.8 Διάγραμμα οντοτήτων-συσχετίσεων

Παρακάτω, θα παρουσιαστούν τα διαγράμματα οντοτήτων-συσχετίσεων (διάγραμμα Ο/Σ - ER diagram) για τις δύο βάσεις δεδομένων που έχουν αναπτυχθεί.

4.8.1 Βάση δεδομένων του Coordinator



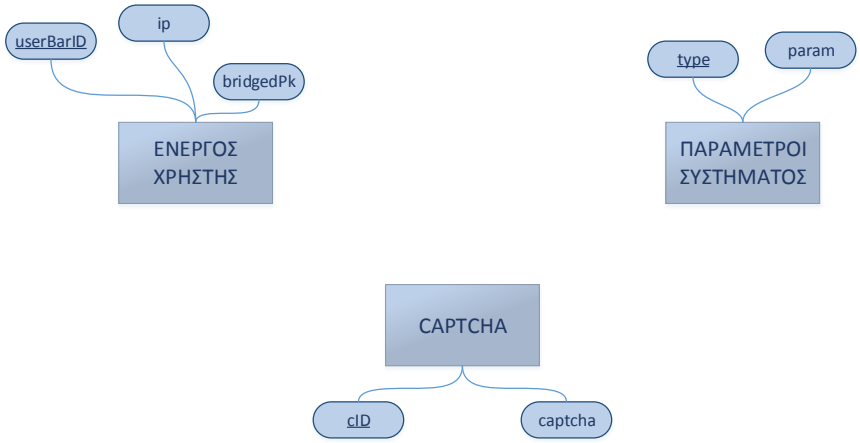
Εικόνα 4.15 Διάγραμμα Ο/Σ του Coordinator

Δημιουργώντας το διάγραμμα Ο/Σ παρατηρείται ότι κάποιοι πίνακες της βάσης του Coordinator δεν μπορούν να συσχετιστούν με τους υπόλοιπους.

Ο πίνακας BAR_activeUsers (Ενεργός χρήστης) δεν πρέπει να συσχετιστεί με κάποιον από τους άλλους πίνακες προκειμένου να διασφαλίζεται η ανωνυμία του χρήστη στο πρωτόκολλο BAR, ακόμα και από τον ίδιο τον Coordinator.

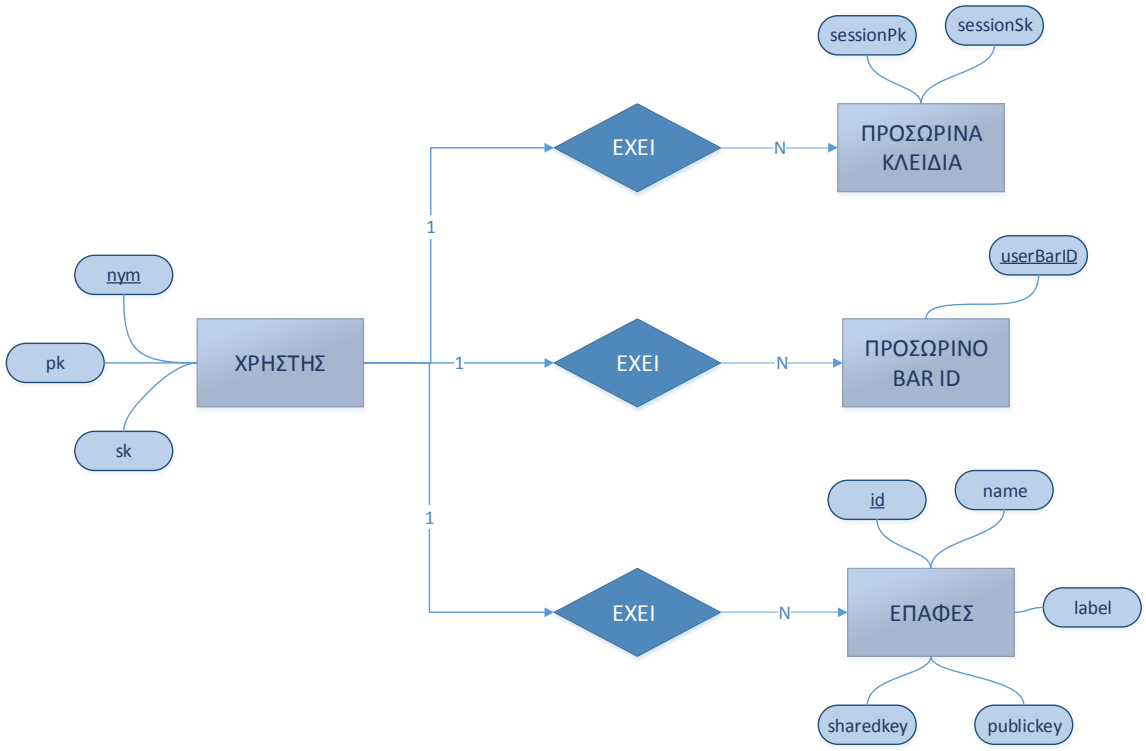
Ο πίνακας BAR_systemParams (Παράμετροι συστήματος) αποτελεί τον δημόσιο κατάλογο του Coordinator και εξασφαλίζει την ομαλή λειτουργία του πρωτοκόλλου BAR.

Ο πίνακας BAR_captcha χρησιμοποιείται από την εφαρμογή BarApp για τη προστασία της από επιθέσεις κατά τη διαδικασία της εγγραφής του χρήστη.



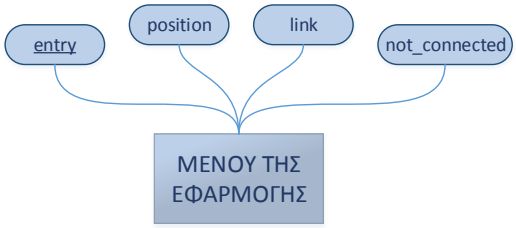
Εικόνα 4.16 Οντότητες της βάσης του Coordinator που δεν συσχετίζονται

4.8.2 Βάση δεδομένων της εφαρμογής BarApp



Εικόνα 4.17 Διάγραμμα Ο/Σ της εφαρμογής BarApp

Και σε αυτή την περίπτωση, δημιουργώντας το διάγραμμα Ο/Σ παρατηρείται ότι ο πίνακας BAR_SiteMenu δεν μπορεί να συσχετιστεί με τους υπόλοιπους. Αυτό συμβαίνει διότι ο συγκεκριμένος πίνακας δεν σχετίζεται με τα στοιχεία του χρήστη που απαιτούνται από το πρωτόκολλο για τη διασφάλιση της ανωνυμίας του, αλλά με τη λειτουργία της εφαρμογής.

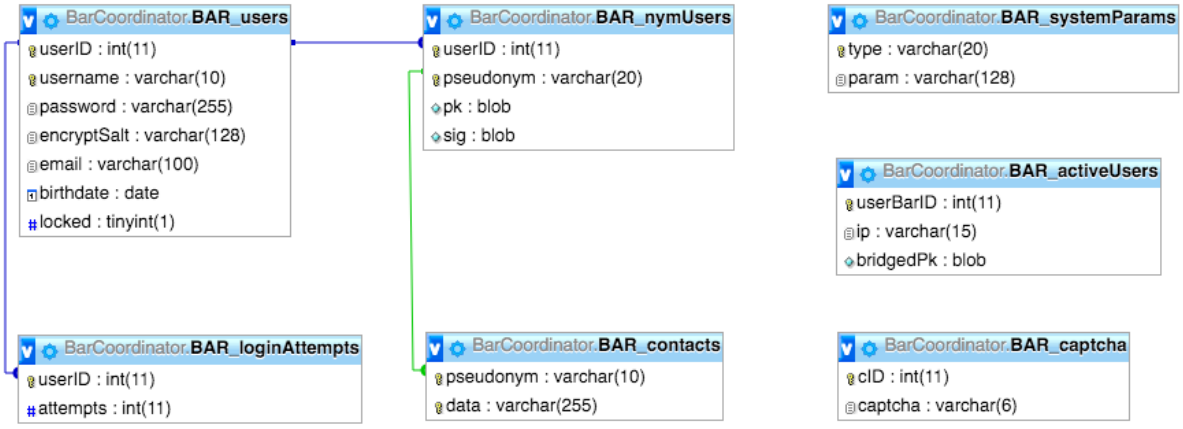


Εικόνα 4.18 Οντότητες της βάσης της BarApp που δεν συσχετίζονται

4.9 Προσχέδιο βάσης δεδομένων

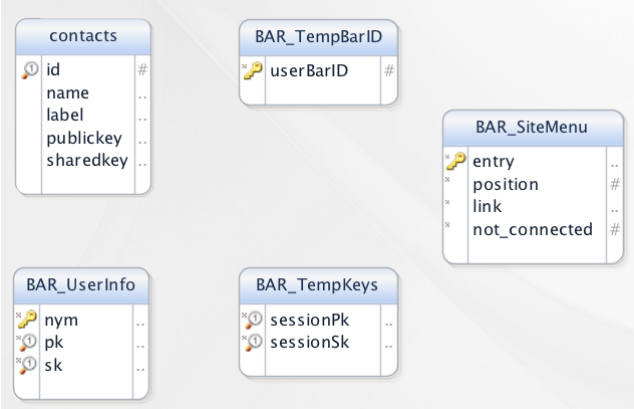
Παρακάτω, παρουσιάζεται ένα προσχέδιο για τις δύο βάσεις δεδομένων, όπως εμφανίστηκε στα προγράμματα που χρησιμοποιήθηκαν για την επεξεργασία τους.

4.9.1 Βάση δεδομένων του Coordinator



Εικόνα 4.19 Προσχέδιο βάσης δεδομένων Coordinator

4.9.2 Βάση δεδομένων της εφαρμογής BarApp



Εικόνα 4.20 Προσχέδιο βάσης δεδομένων BarApp

Κεφάλαιο 5

Επέκταση της υλοποίησης: Η εφαρμογή BarApp

Η εφαρμογή BarApp [54] δημιουργήθηκε με σκοπό να παρέχει στον χρήστη έναν εύκολο, αλλά συγχρόνως λειτουργικό τρόπο να χρησιμοποιήσει το πρωτόκολλο BAR. Πλέον, όλες οι απαραίτητες διαδικασίες που απαιτούνται κατά την εγγραφή ή την είσοδο στην υπηρεσία γίνονται αυτοματοποιημένα από την εφαρμογή, χωρίς να χρειάζεται η διαχείρισή τους από τον χρήστη.

Η εφαρμογή έχει υλοποιηθεί σε μορφή ιστότοπου (site), ώστε να τρέχει μέσα από ένα πρόγραμμα περιήγησης (browser). Με τη δημιουργία της, στοχεύει στην επέκταση της ήδη υπάρχουσας υλοποίησης του συστήματος BAR, καθώς και στην ανάπτυξη των επιμέρους λειτουργιών που απαιτούνται για την προετοιμασία των χρηστών, ώστε να είναι σε θέση να χρησιμοποιήσουν το σύστημα. Έτσι, έχουν προστεθεί οι νέες οντότητες που απαιτούνται (πχ. Coordinator και βάση δεδομένων) και έχουν διαμορφωθεί τα επιμέρους πρωτόκολλα που έχουν περιγραφεί προηγουμένως (εγγραφή χρήστη, είσοδο χρήστη, ανταλλαγή κλειδιών, BCP). Τέλος, έχει αναπτυχθεί στην εφαρμογή ένα γραφικό περιβάλλον, το οποίο επιτρέπει στον χρήστη μία εύκολη και συγχρόνως ευχάριστη ανώνυμη περιήγηση.

Έτσι, η δημιουργία της εφαρμογής BarApp, σε συνδυασμό με την ήδη υπάρχουσα υλοποίηση διασφαλίζουν την ολοκλήρωση των διαδικασιών που απαιτούνται για την επίτευξη της ανώνυμης επικοινωνίας σε ένα σύστημα BAR

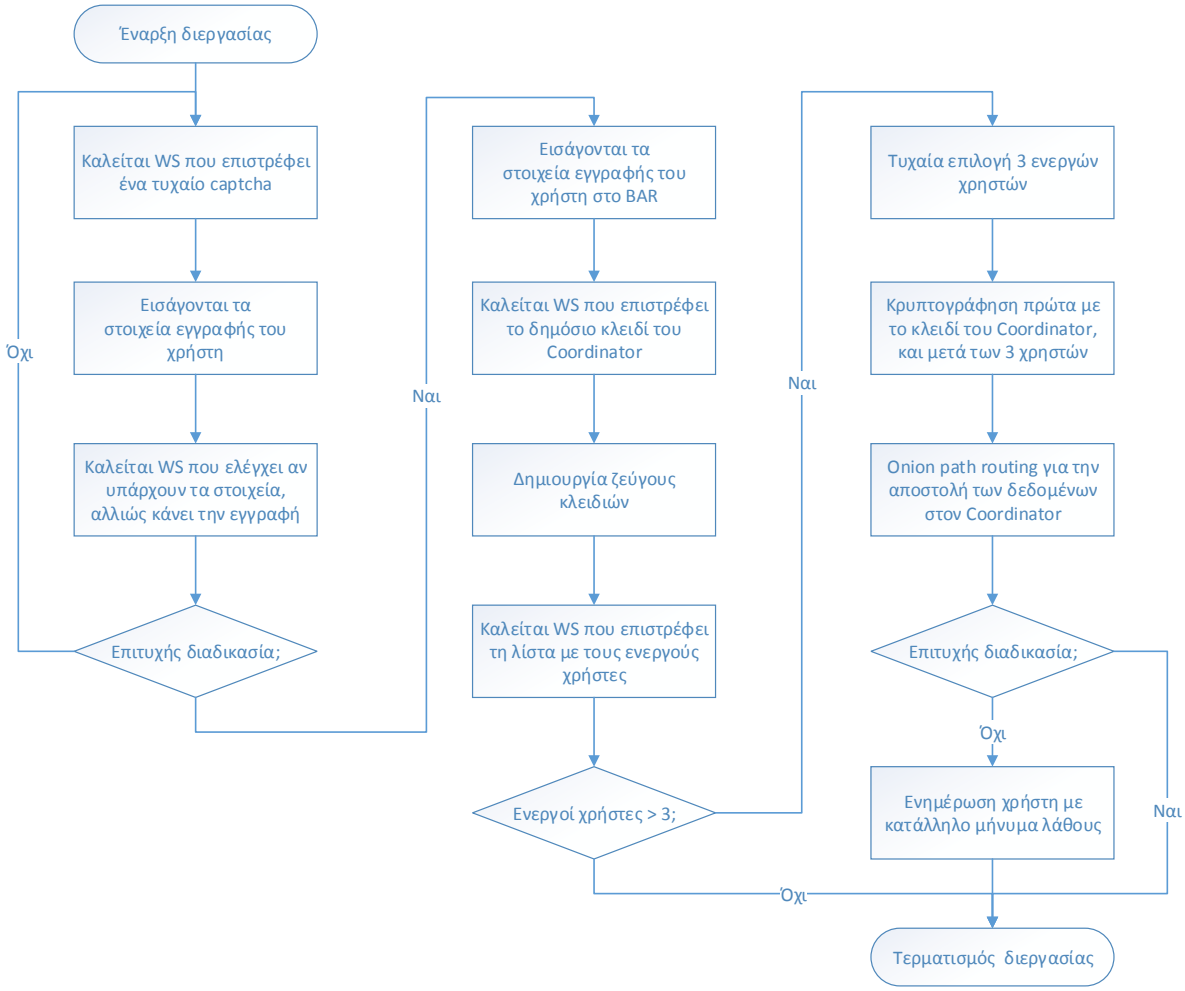
Σε αυτό το κεφάλαιο θα γίνει παρουσίαση της λειτουργίας της εφαρμογής και θα αναλυθούν μερικά από τα πιο σημαντικά στάδια της υλοποίησής της.

5.1 Διαγράμματα λειτουργίας BarApp

Οι βασικές λειτουργίες που καλύπτει η εφαρμογή BarApp αναφέρονται παρακάτω:

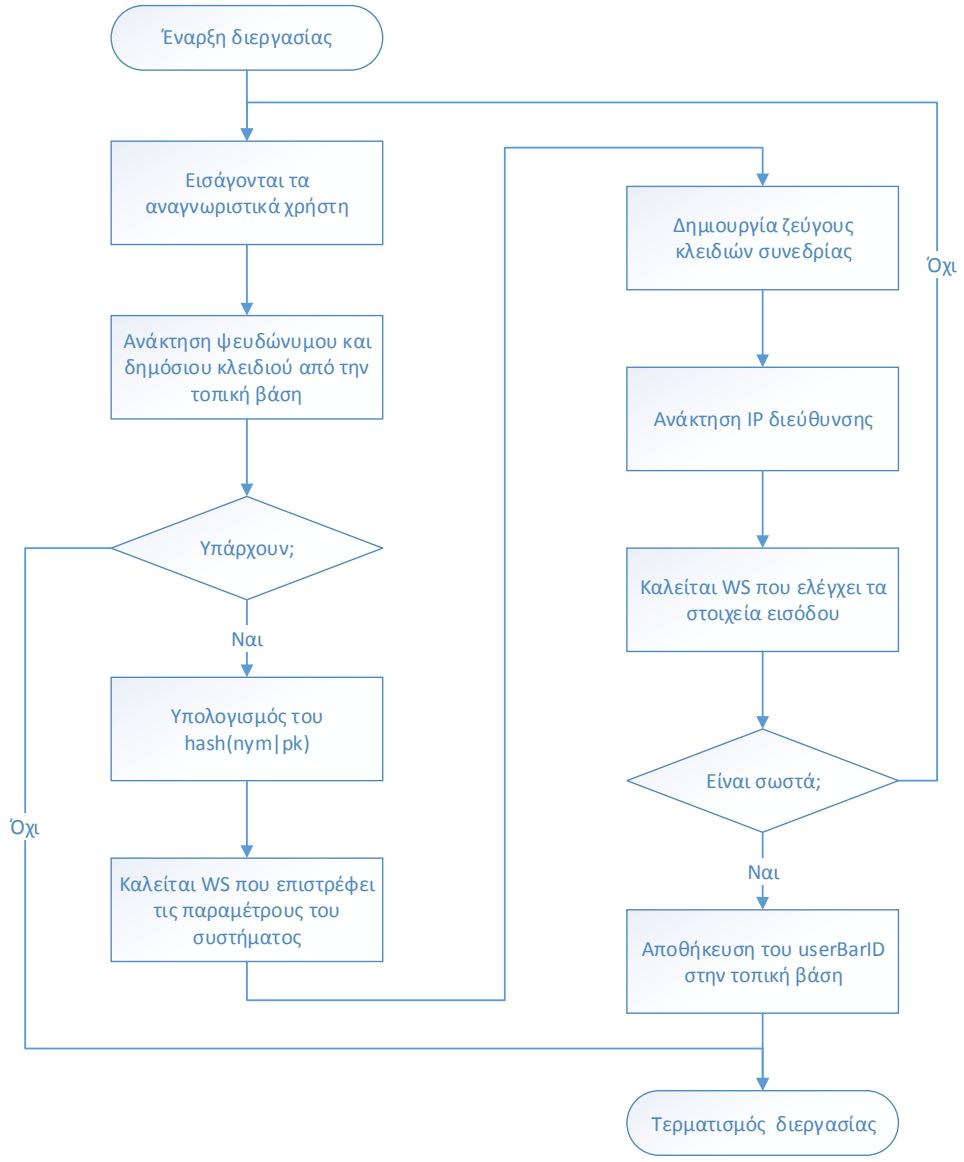
- Εγγραφή χρήστη (User Registration)
- Σύνδεση χρήστη (User Login)
- Ανταλλαγή Κλειδιών (Key Exchange)
- Κεντρική οθόνη με τα διαθέσιμα BAR services
- Επικοινωνία BAR με χρήστη του ίδιου Cluster
- Επικοινωνία BAR με χρήστη διαφορετικού Cluster
- Αποσύνδεση χρήστη (Logout)

5.1.1 Πρωτόκολλο εγγραφής χρήστη (User registration protocol)



Εικόνα 5.1 Διάγραμμα: Εγγραφή χρήστη

5.1.2 Πρωτόκολλο σύνδεσης χρήστη (User login protocol)

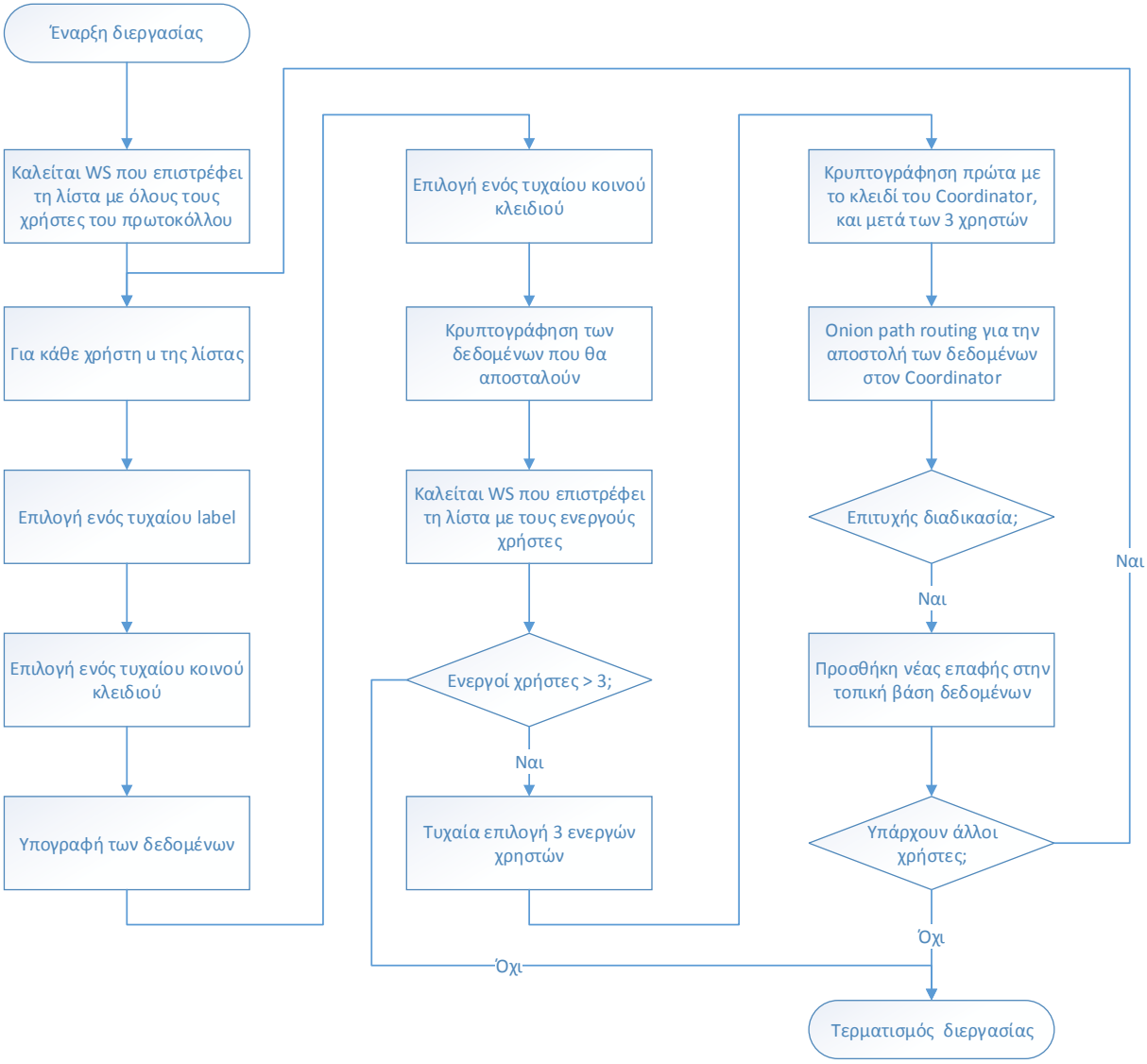


Εικόνα 5.2 Διάγραμμα: Σύνδεση χρήστη

5.1.3 Πρωτόκολλο ανταλλαγής κλειδιών (Key exchange protocol)

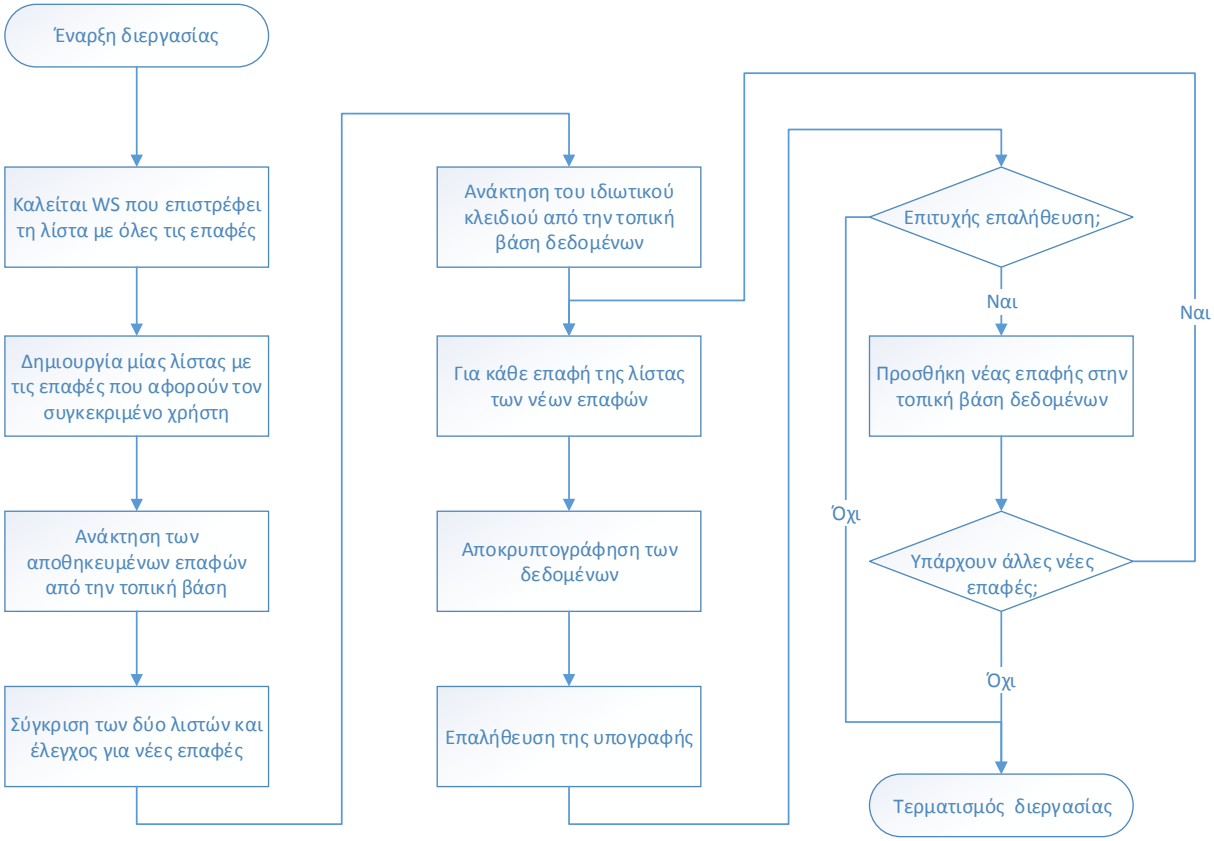
Όπως αναφέρθηκε και στα προηγούμενα κεφάλαια, το πρωτόκολλο ανταλλαγής κλειδιών πραγματοποιείται σε δύο διαφορετικές καταστάσεις. Έτσι, παρακάτω θα παρουσιαστούν τα διαγράμματα ροής για τις περιπτώσεις που ένας χρήστης έχει μόλις εγγραφεί στο πρωτόκολλο και δημιουργεί αιτήματα για νέες επαφές και όταν ένας ήδη εγγεγραμμένος χρήστης εισέρχεται στο σύστημα και ελέγχει για νέες επαφές.

- *Για έναν νέο εγγεγραμμένο χρήστη:*



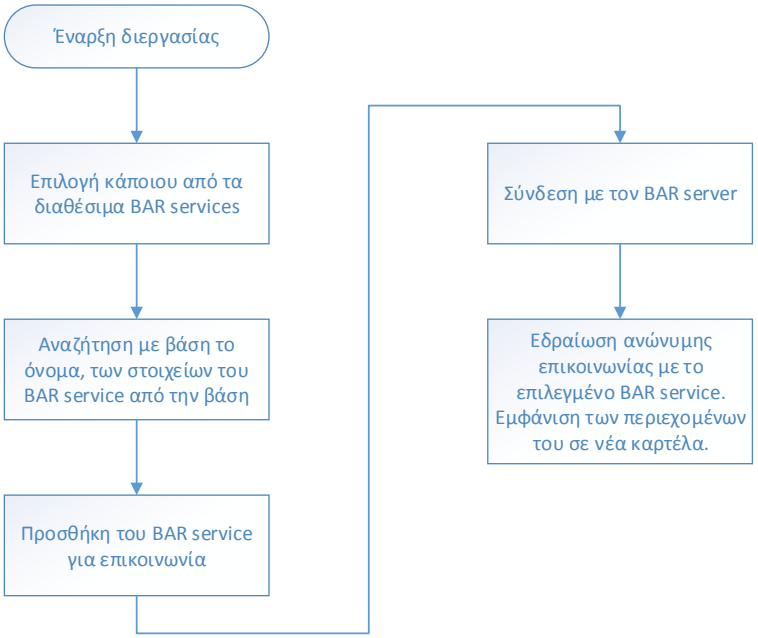
Εικόνα 5.3 Διάγραμμα: Ανταλλαγή κλειδιών (για νέους χρήστες)

- *Για υφιστάμενους χρήστες του συστήματος BAR*



Εικόνα 5.4 Διάγραμμα: Ανταλλαγή κλειδιών (για υφιστάμενους χρήστες)

5.1.4 Κεντρική οθόνη της εφαρμογής (Διαθέσιμα BAR services)



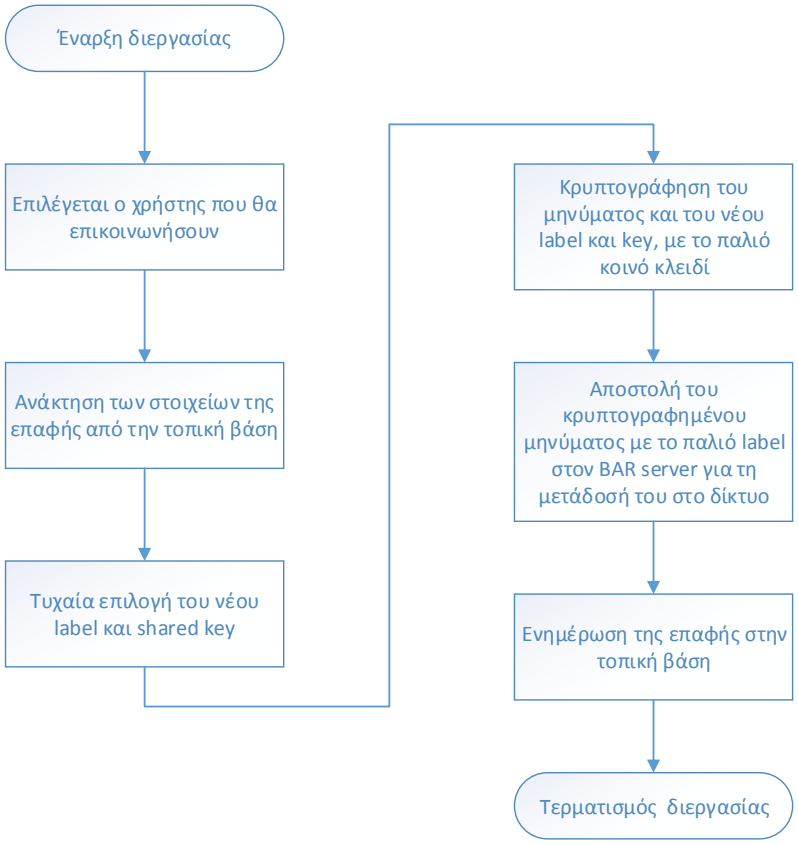
Εικόνα 5.5 Διάγραμμα: Κεντρική οθόνη της εφαρμογής

5.1.5 Πρωτόκολλο επικοινωνίας BAR (BCP) για χρήστες του ίδιου Cluster

Κατά την εγκατάσταση επικοινωνίας στο πρωτόκολλο BAR, όταν ο αποστολέας και ο παραλήπτης βρίσκονται στο ίδιο Cluster, συμμετέχουν τρεις τύποι χρηστών: ο αποστολέας του μηνύματος, ο παραλήπτης, καθώς και οι υπόλοιποι χρήστες του Cluster, οι οποίοι λαμβάνουν το κρυπτογραφημένο μήνυμα και το διαγράφουν.

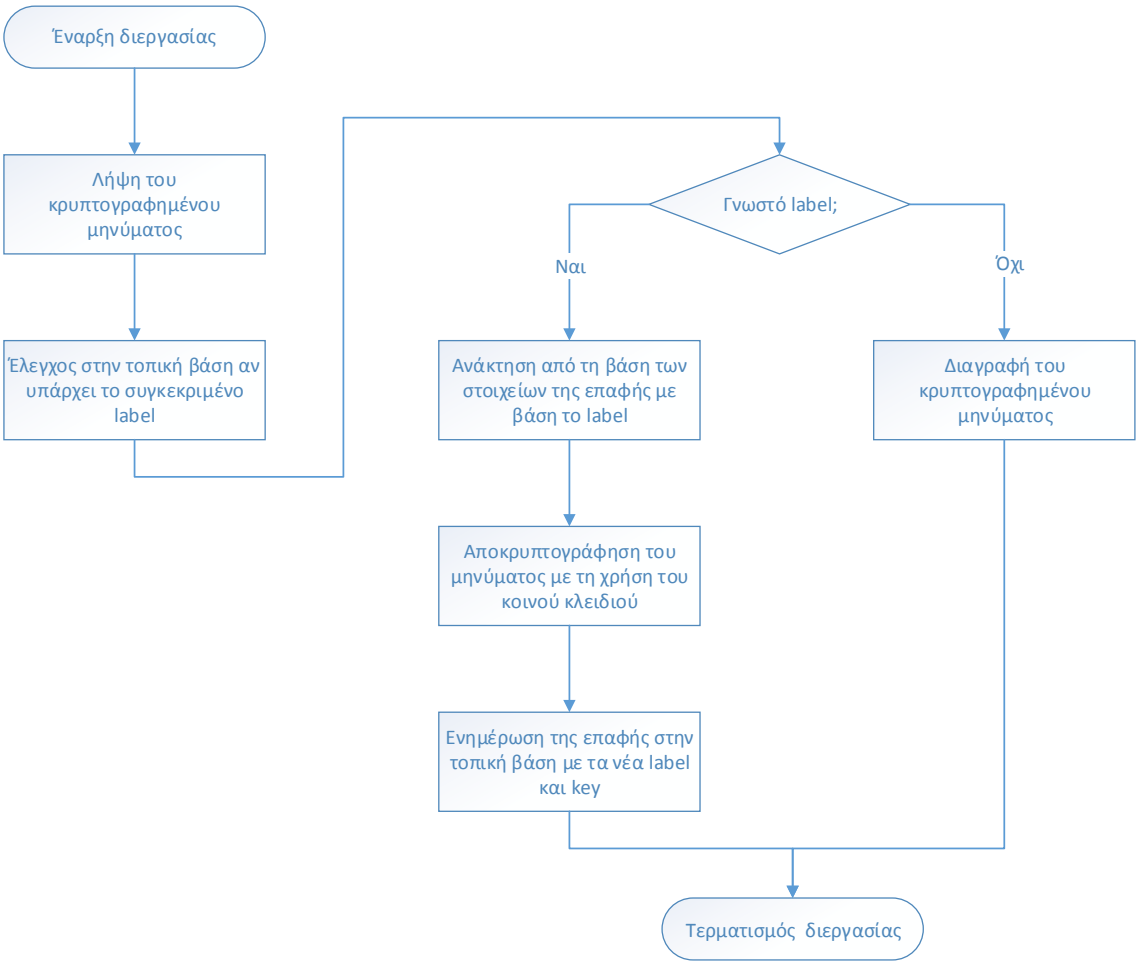
Στη συνέχεια, θα παρουσιαστούν τα διαγράμματα που παρουσιάζουν τη διαδικασία που κάνει ο κάθε τύπος χρήστη.

- *Ο αποστολέας του μηνύματος*



Εικόνα 5.6 Διάγραμμα: Πρωτόκολλο BCP για ίδιο Cluster - Αποστολέας

- *Οι υπόλοιποι χρήστες του Cluster και ο πραγματικός παραλήπτης*



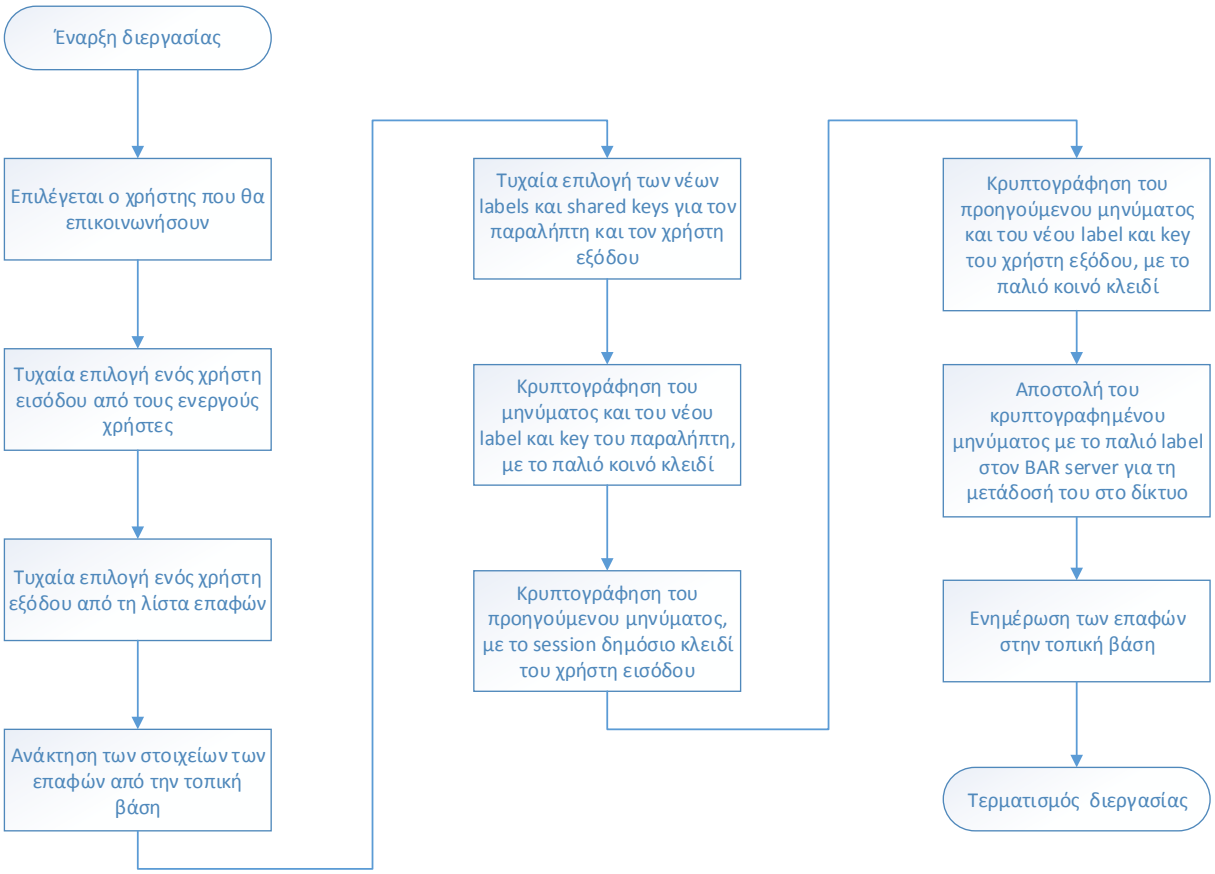
Εικόνα 5.7 Διάγραμμα: Πρωτόκολλο BCP για ίδιο Cluster - Παραλήπτες

5.1.6 Πρωτόκολλο επικοινωνίας BAR (BCP) για χρήστες διαφορετικού Cluster

Αντίστοιχα, κατά την εγκατάσταση ανώνυμης επικοινωνίας, όταν ο παραλήπτης και ο αποστολέας βρίσκονται σε διαφορετικό Cluster, συμμετέχουν έξι κατηγορίες χρηστών: ο αποστολέας του μηνύματος, οι χρήστες του Cluster του αποστολέα, ο χρήστης εξόδου που αναλαμβάνει την αποστολή του μηνύματος στο άλλο Cluster, ο χρήστης εισόδου που αναλαμβάνει την παραλαβή του μηνύματος, οι χρήστες του Cluster του παραλήπτη και ο ίδιος ο παραλήπτης.

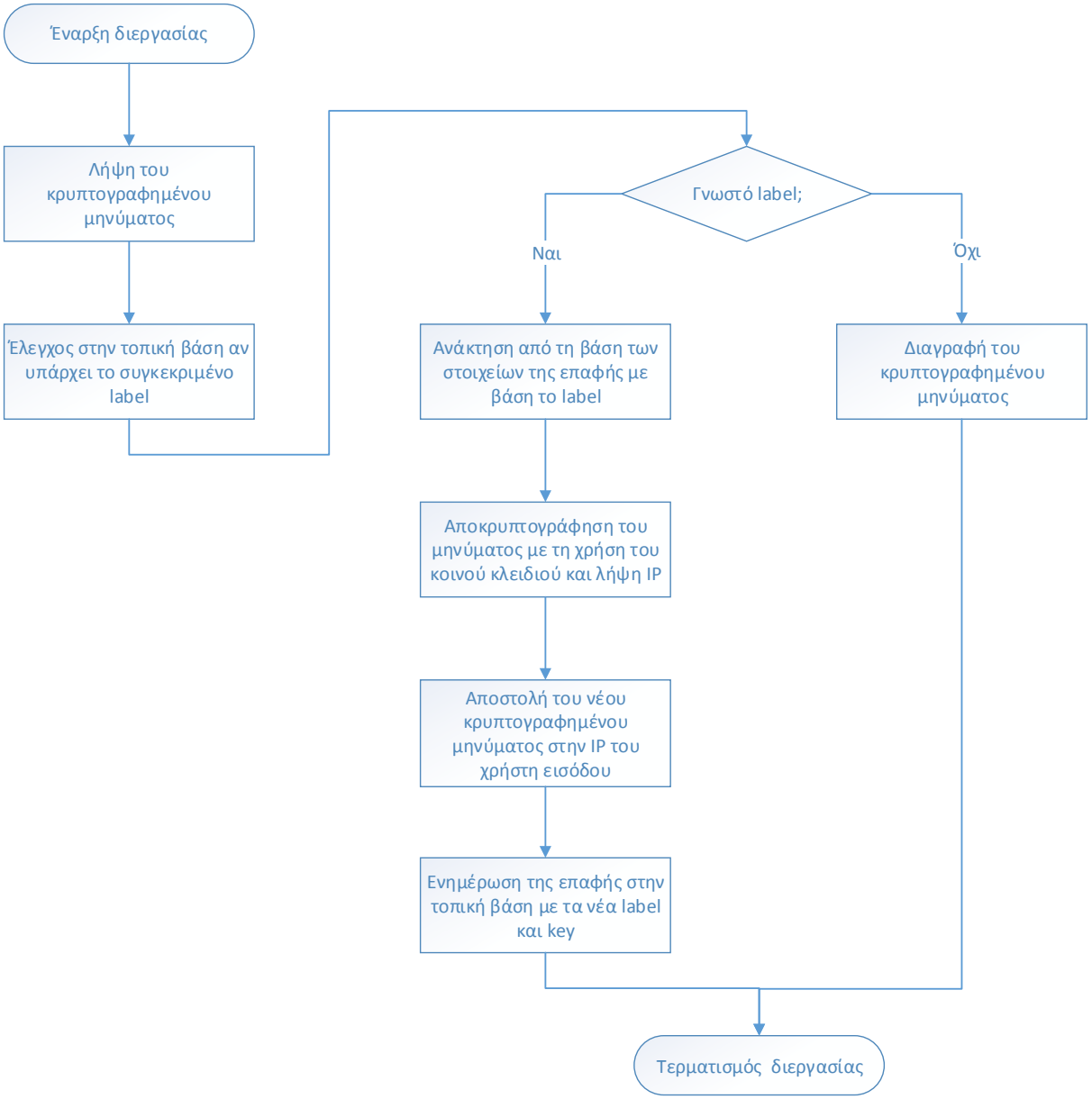
Στη συνέχεια, θα παρουσιαστούν τα διαγράμματα που παρουσιάζουν τη διαδικασία που κάνει ο κάθε τύπος χρήστη.

- *Ο αποστολέας του μηνύματος*



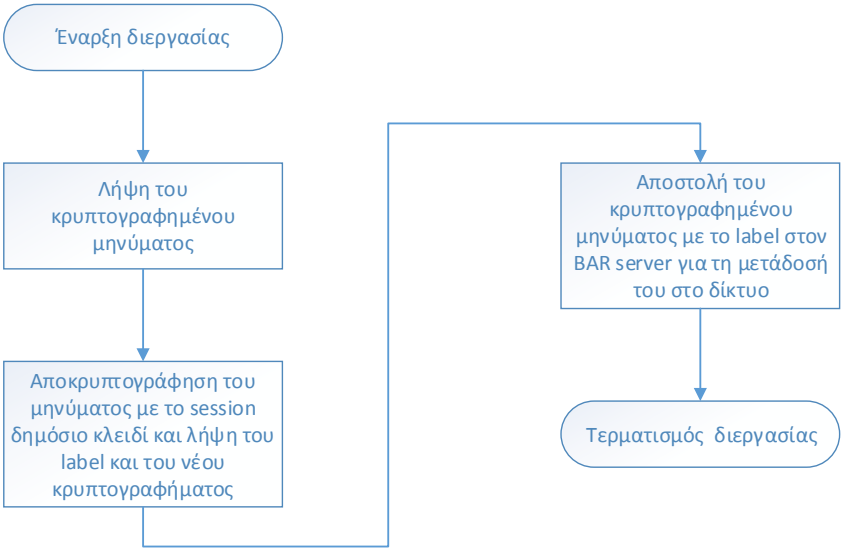
Εικόνα 5.8 Διάγραμμα: Πρωτόκολλο BCP για διαφορετικό Cluster - Αποστολέας

- *Οι υπόλοιποι χρήστες του Cluster του αποστολέα και ο χρήστης εξόδου*



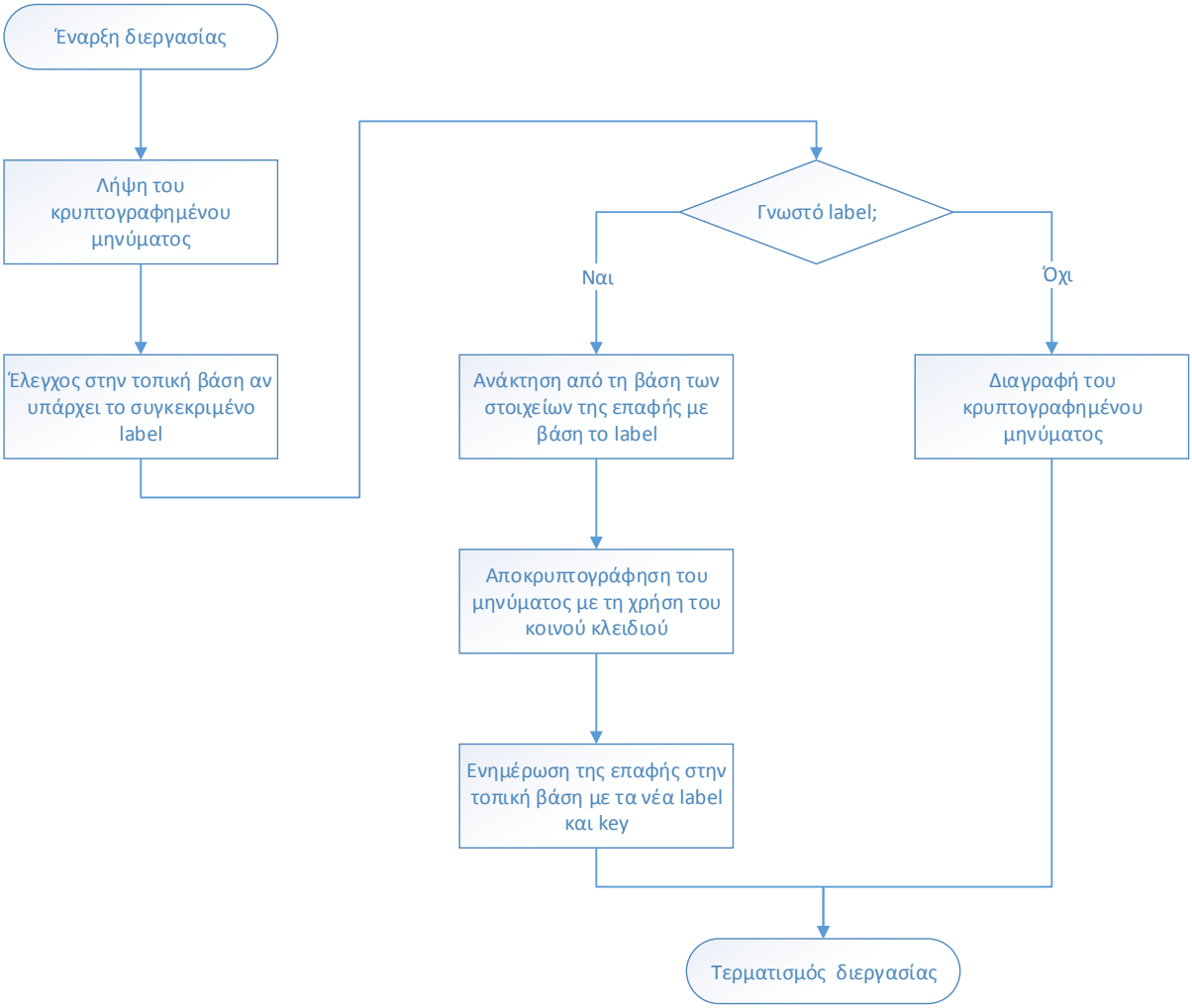
Εικόνα 5.9 Διάγραμμα: Πρωτόκολλο BCP για διαφορετικό Cluster - Παραλήπτες ίδιου Cluster

- *Ο χρήστης εισόδου*



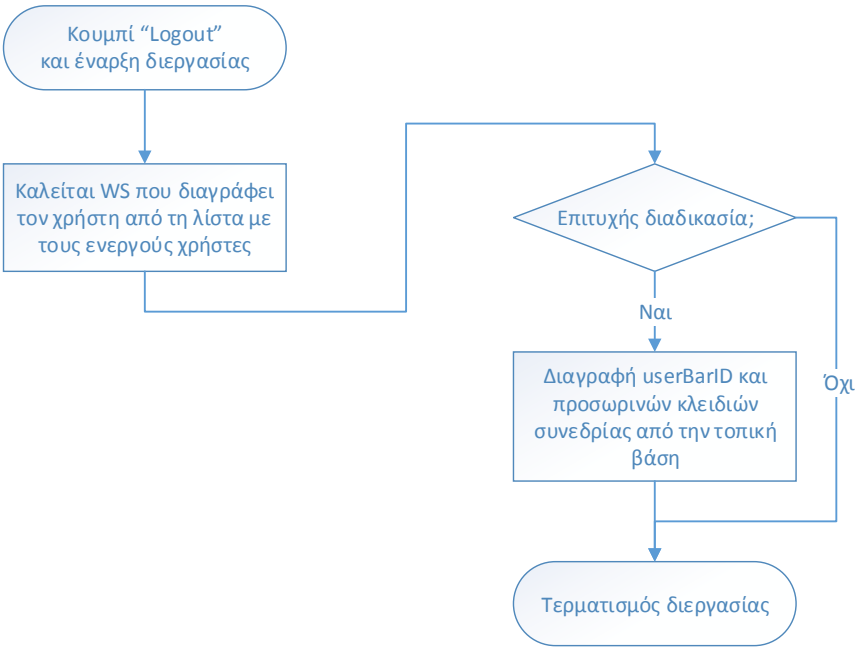
Εικόνα 5.10 Διάγραμμα: Πρωτόκολλο BCP για διαφορετικό Cluster – Χρήστης εισόδου

- *Οι υπόλοιποι χρήστες του Cluster του παραλήπτη και ο πραγματικός παραλήπτης*



Εικόνα 5.11 Διάγραμμα: Πρωτόκολλο BCP για διαφορετικό Cluster - Χρήστες Cluster παραλήπτη

5.1.7 Αποσύνδεση χρήστη (User logout)



Εικόνα 5.12 Διάγραμμα: Αποσύνδεση χρήστη

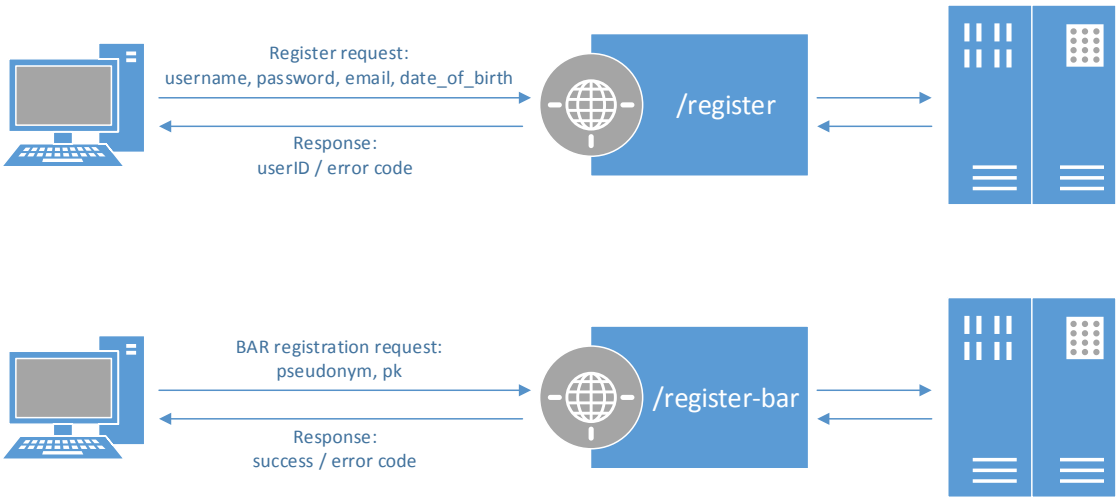
5.2 Διαγράμματα λειτουργίας Web Service

Στον Coordinator λειτουργούν κάποια Web Services, τα οποία επιτρέπουν την ανταλλαγή πληροφοριών και την επικοινωνία της εφαρμογής με την κεντρική βάση δεδομένων.

Οι βασικές λειτουργίες που καλύπτουν τα Web Services αναφέρονται παρακάτω:

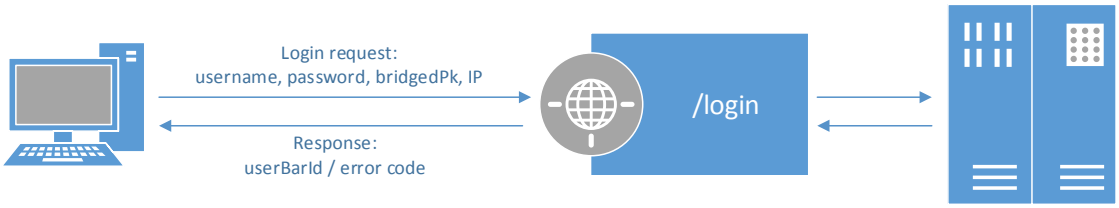
- Εγγραφή χρήστη (User Registration)
- Σύνδεση χρήστη (User Login)
- Ανταλλαγή Κλειδιών (Key Exchange)
- Αποσύνδεση χρήστη (Logout)

5.2.1 Εγγραφή χρήστη (User registration)



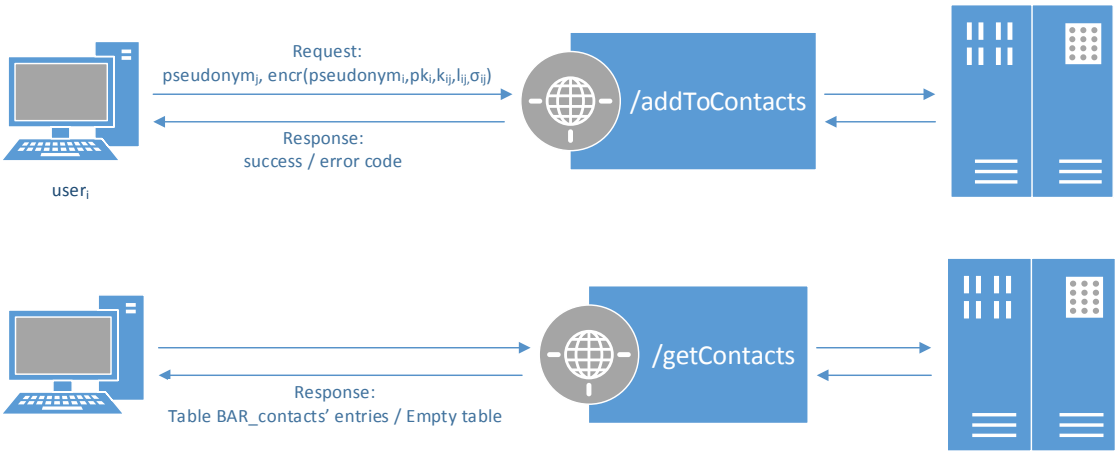
Εικόνα 5.13 Web Services - Εγγραφή χρήστη

5.2.2 Σύνδεση χρήστη (User login)



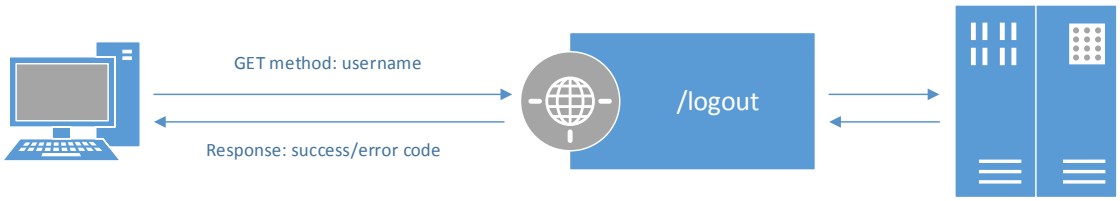
Εικόνα 5.14 Web Services - Σύνδεση χρήστη

5.2.3 Ανταλλαγή κλειδιών (Key exchange)



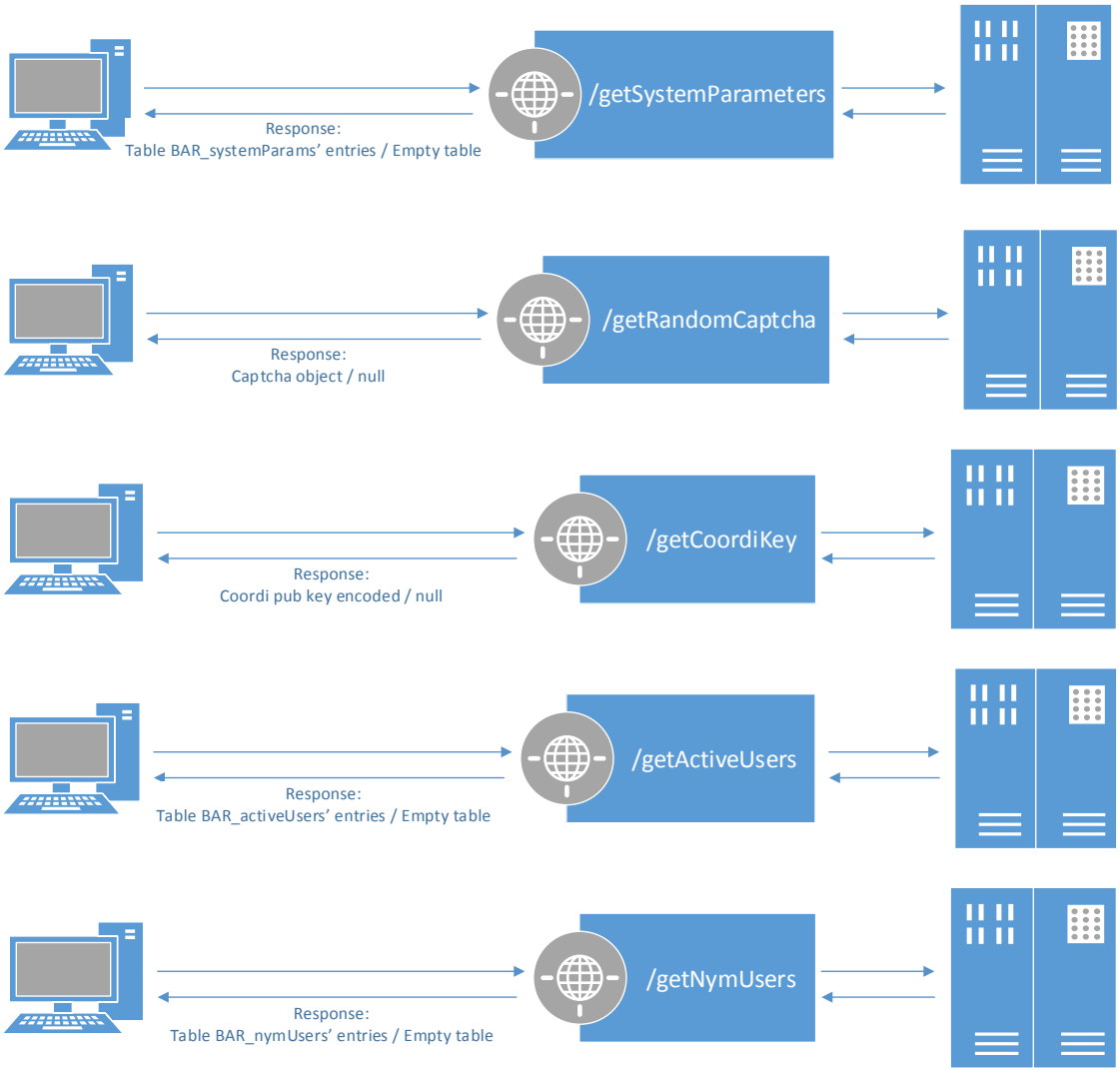
Εικόνα 5.15 Web Services - Ανταλλαγή κλειδιών

5.2.4 Αποσύνδεση χρήστη (User logout)



Εικόνα 5.16 Web Services - Αποσύνδεση χρήστη

5.2.5 Άλλα Web Services



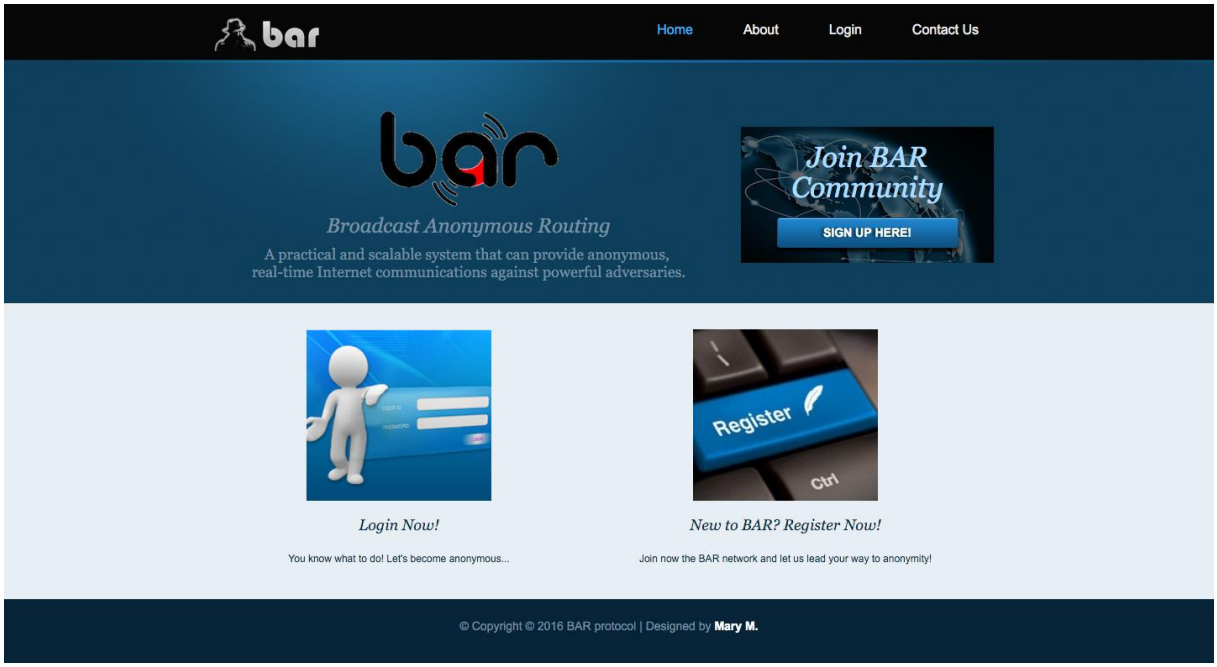
Εικόνα 5.17 Άλλα Web Services

5.3 Προγραμματισμός - Λειτουργία της εφαρμογής

Παρακάτω αναφέρεται αναλυτικά η λειτουργία της εφαρμογής BarApp [54] και παραπέμπονται σημαντικά αποσπάσματα κώδικα, τα οποία και αναλύονται.

5.3.1 Αρχική σελίδα εφαρμογής

Αποτελεί την πρώτη οθόνη που εμφανίζεται όταν ο χρήστης ενεργοποιεί την εφαρμογή. Στην αρχική σελίδα εμφανίζονται οι λειτουργίες της εφαρμογής, καθώς και οι σελίδες με τις σχετικές πληροφορίες του συστήματος BAR και επικοινωνίας με το διαχειριστή.

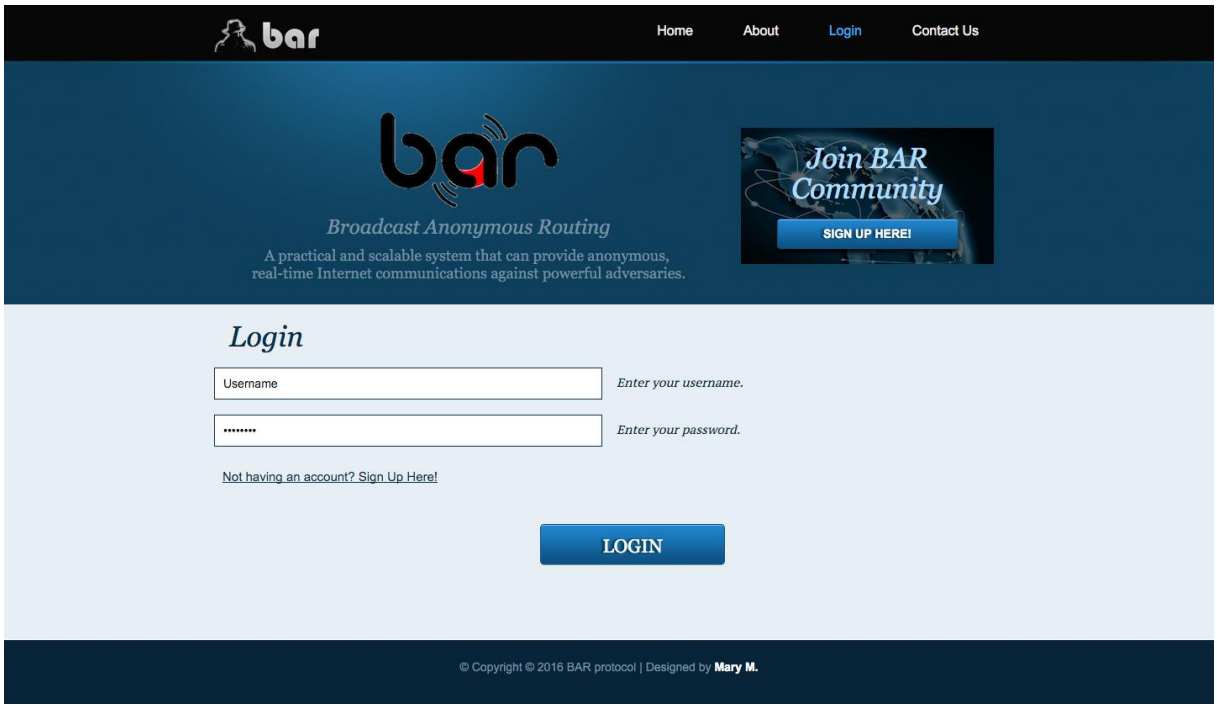


Εικόνα 5.18 Αρχική σελίδα εφαρμογής

5.3.2 Σύνδεση - Αποσύνδεση χρήστη

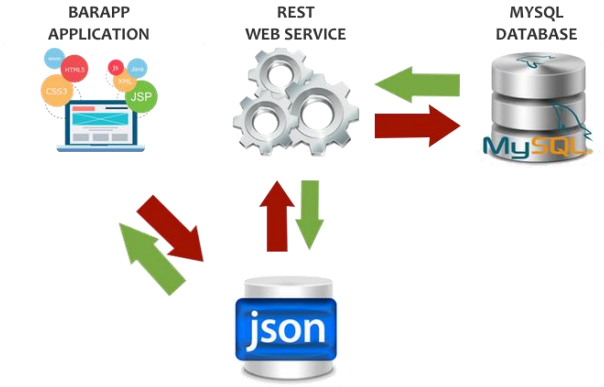
Επιλέγοντας από την αρχική σελίδα ή από το μενού τον σύνδεσμο “Login”, ο χρήστης ανακατευθύνεται στη σελίδα σύνδεσης. Στη συγκεκριμένη οθόνη, συμπληρώνονται τα απαραίτητα αναγνωριστικά πρόσβασης. Στη συνέχεια, πατώντας το κουμπί Login ελέγχεται η εγκυρότητα των στοιχείων που έχουν δοθεί και αν η εισαγωγή είναι σωστή, ενεργοποιείται το BAR πρωτόκολλο σύνδεσης για την ανώνυμη περιήγηση του χρήστη και καλείται η σελίδα με τα διαθέσιμα BAR services. Σε περίπτωση λάθους, εμφανίζεται ανάλογο μήνυμα που ενημερώνει αν κάποιο από τα αναγνωριστικά πρόσβασης είναι λανθασμένο, αν ο λογαριασμός είναι κλειδωμένος λόγω πολλών λανθασμένων προσπαθειών πρόσβασης ή αν απέτυχε το BAR πρωτόκολλο σύνδεσης.

Στην καρτέλα login παρέχεται επίσης η δυνατότητα ανακατεύθυνσης στη σελίδα εγγραφής χρήστη.



Εικόνα 5.19 Είσοδος χρήστη

Πατώντας το κουμπί Login πραγματοποιείται η διαδικασία που θα περιγραφεί στη συνέχεια. Τα στοιχεία που ορίστηκαν ως όνομα χρήστη και κωδικός, μαζί με τα δεδομένα που απαιτούνται από το BAR πρωτόκολλο σύνδεσης (το δημόσιο κλειδί συνεδρίας του χρήστη, η IP διεύθυνση και ο αριθμός του BAR server και Cluster που έχει βρεθεί ότι ανήκει ο χρήστης) στέλνονται στον διακομιστή με τον οποίο είναι συνδεδεμένη η εφαρμογή. Επειδή, δεν είναι ασφαλές να πραγματοποιηθεί επικοινωνία μεταξύ της εφαρμογής και της βάσης δεδομένων, χρησιμοποιείται ενδιάμεσος κώδικας που μετατρέπει τα δεδομένα σε κατανοητή μορφή και για τις δύο πλευρές.



Εικόνα 5.20 Επικοινωνία BarApp - MySQL Database

Τα δεδομένα στέλνονται από την εφαρμογή με τη μορφή json, ένα Restful Web Service τα επεξεργάζεται και συνδέεται με τη βάση δεδομένων για να κάνει τον έλεγχο. Αφού γίνει ο έλεγχος, τα στοιχεία αποθηκεύονται σε μεταβλητές με κωδικοποίηση json τα οποία μπορεί να επεξεργαστεί η εφαρμογή.

Συγκεκριμένα, τα δεδομένα εισόδου στέλνονται με HTTP_POST στον διακομιστή. Ο διακομιστής λαμβάνει τα δεδομένα και ενεργοποιείται μία συνάρτηση, η οποία ελέγχει στη βάση δεδομένων αν υπάρχει ο συγκεκριμένος χρήστης και αν βρεθεί, χρησιμοποιείται η τιμή του πεδίου encryptSalt για να δημιουργηθεί

από τη συνάρτηση σύνοψης SHA-256 το ψηφιακό αποτύπωμα του κωδικού. Σε περίπτωση που δεν υπάρχει το δοθέν όνομα χρήστη, επιστρέφεται η αντίστοιχη τιμή στη μεταβλητή result και η συνάρτηση τερματίζεται. Αν υπάρχει ο συγκεκριμένος χρήστης, ελέγχονται τα υπόλοιπα στοιχεία που έχουν δοθεί και καλείται η συνάρτηση της σύνδεσης για το πρωτόκολλο BAR. Αν δεν έχει προκύψει κάποιο πρόβλημα, η μεταβλητή result επιστρέφει μία τυχαία τιμή, η οποία αναφέρεται στο αναγνωριστικό που θα έχει ο χρήστης στο πρωτόκολλο BAR, στην εφαρμογή επιτρέποντας στον χρήστη να εισέλθει στη σελίδα με τα διαθέσιμα BAR services. Οι τιμές που μπορεί να πάρει η μεταβλητή result αναφέρονται παρακάτω:

- Αν είναι όλα σωστά, η μεταβλητή result επιστρέφει το αναγνωριστικό που θα έχει ο χρήστης για την τρέχουσα σύνοδο στο πρωτόκολλο BAR.
- Αν δεν υπάρχει ο συγκεκριμένος χρήστης ή ο κωδικός που δόθηκε είναι λάθος, επιστρέφεται η τιμή “-105”.
- Αν ο λογαριασμός είναι κλειδωμένος, η μεταβλητή result επιστρέφει την τιμή “-101”.
- Αν υπήρχε κάποιο γενικό πρόβλημα επικοινωνίας, επιστρέφεται η τιμή “-102”.

Η εφαρμογή BarApp λαμβάνει την επιστρεφόμενη τιμή και εμφανίζει το αντίστοιχο μήνυμα στον χρήστη. Σε περίπτωση που έχουν δοθεί σωστά τα στοιχεία εισόδου και έχει ολοκληρωθεί με επιτυχία η διαδικασία της σύνδεσης στο σύστημα BAR, ο χρήστης ανακατευθύνεται στη σελίδα με τα διαθέσιμα BAR services.

Αποσύνδεση γίνεται από την επιλογή Logout που υπάρχει στο μενού της εφαρμογής. Μόλις πατηθεί το Logout, καλείται το Web Service που θα διαγράψει την εγγραφή από τον κατάλογο με τους ενεργούς χρήστες του πρωτοκόλλου. Στη συνέχεια, η εφαρμογή διαγράφει τα δεδομένα του χρήστη, που απαιτούνται για την ομαλή λειτουργία του BAR πρωτοκόλλου, από την τοπική βάση δεδομένων του και τον επιστρέφει στην αρχική σελίδα.

Σημαντικά σημεία κώδικα:

Από τα πιο σημαντικά σημεία της διαδικασίας του login είναι η υλοποίηση του υπο-πρωτοκόλλου του συστήματος BAR για τη σύνδεση του χρήστη. Έτσι, παρακάτω θα παρουσιαστούν αποσπάσματα κώδικα από τη συνάρτηση εισόδου login, όπως παρουσιάζεται στην εφαρμογή BarApp.

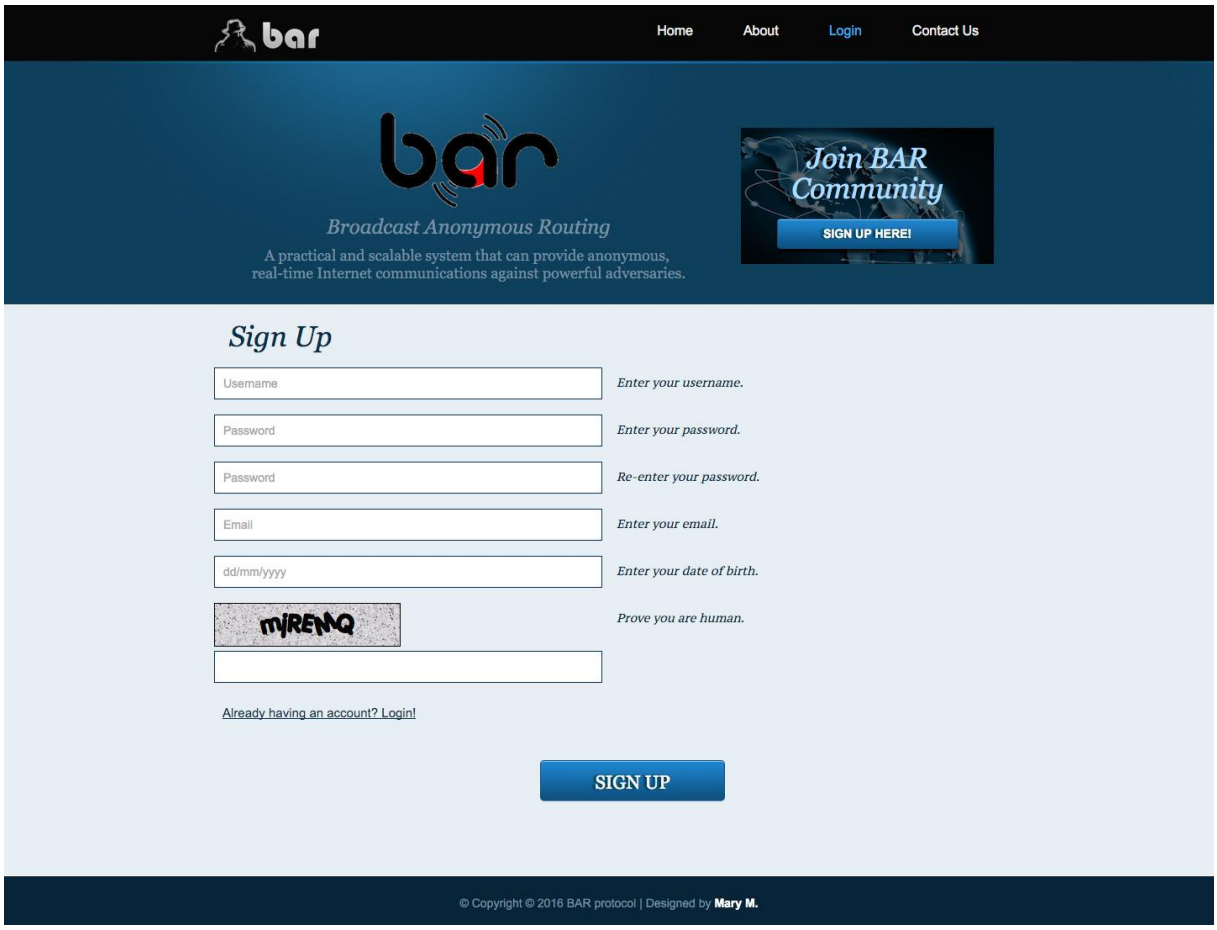
Η συνάρτηση login, αρχικά λαμβάνει τα στοιχεία εισόδου που έχει δώσει ο χρήστης και ελέγχει την εγκυρότητά τους. Στη συνέχεια, ανακτά από την τοπική βάση δεδομένων το ψευδώνυμο του χρήστη για το πρωτόκολλο BAR και το δημόσιο κλειδί του, και υπολογίζει τη συνάρτηση hash. Έπειτα, και χρησιμοποιώντας τον κατάλογο με τις δημόσιες παραμέτρους του συστήματος, που έχει λάβει από τον Coordinator, υπολογίζει τον BAR server και το Cluster που ανήκει. Στη συνέχεια, δημιουργεί ένα ζεύγος δημόσιου-ιδιωτικού κλειδιού, που θα χρησιμοποιεί στην τρέχουσα συνεδρία εισόδου. Τέλος, στέλνει όλα τα δεδομένα ως κλάση *LoginData* στο αντίστοιχο Web Service του διακομιστή.

```
[...]
    LoginData ld = new LoginData();

    ld.setUsername(request.getParameter("username").toString());
    ld.setPassword(request.getParameter("password").toString());

    /* ----- (a) Get nym, pk ----- */
    String nym = UserControl.getNym();
    PublicKey pubKey = UserControl.getPK(false);

    [...]
        String pkString = DatatypeConverter.printBase64Binary(pubKey.getEn-
coded());

        // Compute hash(nym|pk)
        SHAencrypt.SHA256encrypt(nym + "-" + pkString);

        /* ----- (b) WS -- Get System Parameters ----- */
        systemParams = WS_SystemParams.getSystemParams();
        [...]

        /* ----- (c) Create a pair of session keys ----- */
        RSAkeys.createKeys("");

        PublicKey sessionPubKey = UserControl.getPK(true);
        ld.setBridgedPk(DatatypeConverter.printBase64Binary(sessionPub-
Key.getEncoded()));

        // Get IP address
        ld.setIp(UserControl.getIP());

        /* ----- (d) WS -- Send Login Data ----- */
        ld.setServerNo(serverNo);
        ld.setClusterNo(clusterNo);

        String result = WS_Users.login(ld);
```

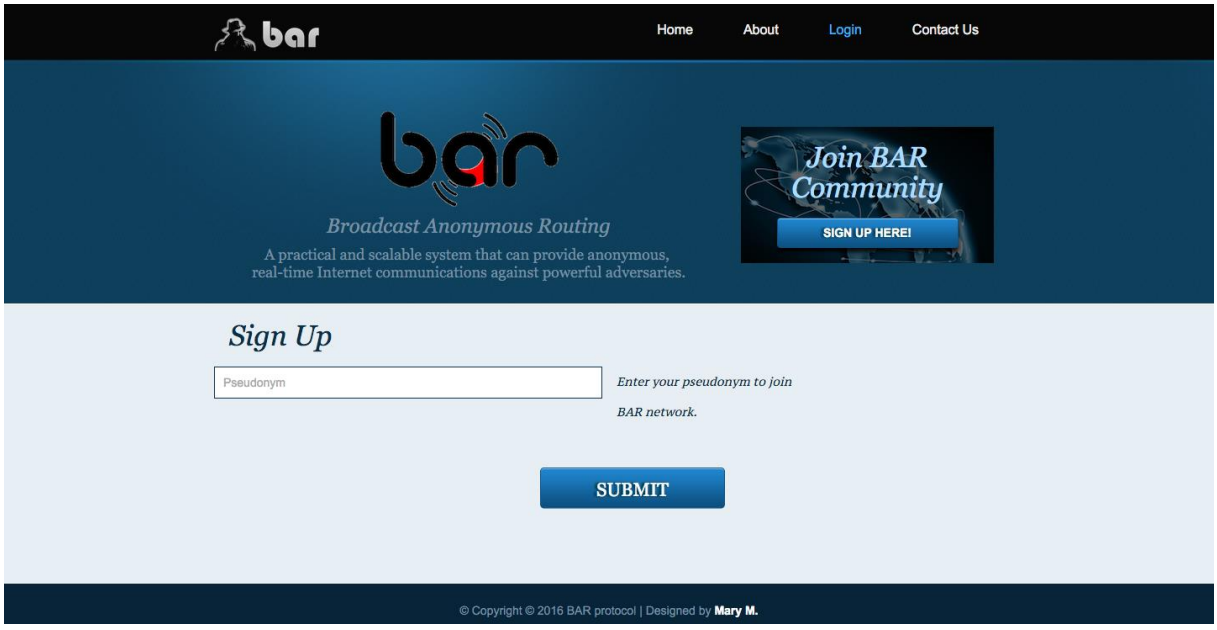
5.3.3 Εγγραφή χρήστη

Επιλέγοντας από την αρχική σελίδα τον σύνδεσμο του “Register” ή μέσα από τη σελίδα του login, ο χρήστης ανακατευθύνεται στη σελίδα εγγραφής. Στη συγκεκριμένη οθόνη, συμπληρώνονται τα επιθυμητά αναγνωριστικά που θα χρησιμοποιεί ο χρήστης για την είσοδό του στην εφαρμογή, καθώς και κάποια άλλα δεδομένα. Στη συνέχεια, πατώντας το κουμπί *Sign Up*, ελέγχεται η εγκυρότητα των στοιχείων που έχουν δοθεί και αν είναι σωστά ενεργοποιείται μία νέα οθόνη, η οποία του ζητάει να δηλώσει το ψευδώνυμο που θα χρησιμοποιεί για την επικοινωνία του στο πρωτόκολλο BAR. Έπειτα, και εάν όλες οι προηγούμενες διαδικασίες είναι επιτυχημένες, ενεργοποιείται το υπο-πρωτόκολλο BAR για την εγγραφή του χρήστη.

Στη σελίδα εγγραφής δίνεται η δυνατότητα για ανακατεύθυνση του χρήστη στην οθόνη σύνδεσης.



Εικόνα 5.21 Εγγραφή χρήστη



Εικόνα 5.22 Εγγραφή χρήστη (πρωτόκολλο BAR)

Πατώντας τα κουμπί Sign Up, γίνεται η παρακάτω διαδικασία. Τα στοιχεία που ορίστηκαν για την εγγραφή του χρήστη (όνομα χρήστη, κωδικός πρόσβασης και επιβεβαίωσή του, email και ημερομηνία γέννησης, μαζί με τον κωδικό captcha, αποστέλλονται στον διακομιστή με τον οποίο είναι συνδεδεμένη η εφαρμογή.

Τα δεδομένα στέλνονται από την εφαρμογή με μορφή json, ένα Restful Web Service τα επεξεργάζεται και συνδέεται με τη βάση δεδομένων για να κάνει τον έλεγχο.

Συγκεκριμένα, τα δεδομένα εισόδου στέλνονται με HTTP_POST στον διακομιστή. Εκείνος, στη συνέχεια ενεργοποιεί τη συνάρτηση */register*, η οποία αρχικά ελέγχει την ταύτιση των δύο κωδικών και την εγκυρότητα του email και την ημερομηνίας γέννησης. Έπειτα, ελέγχει αν τα κάποιο από τα στοιχεία αντιστοιχεί σε κάποιον ήδη υπάρχων χρήστη και επιστρέφεται η αντίστοιχη τιμή λάθους στη μεταβλητή result. Αν όλες οι προηγούμενες διαδικασίες γίνουν με επιτυχία, δημιουργείται η επιπλέον πληροφορία που θα χρησιμοποιηθεί για την κρυπτογράφηση του κωδικού πρόσβασης μέσω της συνάρτησης σύνοψης SHA-256. Τέλος, εισάγεται ο χρήστης στη βάση δεδομένων και επιστρέφεται η μεταβλητή result στην εφαρμογή. Οι τιμές που μπορεί να πάρει η μεταβλητή result αναφέρονται παρακάτω:

- Αν είναι όλα σωστά, η μεταβλητή result επιστρέφει το αναγνωριστικό userID που έχει ο νέος χρήστης
- Αν δεν ταιριάζουν οι δύο κωδικοί πρόσβασης που δόθηκαν, επιστρέφεται η τιμή “-201”.
- Αν το email που δηλώθηκε δεν είναι σωστό, η μεταβλητή result επιστρέφει την τιμή “-202”.
- Αν υπάρχει ήδη στη βάση δεδομένων κάποια εγγραφή με το ίδιο όνομα χρήστη, επιστρέφεται η τιμή “-203”.
- Αν υπάρχει ήδη στη βάση δεδομένων κάποια εγγραφή με το ίδιο email, επιστρέφεται η τιμή “-204”.
- Αν η ημερομηνία γέννησης δεν είναι σωστή, η μεταβλητή result επιστρέφει την τιμή “-205”.
- Αν η ημερομηνία γέννησης που δηλώθηκε δεν είναι ανάμεσα στα επιτρεπτά όρια ηλικίας, επιστρέφεται η τιμή “-206”.
- Αν ο κωδικός captcha δεν είναι σωστός, η μεταβλητή result επιστρέφει “-207”.
- Αν υπήρχε κάποιο γενικό πρόβλημα επικοινωνίας, επιστρέφεται η τιμή “-208”.

Η εφαρμογή BarApp λαμβάνει την επιστρεφόμενη τιμή και εμφανίζει το αντίστοιχο μήνυμα στον χρήστη. Σε περίπτωση που έχει ολοκληρωθεί με επιτυχία η διαδικασία της εγγραφής, ο χρήστης ανακατευθύνεται στη σελίδα εγγραφής του πρωτοκόλλου BAR, στην οποία πρέπει να δηλώσει το ψευδώνυμο που θα χρησιμοποιεί για την ανώνυμη επικοινωνία του.

Πατώντας το κουμπί “Submit” στη σελίδα *register-bar*, ενεργοποιείται το υπο-πρωτόκολλο εγγραφής του πρωτοκόλλου BAR. Αρχικά, δημιουργούνται το ζεύγος δημόσιου και ιδιωτικού κλειδιού για τον συγκεκριμένο χρήστη και αποθηκεύεται στην τοπική βάση δεδομένων. Στη συνέχεια, επιλέγονται τρεις ενεργοί χρήστες από τον αντίστοιχο δημόσιο κατάλογο του Coordinator και κρυπτογραφούνται τα στοιχεία που θα αποσταλούν με το δημόσιο κλειδί συνόδου του καθενός. Τέλος, αποστέλλονται τα στοιχεία εγγραφής για το πρωτόκολλο BAR στον Coordinator, μέσα από μία διαδρομή onion path routing.

Σημαντικά σημεία κώδικα:

Το παρακάτω απόσπασμα κώδικα προέρχεται από τη συνάρτηση εγγραφής στο πρωτόκολλο BAR, που αποτελεί μέρος της εφαρμογής BarApp. Αποτελεί ουσιαστικά τη διαδικασία που πραγματοποιείται προκειμένου να δημιουργηθούν τα απαραίτητα δεδομένα που χρειάζεται το πρωτόκολλο για τη λειτουργία του και να αποσταλούν στον Coordinator.

```
[...]
BarSignupData bsd = new BarSignupData();

/* ----- (a) WS -- Get Coordi Key ----- */
encCoordiKey = WS_Users.getCoordiKey();

[...]
    bsd.setUserID(session.getAttribute("userID").toString());
    bsd.setPseudonym(request.getParameter("pseudonym").toString());

    /* ----- (b) Create a pair of keys ----- */
    RSAkeys.createKeys(request.getParameter("pseudonym"));

    PublicKey pubKey = UserControl.getPK(false);
    bsd.setPk(DatatypeConverter.printBase64Binary(pubKey.getEncoded()));

    /* ----- (c) WS -- Get Active Users ----- */
    ArrayList<BARactiveUsers> activeUsers = WS_ActiveUsers.getActiveUsers();

    /* ----- (d) Check if activeUsers>3 to set onion path routing ----- */
    if (activeUsers.size() < onionNodes) {
       [...]
    }
    else {
        /* ----- (e) Choose randomly 3 active users ----- */
        ArrayList<BARactiveUsers> rndAU =
    WS_ActiveUsers.getRndActiveUsers(activeUsers);

        /* ----- (f) Encrypt data with Coordi PK ----- */
        String encData = DataControl.encryptBarDataToCoordi(encCoordiKey,
    bsd);
        if (encData.equals("")) { %>
             [...]
        } else {
            /* --- (g) Encrypt data with each of the 3 random users' PKs -- */
            encData = DataControl.encryptDataToAU(true, rndAU, encData);

            if (encData.equals("")) { %>
                [...]
            } else {
                /* --- (h) Onion path routing to send enc{BarSignupData} -- */
                boolean result = DataControl.sendData(encData,
    rndAU.get(rndAU.size()-1).getIp());
```

5.3.4 Ανταλλαγή κλειδιών

Αμέσως μετά τη διαδικασία της εγγραφής, ενεργοποιείται το πρωτόκολλο ανταλλαγής κλειδιών. Προκειμένου να ενεργοποιηθεί η ανώνυμη επικοινωνία του συστήματος BAR, όλοι οι χρήστες θα πρέπει να έχουν ανταλλάξει ένα κοινό μυστικό κλειδί και ένα label.

Έτσι, αμέσως μετά τη διαδικασία της εγγραφής, ο νέος χρήστης του συστήματος επιλέγει κάθε έναν από τους χρήστες του καταλόγου *UsersList* και επιλέγει ένα τυχαίο κλειδί και ένα label. Το υπογράφει, το κρυπτογραφεί και το αποστέλλει μέσω διαδρομής onion path routing στον Coordinator, ο οποίος είναι υπεύθυνος να το καταχωρήσει στη δημόσια λίστα των επαφών. Τέλος, ο χρήστης καταχωρεί την κάθε επαφή που έχει στείλει, στην τοπική βάση δεδομένων του.

Αντίστοιχα, ένας ήδη εγγεγραμμένος χρήστης που εισέρχεται στην υπηρεσία, ζητάει από τον Coordinator τη δημόσια λίστα με τις επαφές, επιλέγει ποιες απευθύνονται σε εκείνον, τις αποκρυπτογραφεί και τις καταχωρεί στην τοπική βάση δεδομένων του.

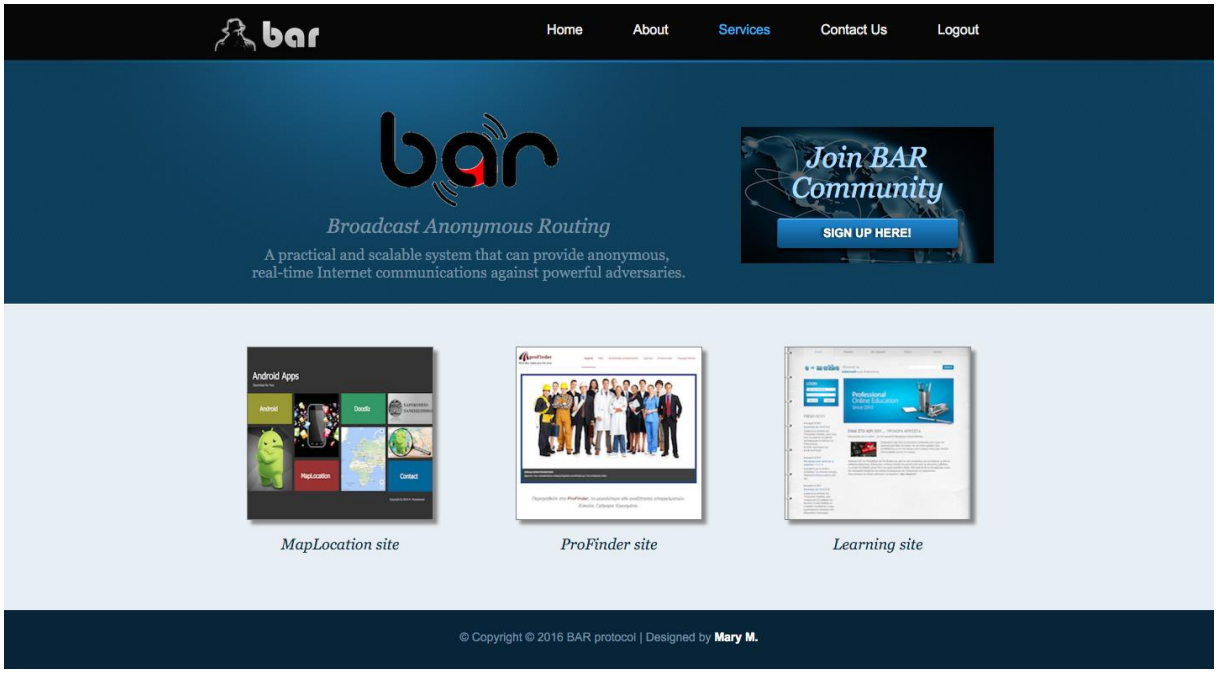
Σημαντικά σημεία κώδικα:

Παρακάτω, παρουσιάζεται το απόσπασμα του κώδικα που σχετίζεται με τη δημιουργία των κλειδιών και labels και την αποστολή τους από έναν χρήστη που μόλις εγγράφηκε στην υπηρεσία.

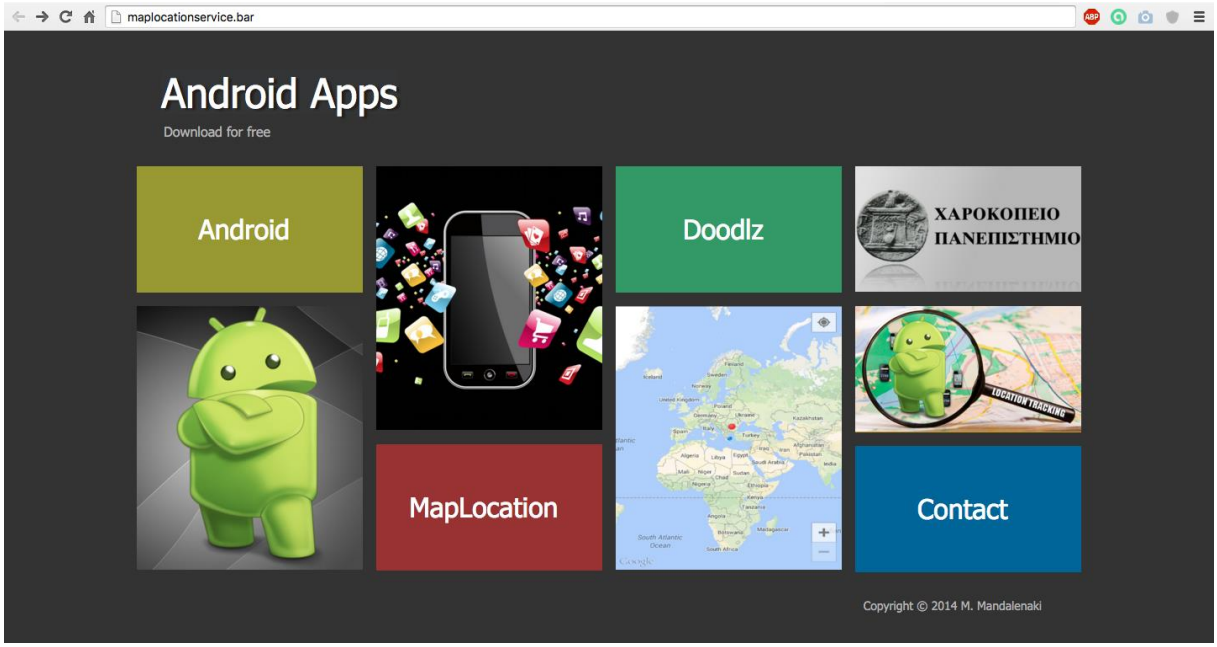
```
public static boolean keyExchangeProtocol_send(String pseudonym) throws Exception {
    /* ----- (a) WS -- Get all BAR users (nymUsers) ----- */
    ArrayList<BARnymUsers> nymUsers = WS_NymUsers.getNymUsers();
    [...]

    /* ----- (b) For each entry: ----- */
    for (BARnymUsers u:nymUsers) {
        /* ----- (b1) Choose a random label ----- */
        Random r = new Random();
        labelBytes = new byte[16];
        r.nextBytes(labelBytes);

        label = getHexString(labelBytes);

        /* ----- (b2) Choose a random shared key ----- */
        [...]
        sharedKey = getHexString(keyBytes);

        /* ----- (b3) Sign data ----- */
        PublicKey pubKey = RSAencrypt.decryptOwnPk();
        ContactsPairingData cpd = new ContactsPairingData(u.pseudonym, pubKey.toString(), u.pk,
    sharedKey, label);
        byte[] sig = signData(cpd);

        /* ----- (b4) Get userj pk ----- */
        PublicKey pkj = RSAencrypt.decryptPk(u.pk);

        /* ----- (b5) Generate encodedData ----- */
        String encData = pseudonym + "-" + pubKey.toString() + "-" + sharedKey + "-" + label + "-" +
    new String(sig, StandardCharsets.UTF_8);
        byte[] encDataBytes = RSAencrypt.RSAenc(encData, pkj);
        encData = new String(encDataBytes, StandardCharsets.UTF_8);

        /* ----- (c) Create onion routing path to Coordinator ----- */

        /* ----- (c1) WS -- Get Active Users ----- */
        ArrayList<BARactiveUsers> activeUsers = WS_ActiveUsers.getActiveUsers();

        /* ----- (c2) Check if activeUsers>3 to set onion path routing ----- */
        if (activeUsers.size() < onionNodes) { [...] }

        /* ----- (c3) Choose randomly 3 active users ----- */
        ArrayList<BARactiveUsers> rndAU = WS_ActiveUsers.getRndActiveUsers(activeUsers);

        /* ----- (c4) Encrypt data with Coordi PK ----- */
        String encCoordiKey = WS_Users.getCoordiKey();
        encData = DataControl.encryptDataToCoordi(encCoordiKey, encData);
        if (encData.equals("")) { [...] }

        /* ----- (c5) Encrypt data with each of the 3 random users' PKs ----- */
        encData = DataControl.encryptDataToAU(false, rndAU, encData);

        if (encData.equals("")) { [...] }

        /* ----- (c6) Onion path routing to send enc{KeyExchangeData} ----- */
        success = DataControl.sendData(encData, rndAU.get(rndAU.size()-1).getIp());

        if (success) {
            /* ----- (c7) If success, add contact to db ----- */
            DBcontacts.dbContactInsert(u.getPseudonym(), u.getPk(), sharedKey, label, null);
        }
    }
}
```

5.3.5 Σελίδα με τα διαθέσιμα BAR services

Μετά την επιτυχή εισαγωγή των στοιχείων εισόδου, ο χρήστης εισέρχεται στην κεντρική οθόνη του συστήματος BAR. Στη συγκεκριμένη καρτέλα παρουσιάζεται μία λίστα με τα διαθέσιμα BAR Hidden Services, με τα οποία έχει γίνει η διαδικασία ανταλλαγής του κοινού κλειδιού και label. Τα BAR Hidden Services είναι ουσιαστικά τρεις ιστοσελίδες, οι διευθύνσεις των οποίων δεν είναι γνωστές. Έτσι, ο χρήστης επιλέγει ένα service που θα ήθελε να επικοινωνήσει, το οποίο ανοίγει σε διαφορετική καρτέλα.



Εικόνα 5.23 Κεντρική σελίδα με διαθέσιμα BAR services



Εικόνα 5.24 Ανώνυμη επικοινωνία με ένα BAR service

Σημαντικά σημεία κώδικα:

Για την αρχικοποίηση του πρωτοκόλλου BAR και των services, χρησιμοποιήθηκαν οι εντολές της γλώσσας python που υποστήριζε η προηγούμενη υλοποίηση του συστήματος.

Έτσι, αρχικά προστίθεται το κάθε BAR service στη λίστα επαφών του χρήστη και στη συνέχεια, γίνεται η είσοδο του χρήστη στον BAR server.

Δυστυχώς, στην παρούσα υλοποίηση δεν έχει γίνει η ανταλλαγή των κλειδιών ενός χρήστη με ένα BAR service, καθώς τα services είναι οντότητες που παραμένουν μόνιμα συνδεδεμένες, με αποτέλεσμα να μην εκκινείται το υπο-πρωτόκολλο της ανταλλαγής κλειδιών για την εύρεση νέων επαφών. Η επίλυση του συγκεκριμένου προβλήματος αποτελεί μελλοντική επέκταση.

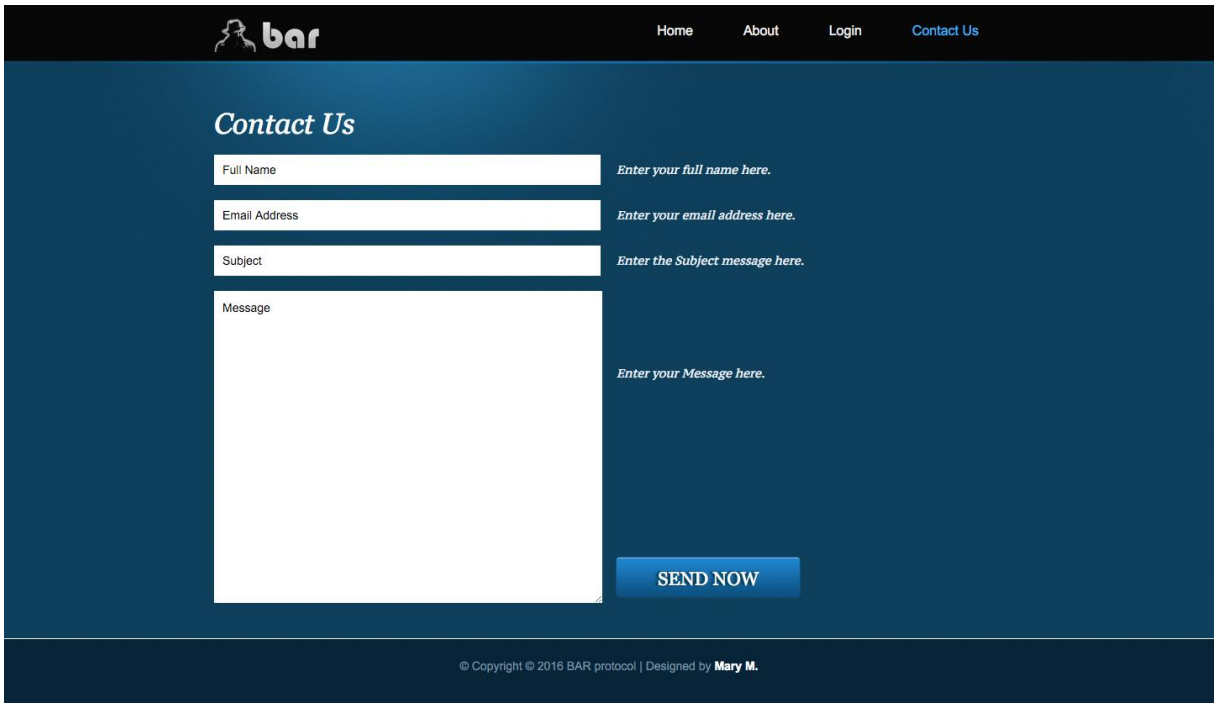
```
/** Initialize BAR protocol (python commands) */
String[] cmd1 = {"/bin/sh", "-c", "cd ~ && "
        + "git clone https://github.com/sophron/BAR.git"};
Process process = Runtime.getRuntime().exec(cmd1);
String[] cmd2 = {"/bin/sh", "-c", "cd ~/BAR && "
        + "sudo python setup.py install"};
process = Runtime.getRuntime().exec(cmd2);

if (serv.equals("maplocation")) {
    /** -------------------- Add the hidden service -------------------- */
    String[] cmd3 = {"/bin/sh", "-c", "cd ~/BAR && "
            + " python bin/bar contacts add --name a-service-name --label " +
l1 + " --sharedkey " + k1};
    process = Runtime.getRuntime().exec(cmd3);

    /** -------------------- Login to BAR server -------------------- */
    String[] cmd4 = {"/bin/sh", "-c", "cd BarApp/libs/BAR && "
        + "python bin/bar login --name a-service-name --role hidden-client --
server " + server};
    process = Runtime.getRuntime().exec(cmd4);

    response.sendRedirect("http://maplocationservice.bar/");
}
```

Ο παρακάτω κώδικας παρουσιάζει τη λίστα με τα τρία BAR services, τα οποία έχουν μία διεύθυνση της μορφής **.bar*, με αποτέλεσμα να διασφαλίζεται η ανωνυμία της πραγματικής IP διεύθυνσής τους.

```
<ul id="services" class="wrapper clearfix">
    <li>
        <a href="imports/call_service.jsp?serv=maplocation"><img src="im-
ages/maplocation_img.png" alt="Img" height="204" width="220"></a>
        <h3><a href="imports/call_service.jsp?serv=maplocation">MapLocation
site</a></h3>
    </li>
    <li>
        <a href="imports/call_service.jsp?serv=profinder"><img src="im-
ages/profinder_img.png" alt="Img" height="204" width="220"></a>
        <h3><a href="imports/call_service.jsp?serv=profinder">ProFinder
site</a></h3>
    </li>
    <li>
        <a href="imports/call_service.jsp?serv=learning"><img src="im-
ages/learning_img.png" alt="Img" height="204" width="220"></a>
        <h3><a href="imports/call_service.jsp?serv=learning">Learning
site</a></h3>
    </li>
</ul>
```

5.3.6 Σελίδα επικοινωνίας

Η συγκεκριμένη σελίδα χρησιμοποιείται για την επικοινωνία του χρήστη με τον διαχειριστή του συστήματος για οποιοδήποτε σχόλιο ή απορία. Έχει υλοποιηθεί μία φόρμα επικοινωνίας, όπως παρουσιάζεται στην παρακάτω εικόνα. Αφού συμπληρώσει ο χρήστης όλα τα στοιχεία και πατήσει το κουμπί “Send Now”, στέλνεται ένα email στον διαχειριστή του συστήματος με όλες τις πληροφορίες που έχουν δηλωθεί.



Εικόνα 5.25 Φόρμα επικοινωνίας

5.4 Ασφάλεια που υποστηρίζεται στην εφαρμογή

Η ασφάλεια διαδραματίζει πολύ σημαντικό ρόλο για τη λειτουργία της εφαρμογής. Πρέπει να προφυλάσσεται από επιθέσεις που έχουν σκοπό να υποκλέψουν ή να αλλοιώσουν στοιχεία χρηστών, δεδομένα, καθώς και να προσπαθήσουν αν μπουν στο σύστημα ως χρήστες χωρίς να είναι εγγεγραμμένοι. Επίσης, πρωταρχικός ρόλος της εφαρμογής είναι η διασφάλιση της ανωνυμίας των χρηστών και της επικοινωνίας τους μέσα σε ένα σύστημα BAR.

Πρώτο βήμα για την ασφάλεια είναι η κωδικοποίηση όλων των κωδικών που χρησιμοποιούνται από τους χρήστες πριν την αποστολή τους στη βάση δεδομένων έτσι ώστε αν μπορέσει κάποιος να "ψαρέψει" έναν κωδικό με τη μέθοδο του *phishing*, να μην καταφέρει να τον αποκωδικοποιήσει. Η συνάρτηση που χρησιμοποιείται για την κρυπτογράφηση είναι η SHA-256 [49], η οποία είναι μονόδρομη. Δηλαδή, αν γνωρίζει κάποιος τον αρχικό κωδικό μπορεί να βρει τον SHA-256 αντίστοιχό του. Αν δεν τον γνωρίζει όμως, είναι πρακτικά ακατόρθωτο να τον αποκωδικοποιήσει.

Για να διασφαλιστεί επιπλέον ασφάλεια του κωδικού του χρήστη και από "*επιθέσεις λεξικών*" *(dictionary attacks)* πέρα από τις υπόλοιπες επιθέσεις που προστατεύονται από την ίδια τη συνάρτηση σύνοψης, χρησιμοποιείται μία πληροφορία (*salt*), η οποία έχει μέγεθος 16 bits και δυαδική μορφή. Η συγκεκριμένη πληροφορία αποθηκεύεται στη βάση δεδομένων, κρυπτογραφημένη με το σύστημα Base64, έναν τρόπο κωδικοποίησης από δυαδικό σε κείμενο.

Κατά τη διαδικασία της εγγραφής του χρήστη, δημιουργείται το salt και κωδικοποιείται, για να αποθηκευτεί στη συνέχεια στη βάση δεδομένων. Αντίθετα, για την επαλήθευση των στοιχείων του χρήστη λαμβάνεται το salt από τη βάση δεδομένων και αποκωδικοποιείται με τη βοήθεια του Base64 decoder, για να χρησιμοποιηθεί στη δημιουργία του ψηφιακού ίχνους του κωδικού που έδωσε ο χρήστης και να συγκριθεί με τον ήδη αποθηκευμένο κρυπτογραφημένο κωδικό.

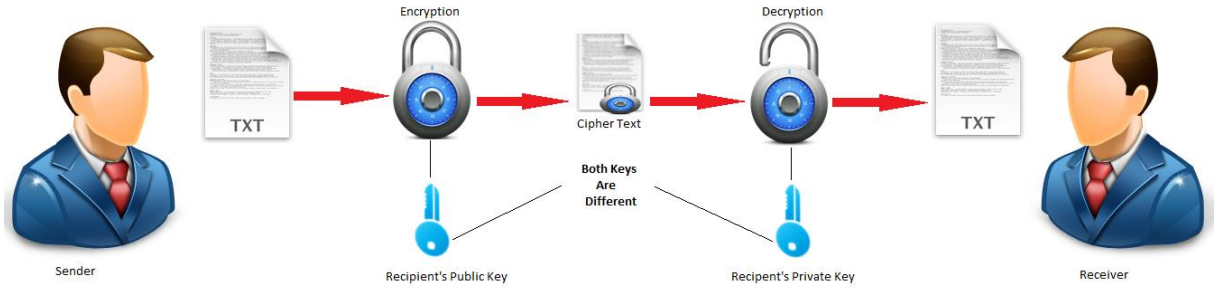
Οι τρεις συναρτήσεις που χρησιμοποιούνται για την κρυπτογράφηση παρουσιάζονται παρακάτω.

```
private static String SHAencrypt (String password, byte[] salt) throws
Exception {
    String encrPass = null;

    MessageDigest md = MessageDigest.getInstance("SHA-256");
    md.update(salt);
    byte[] hash = md.digest(password.getBytes("UTF-8"));

    encrPass = bytesToHex(hash);

    return encrPass;
}

private static byte[] getSalt() throws Exception {
    SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
    byte[] salt = new byte[16];
    sr.nextBytes(salt);

    return salt;
}

public static String bytesToHex(byte[] b) {
    char hexDigit[] = {'0', '1', '2', '3', '4', '5', '6', '7',
                       '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
    StringBuffer buf = new StringBuffer();
    for (int j=0; j<b.length; j++) {
       buf.append(hexDigit[(b[j] >> 4) & 0x0f]);
       buf.append(hexDigit[b[j] & 0x0f]);
    }
    return buf.toString();
}
```

Δεύτερο βήμα για την κατοχύρωση της ασφάλειας του προγράμματος είναι η κωδικοποίηση του ζεύγους των κλειδιών που κρατούνται στην τοπική βάση δεδομένων του χρήστη, καθώς και του Coordinator. Έτσι, χρησιμοποιείται και πάλι η τεχνολογία της Base64 κωδικοποίησης, προκειμένου να μην είναι σε μορφή αναγνώσιμη από κάποιον τρίτο. Ο κώδικας παρακάτω παρουσιάζει τη διαδικασία κωδικοποίησης των RSA κλειδιών μέσα στην εφαρμογή BarApp.

```
public void SaveKeyPair(KeyPair keyPair, String nym) throws IOException {
    PrivateKey privateKey = keyPair.getPrivate();
    PublicKey publicKey = keyPair.getPublic();

    // Create Public Key.
    X509EncodedKeySpec x509EncodedKeySpec = new X509EncodedKeySpec(
                    publicKey.getEncoded());
    String pk =
DatatypeConverter.printBase64Binary(x509EncodedKeySpec.getEncoded());

    // Create Private Key.
    PKCS8EncodedKeySpec pkcs8EncodedKeySpec = new PKCS8EncodedKeySpec(
                    privateKey.getEncoded());
    String sk =
DatatypeConverter.printBase64Binary(pkcs8EncodedKeySpec.getEncoded());

    // Store Keys to DB
    if (nym.equals("")) {
        /* Session Keys are created */
        DBinfo.dbKeysInsert(pk, sk);
    } else {
        /* User's Key Pair are created */
        DBinfo.dbInfoInsert(nym, pk, sk);
    }
}
```

Το επόμενο βήμα και το πιο σημαντικό για την ασφάλεια και την ανωνυμία του χρήστη στο πρωτόκολλο BAR είναι η κρυπτογράφηση των μηνυμάτων που ανταλλάσσονται μεταξύ των χρηστών, αλλά και με τον Coordinator, με τη χρήση ενός αλγορίθμου ασύμμετρου κλειδιού. Έτσι, στην εφαρμογή γίνεται χρήση του πρότυπου *RSA* [55]. Ο κάθε χρήστης έχει ορίσει ένα μόνιμο ζεύγος δημόσιου-ιδιωτικού κλειδιού, το οποίο αποστέλλεται κατά την διαδικασία εγγραφής στον Coordinator. Ακόμα, σε κάθε σύνδεση του χρήστη στο πρωτόκολλο, δημιουργείται ένα επιπλέον ζεύγος από RSA κλειδιά, τα οποία χρησιμοποιούνται για την κρυπτογράφηση των δεδομένων που ανταλλάσσονται ανάμεσα σε χρήστες διαφορετικών Clusters, αλλά και στις διαδρομές onion path routing. Η διαδικασία της κρυπτογράφησης και αποκρυπτογράφησης του RSA αλγόριθμου περιγράφεται στην παρακάτω εικόνα.



Εικόνα 5.26 Αλγόριθμος κρυπτογράφησης RSA

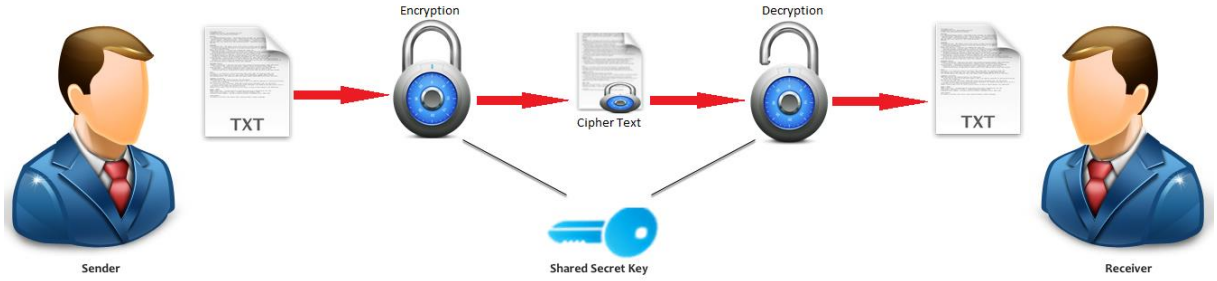
Για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων που στέλνονται στις διαδικασίες ανταλλαγής κλειδιών και στις διαδρομές onion path routing καλείται η κλάση *RSAencrypt* της εφαρμογής. Σε εκείνη, περιλαμβάνονται οι διαδικασίες της κρυπτογράφησης και αποκρυπτογράφησης των δεδομένων με τα χρήση του RSA ασύμμετρου αλγόριθμου.

Το παρακάτω απόσπασμα κώδικα παρουσιάζει τη λειτουργία του αλγόριθμου RSA.

```
public static byte[] RSAenc(String text, PublicKey key) {
    byte[] encDataBytes = new byte[2048];

    try {
      // get an RSA cipher object and print the provider
      final Cipher cipher = Cipher.getInstance(ALGORITHM);
      // encrypt the plain text using the public key
      cipher.init(Cipher.ENCRYPT_MODE, key);
      String AESencText = AESencrypt.AESenc(text);

      encDataBytes =
cipher.doFinal(DatatypeConverter.parseBase64Binary(AESencText));

    } catch (Exception ex) { [...] }

    return encDataBytes;
}
```

```
public static String[] RSAdec(boolean register, String text, PrivateKey key)
{
    String decryptedText = "";
    byte[] decryptedBytes = null;
    String[] parts = null;

    try {
        // get an RSA cipher object and print the provider
        final Cipher cipher = Cipher.getInstance(ALGORITHM);
        // encrypt the plain text using the public key
        cipher.init(Cipher.DECRYPT_MODE, key);
        decryptedBytes =
cipher.doFinal(text.getBytes(StandardCharsets.UTF_8));

        decryptedText = new String(decryptedBytes, StandardCharsets.UTF_8);

        if (register) {
            parts = decryptedText.split(":");
        } else {
            parts = decryptedText.split("-");
        }

    } catch (Exception ex) { [...] }

    return parts;
}
```

Τέλος, στην περίπτωση που οι χρήστες έχουν ανταλλάξει ήδη το κοινό μυστικό κλειδί και label που απαιτείται για την ασφάλεια και ανωνυμία της επικοινωνίας τους, η κρυπτογράφηση των δεδομένων γίνεται με τη χρήση του προτύπου *AES (Advanced Encryption Standard)* [42], που θεωρείται ένας από τους αλγόριθμους συμμετρικού κλειδιού. Οι διαδικασίες της κρυπτογράφησης και αποκρυπτογράφησης με τη συμβολή του συγκεκριμένου αλγόριθμου περιγράφεται στην παρακάτω εικόνα.



Εικόνα 5.27 Αλγόριθμος κρυπτογράφησης AES

Για την κρυπτογράφηση των μηνυμάτων που στέλνονται ανάμεσα σε δύο χρήστες που έχουν ανταλλάξει ήδη ένα κοινό κλειδί, καλείται η συνάρτηση *AESencrypt* της εφαρμογής. Ως κλειδί χρησιμοποιείται το κοινό κλειδί που έχουν ανταλλάξει οι δύο χρήστες, ενώ δημιουργείται και μία τυχαία ακολουθία που αποτελεί το IV (*Initialization Vector)*. Στη συνέχεια, ορίζεται ποιος αλγόριθμος χρησιμοποιείται και αρχικοποιείται. Τέλος, ξεκινάει η διαδικασία της κρυπτογράφησης και η κρυπτογραφημένη πληροφορία μεταδίδεται στο δίκτυο.

Το παρακάτω κομμάτι κώδικα παρουσιάζει τη λειτουργία του αλγόριθμου AES.

```
public static String AESenc(String data) throws Exception {
    getKey();

    // Generate the key
    Key key = generateKey();
    // Create the Cipher
    Cipher c = Cipher.getInstance(ALGORITHM);

    ivSpec = new IvParameterSpec(new byte[16]);

    // Initialize the cipher for encryption
    c.init(Cipher.ENCRYPT_MODE, key, ivSpec);

    byte[] encrData = c.doFinal(data.getBytes());
    String encryptedData = DatatypeConverter.printBase64Binary(encrData);

    return encryptedData;
}

public static String AESdec(String encryptedData, String encryptIV) throws
Exception {
    // Generate the key
    Key key = generateKey();
    // Create the Cipher
    Cipher c = Cipher.getInstance(ALGORITHM);

    ivSpec = new
IvParameterSpec(DatatypeConverter.parseBase64Binary(encryptIV));

    // Initialize the cipher for decryption
    c.init(Cipher.DECRYPT_MODE, key, ivSpec);

    byte[] decordedData = DatatypeConverter.parseBase64Binary(encryptedData);
    byte[] decrData = c.doFinal(decordedData);
    String decryptedData = new String(decrData);

    return decryptedData;
}
```

5.5 Στοιχεία κώδικα εφαρμογής και Web Services

Όπως αναφέρθηκε και στα προηγούμενα κεφάλαια, η αρχική εφαρμογή του συστήματος BAR είναι υλοποιημένη σε γλώσσα Python με τη χρήση της βιβλιοθήκης δικτύωσης Twisted [52].

Στην επέκταση της υλοποίησης και για την εφαρμογή BarApp έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού Java. Η εφαρμογή αποτελείται από κάποια μέρη, τα οποία διαμορφώνουν τη λειτουργικότητά της:

- *Web Pages*

  Για το γραφικό περιβάλλον της εφαρμογής, έχουν υλοποιηθεί κάποιες σελίδες, οι οποίες αποτελούν μέρος του site της. Οι σελίδες αυτές χρησιμοποιούν την τεχνολογία των *JSP (JavaServer Pages)*. Επίσης, υπάρχει ο φάκελος γραμματοσειρών (*fonts*) και ένα αρχείο *CSS (Cascading Style Sheets)*, τα οποία διαμορφώνουν τη μορφοποίηση των σελίδων. Τέλος, σε πολλά σημεία γίνεται χρήση της τεχνολογίας *JavaScript* για τη απαιτούμενη δυναμική υλοποίηση κάποιου μέρους του περιεχομένου.

  Στο σημείο αυτό, θα πρέπει να αναφερθεί ότι οι JSP σελίδες χρειάζονται να υπάρχει κάποιος συμβατός διακομιστής (*Web Server)*, όπως είναι ο Glassfish [53], για την ανάπτυξη και εκτέλεσή τους.

- *Java σελίδες (φάκελος Source Packages)*

  Εκτός από το γραφικό περιβάλλον της εφαρμογής, απαιτείται και ο ορισμός του ελέγχου των διαδικασιών της. Έτσι, έχουν δημιουργηθεί κάποιες Java σελίδες, οι οποίες καλούνται από τις Web Pages και αναλαμβάνουν να εκτελέσουν κάποια διαδικασία. Οι σελίδες αυτές είναι υπεύθυνες για:

- την επικοινωνία της εφαρμογής με τα Web Services του Coordinator,
- την δημιουργία των απαραίτητων *νημάτων (threads)* που θα επιτρέψουν την επικοινωνία μεταξύ των χρηστών,
- την χρήση της τοπικής βάσης δεδομένων που χρειάζεται η εφαρμογή, καθώς και
- την υλοποίηση της ασφάλειας που απαιτείται καθ' όλη τη διάρκεια της λειτουργίας του συστήματος BAR.

- *Βιβλιοθήκες (Libraries)*

  Η εφαρμογή χρησιμοποιεί και κάποιες βιβλιοθήκες Java, οι οποίες επιτρέπουν την προσθήκη κάποιων χρήσιμων λειτουργιών στο πρόγραμμα. Έτσι, χρησιμοποιείται η βιβλιοθήκη *sqlite-jdbc*, η οποία διαχειρίζεται την επικοινωνία της εφαρμογής με την τοπική SQLite βάση δεδομένων. Επίσης, γίνεται χρήση της βιβλιοθήκης *JSON*, προκειμένου να διασφαλιστεί η ανταλλαγή δεδομένων ανάμεσα στην εφαρμογή και τα Web Services.

Όπως αναφέρθηκε και παραπάνω, για την επικοινωνία της εφαρμογής BarApp με τη βάση δεδομένων χρησιμοποιήθηκε ένα Restful Web Service, το οποίο έχει αναπτυχθεί πάνω στον εξυπηρέτη Glassfish του Coordinator. Το Web Service είναι διαχωρισμένο σε κάποιες σελίδες java, οι οποίες καθορίζουν τις οντότητες που χρησιμοποιήθηκαν, ενώ κάποιες άλλες υλοποιούν τις *υπηρεσίες (services)* που χρειάζονται για την επικοινωνία με την εφαρμογή, αλλά και τη βάση δεδομένων. Τέλος, έχουν δημιουργηθεί και κάποιες σελίδες java που αναλαμβάνουν να προσθέσουν στα δεδομένα την απαραίτητη κρυπτογράφηση.

Κεφάλαιο 6

Επίλογος

6.1 Συμπεράσματα

Στην παρούσα διπλωματική, έγινε μία εκτεταμένη ανάλυση του όρου της ανωνυμίας και των συστημάτων της, με απώτερο σκοπό την παρουσίαση των πρωτοκόλλων ανώνυμης επικοινωνίας που έχουν αναπτυχθεί κατά καιρούς. Στη συνέχεια, μελετήθηκε η περίπτωση του πρωτοκόλλου BAR [33], των λειτουργιών του, καθώς και της ήδη υπάρχουσας υλοποίησής του. Κατόπιν, αναπτύχθηκαν οι οντότητες που έχουν περιγραφεί, αλλά δεν έχουν υλοποιηθεί, στον αρχικό σχεδιασμό του συστήματος BAR και απαιτούνται για τη λειτουργία του. Η πιο σημαντική προσθήκη είναι ο κεντρικός συντονιστής (Coordinator) που επιβλέπει τη σωστή χρήση του πρωτοκόλλου και παρέχει την κεντρική βάση δεδομένων. Ακόμα, υλοποιήθηκε η εφαρμογή BarApp [54], η οποία λειτουργεί με τη μορφή ιστότοπου (site) και στοχεύει στην ανάπτυξη των επιμέρους πρωτοκόλλων που απαιτούνται για την ολοκληρωμένη χρήση του συστήματος BAR, όπως είναι αυτό της εγγραφής του χρήστη και της ανταλλαγής κλειδιών. Ένα ακόμα βασικό χαρακτηριστικό της εφαρμογής είναι παροχή γραφικού περιβάλλοντος, το οποίο επιτρέπει στον χρήστη μία εύκολη και συγχρόνως ευχάριστη ανώνυμη περιήγηση. Τέλος, εξετάστηκε η επέκταση του πρωτοκόλλου BAR προκειμένου να επιτρέπεται και η επικοινωνία μεταξύ χρηστών που ανήκουν σε διαφορετικούς διακομιστές, ολοκληρώνοντας έτσι την υλοποίηση του BCP (BAR Communication Protocol) συστήματος.

Το πρωτόκολλο BAR αποτελεί ένα αποδοτικό, ασφαλές και επεκτάσιμο σύστημα για ανώνυμη επικοινωνία στο Διαδίκτυο, παρέχοντας ισχυρή ανωνυμία στους χρήστες του με άριστη και άμεση εμπιστευτικότητα από κάποιον κακόβουλο χρήστη. Σε αντίθεση με τις περισσότερες υλοποιήσεις που κυκλοφορούν σήμερα, το BAR διαφυλάττει και τη μη συνδεσιμότητα των χρηστών που επικοινωνούν, καθώς όλα τα μηνύματα που ανταλλάσσονται, κρυπτογραφούνται και αποστέλλονται στον εξυπηρέτη για τη μετάδοσή τους σε όλους τους χρήστες του συστήματος.

Η αρχική ιδέα ήταν η ανάπτυξη των οντοτήτων, καθώς των απαραίτητων πρωτοκόλλων που απαιτούνται για την ομαλή λειτουργία του πρωτοκόλλου BAR. Παράλληλα, σημαντικό μέρος του πρωτοκόλλου που έπρεπε να υλοποιηθεί, ήταν η επικοινωνία των χρηστών που ανήκουν σε διαφορετικά Clusters. Στη συνέχεια, κρίθηκε απαραίτητη η δημιουργία μίας εφαρμογής που θα επέτρεπε στο χρήστη να χρησιμοποιήσει το πρωτόκολλο BAR, χωρίς την απαίτηση ειδικών γνώσεων. Έτσι, επεκτάθηκε η αρχική σχεδίαση του πρωτοκόλλου, για να εξυπηρετηθούν οι νέες ανάγκες.

Τα αποτελέσματα ήταν ικανοποιητικά πετυχαίνοντας το μεγαλύτερο όγκο των αρχικών στόχων. Η εφαρμογή δοκιμάστηκε και κρίθηκε λειτουργική με περιθώρια βελτίωσης, κυρίως όσον αφορά την εγκατάστασή της. Στα θετικά ήταν η εύκολη χρήση των πρωτοκόλλων εγγραφής, σύνδεσης και ανταλλαγής κλειδιών, καθώς και η ασφαλής επικοινωνία μεταξύ χρηστών του συστήματος. Στα αρνητικά ήταν η δυσκολία εγκατάστασης της εφαρμογής λόγω των απαιτήσεων που υπάρχουν και η αδυναμία ανταλλαγής των απαραίτητων κλειδιών ανάμεσα στους χρήστες και τα BAR services, λόγω της μόνιμης σύνδεσης των τελευταίων.

6.2 Δυνατότητες επέκτασης

Μελλοντικά, είναι απαραίτητη η προσθήκη περισσότερων εξυπηρετών (BAR servers) σε ένα Cluster, προκειμένου να διασφαλιστεί η επεκτασιμότητα του πρωτοκόλλου και η λειτουργία του χωρίς καθυστερήσεις όταν υπάρχουν αρκετοί ενεργοί χρήστες.

Επίσης, είναι εξίσου σημαντική η προσθήκη ενός μηχανισμού στα BAR services, που θα λειτουργεί σαν αυτόνομη υπηρεσία και θα ελέγχει σε τακτά διαστήματα για νέες επαφές. Έτσι, θα επιτευχθεί το πρωτόκολλο ανταλλαγής κλειδιών και ανάμεσα σε χρήστες και υπηρεσίες του πρωτοκόλλου.

Ακόμα, αρκετά σημαντικό στην λειτουργία του πρωτοκόλλου είναι η επικοινωνία σε μη αξιόπιστα δίκτυα. Στην αρχική υλοποίηση, αλλά και στην παρούσα θεωρούμε ότι το δίκτυο είναι αξιόπιστο και δεν υπάρχουν απώλειες μηνυμάτων. Με άλλα λόγια, θα πρέπει να τροποποιηθεί η τοπική βάση δεδομένων της εφαρμογής, προκειμένου να κρατάται και το προηγούμενο κοινό κλειδί που έχουν ανταλλάξει δύο χρήστες.

Έτσι, στην περίπτωση που δεν έχει σταλεί κάποιο μήνυμα (μαζί με το καινούριο κοινό κλειδί), τότε θα δίνεται η δυνατότητα χρήσης του προηγούμενου και η αλλαγή του αμέσως μετά.

Τέλος, θα ήταν αρκετά καλό αν μελλοντικά αναπτυσσόταν μία νέα εφαρμογή, η οποία θα λειτουργούσε ως αυτόνομο πρόγραμμα περιήγησης (όπως είναι η αρχιτεκτονική του Tor), το οποίο θα ήταν αρκετά εύκολο στην εγκατάσταση και δε θα απαιτούσε τη χρήση και άλλων τεχνολογιών, καθώς η παρούσα εφαρμογή απαιτεί την εγκατάσταση ενός εξυπηρέτη, όπως είναι ο Glassfish, για να μπορέσει να λειτουργήσει.

Βιβλιογραφία

1. **Pfitzmann A., Hansen M.**: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology. <http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf> (February 2008)
2. **Chaum D.**: The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. in: Journal of Cryptology 1(1), pp. 65-75 (1988)
3. **Chaum D.**: Untraceable electronic mail, return addresses, and digital pseudonyms. in: Communications of the ACM 24(2) (1981)
4. **Waidner M.**: Unconditional sender and recipient untraceability in spite of active attacks. in: Advances in Cryptology – EUROCRYPT, Lecture Notes in Computer Science, vol. 434, pp. 302–319 (1989)
5. **Dolev S., Otrovsky R.**: Xor-trees for efficient anonymous multicast receiption. in: Advances in Cryptology – CRYPTO'97 (1997)
6. **Goel S., Robson M., Polte M., Sirer E.**: Herbivore: A Scalable and Efficient Protocol for Anonymous Communication, Technical Report 2003-1890, Cornell University, Ithaca, NY (February 2003)
7. **Corrigan-Gibbs H., Ford B., Wolinsky D.I.**: Dissent: Accountable anonymous group messaging. in: Proceedings of the 17th ACM Conference on Computer and Communications Security, CSS'10, pp. 340-350. ACM, New York, NY, USA (2010)
8. **Corrigan-Gibbs H., Ford B., Wolinsky D.I.**: Dining in the Sunshine: Verifiable Anonymous Communication with Verdict. CoRR abs/1209.4819. <http://arxiv.org/abs/1209.4819> (2012)
9. **Pfitzmann B., Pfitzmann A.**: How to break the direct RSA-implementation of MIXes. in: Advances in Cryptology - EUROCRYPT, vol. 434, Lecture Notes in Computer Science, pp. 373–381 (1989)
10. **Danezis G., Dingledine R., Mathewson N.**: Mixminion: design of a type III anonymous remailer protocol. in: IEEE Symposium on Security and Privacy, pp. 2-15 (2003)
11. The anonymizer. <https://www.anonymizer.com>
12. **Berthold O., Federrath H., Köpsell S.**: Web MIXes: a system for anonymous and unobservable Internet access. in: Lecture Notes in Computer Science, pp. 115–129 (2001)
13. **Reed M., Syverson P., Goldschlag D.**: Anonymous connections and onion routing. in: IEEE Journal on Selected Areas in Communications 16(4), pp. 482-494 (1998)
14. **Goldschlag D., Reed M., Syverson P.**: Onion routing for anonymous and private Internet connections. in: Communications of the ACM 42(2), pp. 39-41 (1999)
15. **Dingledine R., Mathewson N., Syverson P.**: Tor: The second-generation onion router. in: Proceedings of the 13th USENIX Security Symposium (August 2004)
16. **Dingledine R., Mathewson N., Syverson P.**: Challenges in deploying low-latency anonymity. in: Technical Report 5540-625, NRL CHACS (2005)
17. **McLachlan J., Tran A., Hopper N., Kim Y.**: Scalable onion routing with torsk. in: Proceedings of the 16th ACM Conference on Computer and Communications Security, CSS'09, pp. 590-599. ACM (2009)
18. **Levine B., Shields C.**: Hordes: a multicast based protocol for anonymity. in: Journal of Computer Security 10(3), pp. 213–240 (2002)
19. **Chen C., Asoni D.E., Barrera D., Danezis G., Perrig A.**: HORNET: high-speed onion routing at the network layer. CoRR abs/1507.05724. <http://arxiv.org/abs/1507.05724> (2015)
20. **Reiter M., Rubin A.**: Crowds: anonymity for web transaction. in: ACM Transactions on Information and System Security 1(1), pp. 66–92 (1998)
21. **Beimel A., Dolev S.**: Buses for anonymous message delivery. in: Journal of Cryptology 16, pp. 25-39 (2003)
22. **Hirt A., Jacobson Jr M.J., Williamson C.L.**: A practical buses protocol for anonymous internet communication. in: PST. Citeseer (2005)

23. **Hirt A., Jacobson M., Williamson C.**: Taxis: scalable strong anonymous communication. in: Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on, pp. 1-10. IEEE (2008)
24. **Young A.L., Yung M.**: The drunk motorcyclist protocol for anonymous communication. in: Communications and Network Security (CNS), 2014 IEEE Conference on. pp. 157-165. IEEE (2014)
25. **Freedman M.J., Morris R.**: Tarzan: A peer-to-peer anonymizing network layer. in: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS'02 (2002)
26. **Rennhard M., Plattner B.**: Practical anonymity for the masses with morphmix. in: Juels, A. (ed.) Proceedings of Financial Cryptography (FC '04). pp. 233-250. Springer-Verlag, LNCS 3110 (2004)
27. **Sherwood R., Bhattacharjee B., Srinivasan A.**: P^5: A protocol for anonymous communications. Journal of Computer Security, IOS Press 13(6), pp. 839-876 (2005)
28. **Warren J.**: Bitmessage: A peer-to-peer message authentication and delivery system. white paper (27 November 2012). <https://bitmessage.org/bitmessage.pdf> (2012)
29. **Stajano F., Anderson R.J.**: The cocaine auction protocol: On the power of anonymous broadcast. in: Pfitzmann A. (ed.) Information Hiding, Lecture Notes in Computer Science, vol. 1768, pp. 434-447. Springer (1999)
30. **Pfitzmann A., Pfitzmann B., Waidner M.**: ISDN-mixes: untraceable communication with very small bandwidth overhead. in: Proceedings of the GI/ITG Conference on Communication in Distributed Systems, pp. 451-463 (February 1991)
31. **Jerichow A., Müller J., Pfitzmann A., Pfitzmann B., Waidner M.**: Real-time MIXes: a bandwidth-efficient anonymity protocol. in: IEEE Journal on Selected Areas in Communications 16(4) (1998)
32. **Kwon A., Lazar D., Devadas S., Ford B.**: Riffle: An efficient communication system with strong anonymity. in: Proceedings on Privacy Enhancing Technologies 2016 (2), pp. 1-20 (2016)
33. **Kotzanikolaou P., Chatzisofroniou G., Burmester M.**: Broadcast anonymous routing (BAR): scalable real-time anonymous communication. in: International Journal of Information Security. Springer-Verlag (February 2016)
34. BAR protocol. <https://sophron.github.io/BAR>
35. **Ren J.,Wu J.**: Survey on anonymous communications in computer networks. in: Computer Communications 33(4), pp. 420–431 (2010)
36. **Sampigethaya K., Poovendran R.**: A Survey on Mix Networks and Their Secure Applications. in: Proceedings of the IEEE 94(12), pp. 2142-2181 (December 2006)
37. **Danezis G.**: Designing and attacking anonymous communication systems – Technical Report no 594. University of Cambridge. <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-594.html> (July 2004)
38. **Feistel H.**: Block cipher cryptographic system. U.S. Patent No 3,798,359 (1974)
39. **Davis R.M.**: The data encryption standard in perspective. in: Communications Society Magazine. IEEE 16(6), pp. 5-9 (1978)
40. ANSI X9.52-1998. Triple Data Encryption Algorithm Modes of Operation (1998)
41. **Rivest R.L.**: The RC5 Encryption Algorithm. in: Proceedings of the Second International Workshop on Fast Software Encryption (FSE). pp. 86-96 (1994)
42. FIPS PUB 197. The official AES standard. <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf> (2001)
43. **Schneier B.**: Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). in: Fast Software Encryption, Cambridge Security Workshop Proceedings (Springer-Verlag), pp. 191-204 (1993)
44. **Rivest R., Schuldt J.**: Spritz - a spongy RC4-like stream cipher and hash function (2014)
45. eStream. <https://en.wikipedia.org/wiki/ESTREAM>
46. **Rivest R.**: The MD4 message-digest algorithm. <http://tools.ietf.org/html/rfc1320> (1992)
47. **Rivest R.**: The MD5 message-digest algorithm. <http://tools.ietf.org/html/rfc1321> (1992)

48. **Eastlake D., Jones P.**: US secure hash algorithm 1 (SHA1). <http://www.hjp.at/doc/rfc/rfc3174.html> (2001)
49. FIPS 180-4. Secure Hash Standard (SHS). <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf> (2015)
50. FIPS PUB 198-1. The Keyed-Hash Message Authentication Code (HMAC). <http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf> (2008)
51. **Diffie W., Hellman M.E.**: New directions in cryptography. in: Information Theory, IEEE Transactions, 22(6), pp. 644-654 (1976)
52. Twisted Library. <https://twistedmatrix.com>
53. Glassfish Web Server. <https://glassfish.java.net>
54. Extended BAR protocol – BarApp. <https://github.com/marym92/BAR-extended>
55. **Rivest R.L., Shamir A., Adleman L.M.**: U.S. Patent No. 4,405,829. Washington DC: U.S. Patent and Trademark Office (1983)
56. **Palme J., Berglund M.**: Anonymity on the Internet (August 2009)
57. I2P. <https://geti2p.net/en>