Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών

Πληροφοριακά και Επικοινωνιακά Συστήματα

Μεταπτυχιακή Διατριβή

Προσθήκη Ασφάλειας σε Κατανεμημένο Σύστημα Παρακολούθησης Υπολογιστικής Αρχιτεκτονικής Νέφους

Γιώργος Ρηγόπουλος

Επιβλέπων Καθηγητής
Δημήτρης Αντωνιάδης

Σεπτέμβριος 2015

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών

Πληροφοριακά και Επικοινωνιακά Συστήματα

Μεταπτυχιακή Διατριβή

Προσθήκη Ασφάλειας σε Κατανεμημένο Σύστημα Παρακολούθησης Υπολογιστικής Αρχιτεκτονικής Νέφους

Γιώργος Ρηγόπουλος

Επιβλέπων Καθηγητής
Δημήτρης Αντωνιάδης

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Πληροφοριακά και Επικοινωνιακά Συστήματα από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Σεπτέμβριος 2015

Περίληψη

Τα συστήματα υπολογιστικής νέφους (cloud) έχουν εδραιωθεί τα τελευταία χρόνια στην κοινωνία της πληροφορικής ως η λύση στο απαιτητικό και δυναμικό περιβάλλον του σύγχρονου κόσμου. Ο κύριος λόγος είναι η ανάγκη για άμεση απομακρυσμένη πρόσβαση στους υπολογιστικούς πόρους και στα εργαλεία που προσφέρονται από αυτήν. Αποτέλεσμα είναι οι ανάγκες για δυναμική διαχείριση και διαμοιρασμό των υπολογιστικών πόρων να αυξάνονται συνεχώς, τόσο σε μικρομεσαίες επιχειρήσεις και οργανισμούς όσο και στις πολύ μεγάλες. Στόχος της παρούσας μεταπτυχιακής διατριβής είναι η προσθήκη ασφάλειας κατά την μετάδοση των δεδομένων του λογισμικού παρακολούθησης υπολογιστικής νέφους JCatascopia. Για το σκοπό αυτό έγινε ποσοτική έρευνα μέσω πειραμάτων σε ένα σύστημα υπολογιστικής νέφους. Μέσα από βιβλιογραφική έρευνα, επιλέχθηκε για την υλοποίηση ο αλγόριθμος κρυπτογράφησης AES ως ο πλέον ασφαλής. Διαπιστώθηκε ότι ο κρυπτογραφικός αλγόριθμος AES είναι ο καταλληλότερος για την κρυπτογράφηση των δεδομένων καθώς η απαίτηση σε υπολογιστικούς πόρους είναι αμελητέα, κάτι πολύ κρίσιμο σε ένα σύστημα όπως το JCatascopia.

Summary

Cloud computing has been established last years in IT community as the solution for the demanding and dynamic environment of the modern world. The main reason is the need for immediate remote access to the computing resources and the tools that cloud computing offers. As a result the need for dynamic management and computing resource sharing to be continually increasing, both to small-middle size businesses - organizations and so to the big ones. The goal of this dissertation is to add security during data transmission of JCatascopia, a cloud monitoring system. Quantitative research through experimental method has taken place. Through bibliography, AES cryptographic algorithm selected as the safer. Through experiments found that AES requirements are very low in resources, which make him the most suitable for a system as JCatascopia.

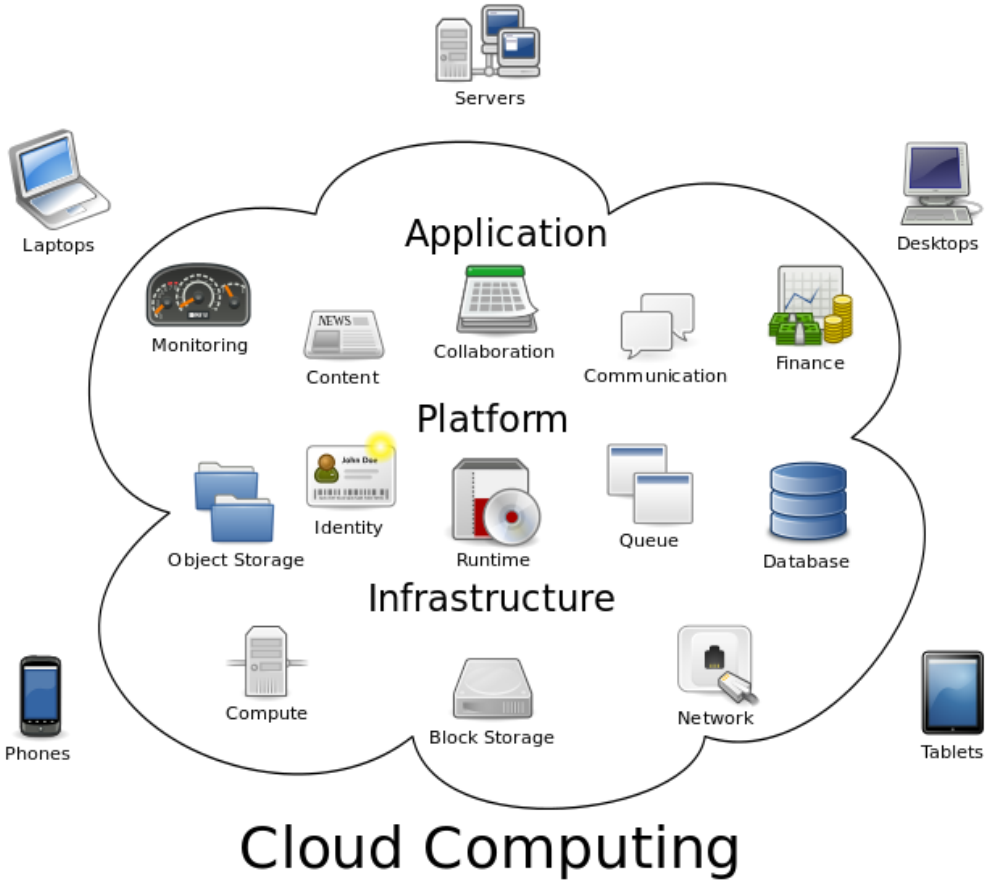
Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω την αρραβωνιαστικιά μου, Γκιούρα Σωτηρία, για την υπομονή της και την προτροπή της ώστε να ολοκληρώσω αυτήν τη διατριβή. Στη συνέχεια θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Αντωνιάδη Δημήτρη, που με στήριξε στην υλοποίησή της και για το πνεύμα συνεργασίας που επέδειξε.

Πίνακας περιεχομένων

Κεφάλαιο 1
Εισαγωγή

Τα συστήματα υπολογιστικής νέφους (cloud) έχουν εδραιωθεί τα τελευταία χρόνια στην κοινωνία της πληροφορικής ως η λύση στο απαιτητικό και δυναμικό περιβάλλον του σύγχρονου κόσμου. Ο κύριος λόγος είναι η ανάγκη για άμεση απομακρυσμένη πρόσβαση στους υπολογιστικούς πόρους και στα εργαλεία που προσφέρονται από αυτήν. Αποτέλεσμα είναι οι ανάγκες για δυναμική διαχείριση και διαμοιρασμό των υπολογιστικών πόρων να αυξάνονται συνεχώς, τόσο σε μικρομεσαίες επιχειρήσεις και οργανισμούς όσο και στις πολύ μεγάλες. Σύμφωνα με τη Eurostat μία στις πέντε εταιρείες το 2014 έκαναν χρήση υπηρεσιών υπολογιστικής νέφους [1]. Η αρχιτεκτονική ενός συστήματος υπολογιστικής νέφους αποτελείται από το φυσικό επίπεδο και το αφαιρετικό επίπεδο. Το φυσικό είναι κυρίως το υλικό όπως οι εξυπηρετητές, ο αποθηκευτικός χώρος, οι συσκευές δικτύου και τα λειτουργικά αυτών. Το αφαιρετικό επίπεδο «κάθεται» πάνω στο φυσικό επίπεδο και στην πράξη είναι το λογισμικό που αναπτύσσεται και παρέχει την υπηρεσία στον τελικό χρήστη. Στο σχήμα 1 φαίνονται τα μοντέλα υπηρεσίας της υπολογιστικής νέφους και κάποια παραδείγματα για το κάθε ένα από αυτά.



Σχήμα 1: Μεταφορικά η Υπολογιστική Νέφους (Wikipedia, 2015, Cloud Computing [2]).

Η διαχείριση αυτών των συστημάτων είναι αρκετά σύνθετη λόγο της φύσης της αρχιτεκτονικής τους είτε αφορά Λογισμικό ως μια Υπηρεσία (Software as a Service, SaaS), είτε Πλατφόρμα ως μια Υπηρεσία (Platform as a Service, PaaS), είτε Υποδομή ως μια Υπηρεσία (Infrastructure as a Service, IaaS).

Ένα σύστημα Υπολογιστικής Νέφους απαιτεί διαρκής παρακολούθηση (monitoring) από λογισμικό ώστε α. να δίνει τη δυνατότητα στους διαχειριστές, προσωπικό ή συστήματα που έχουν το ρόλο διαχειριστή, να παίρνουν αποφάσεις διαχειριστικού χαρακτήρα όπως σωστή αναδιάταξη πόρων, β. να προλαμβάνονται προβλήματα γ. να γίνεται άμεση αποκατάσταση προβλημάτων πριν την επέκτασή τους.

Το JCatascopia είναι ένα Σύστημα Παρακολούθησης Υπολογιστικής Νέφους πολλαπλών επιπέδων που παρέχει δια-λειτουργικότητα, τρέχει με τρόπο διαφανή,

δημιουργεί υψηλού επιπέδου μετρικές και είναι ανοικτού κώδικα [3]. Το σύστημα αποτελείται από τουλάχιστον έναν ή περισσότερους εξυπηρετητές και έναν ή περισσότερους πράκτορες. Ο κάθε πράκτορας είναι μια διαφανής εφαρμογή που τρέχει σε κάθε έναν πελάτη της υπολογιστικής νέφους και στέλνει μετρικές όπως π.χ. το ποσοστό της CPU που χρησιμοποιείται, στον εξυπηρετητή, ανά τακτά χρονικά διαστήματα. Ο εξυπηρετητής συλλέγει αυτές τις μετρικές από όλους τους πράκτορες σε μια βάση δεδομένων και προσφέρει ένα σημείο διεπαφής στον διαχειριστή, ο οποίος μπορεί να δει σε γραφικές παραστάσεις υψηλού επιπέδου τις μετρικές που έχουν συλλεχθεί. Η προσθαφαίρεση των πρακτόρων-πελατών γίνεται χωρίς την παρέμβαση του διαχειριστή. Σε αυτή τη μεταπτυχιακή διατριβή προστίθεται ασφάλεια κατά τη μετάδοση των δεδομένων και θέτουμε τις βάσεις για την πολλαπλή μίσθωση στο JCatascopia μέσω υποδομής δημοσίου κλειδιού RSA[1]. Μελετήθηκαν άλλα συστήματα που προσφέρουν παρόμοια λειτουργικότητα ή και ίδια και διαπιστώθηκε το κενό όσον αφορά την πληρότητα των χαρακτηριστικών που απαιτείται από αυτά στην υπολογιστική νέφους. Δια μέσω της μελέτης διάφορων ασφαλών μεθόδων μετάδοσης και κρυπτογράφησης δεδομένων προσδιορίσαμε την πιο κατάλληλη για ένα περιβάλλον παρακολούθησης υπολογιστικής αρχιτεκτονικής νέφους.

1.1 Λόγος Της Ερευνητικής Εργασίας

Η καινοτομία οποιοδήποτε συστήματος παρακολούθησης υπολογιστικής αρχιτεκτονικής νέφους στηρίζεται στην ανάγκη να είναι ελαφρύ και να μην παρεμβάλλεται στο σύστημα ή στην εφαρμογή που παρακολουθεί. Εξορισμού, οι μέθοδοι ασφαλούς μεταφοράς και κρυπτογράφησης θεωρούνται ότι καταναλώνουν πολλούς πόρους, για αυτό το λόγο και εντοπίζουμε εκείνες που ταιριάζουν σε ένα περιβάλλον υπολογιστικής αρχιτεκτονικής νέφους.

[1] https://en.wikipedia.org/wiki/RSA_%28cryptosystem%29

1.2 Σπουδαιότητα της Έρευνας

Η σπουδαιότητα της έρευνας έγκειται στο ότι δεν υπάρχει καμία εφαρμογή παρακολούθησης υπολογιστικής νέφους ανοικτού κώδικα που παρέχει τον συνδυασμό όλων των παρακάτω χαρακτηριστικών και παράλληλα ασφαλούς μετάδοσης δεδομένων.

- Μικρό αποτύπωμα εκτέλεσης
- Χαμηλό κόστος
- Ελαστικότητα
- Εύκολη εφαρμογή
- Υψηλή ασφάλεια
- Εφαρμογή σε κάθε υποδομή υπολογιστικής αρχιτεκτονικής νέφους

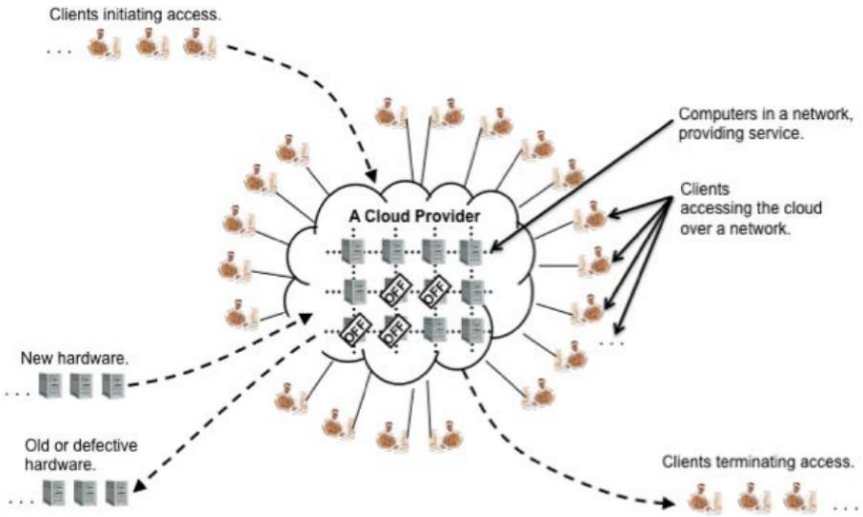
1.3 Ορολογία

Οι παρακάτω ορισμοί είναι η βάση για την περαιτέρω ανάλυση της λειτουργίας και της ασφαλούς μετάδοσης των δεδομένων του συστήματος παρακολούθησης υπολογιστικής νέφους JCatascopia, του οποίου τη λειτουργία θα αναλύσουμε σε επόμενο κεφάλαιο, καθώς θα είναι και το κυρίως θέμα αυτής της μεταπτυχιακής διατριβής.

1.3.1 Τι είναι η Υπολογιστική Αρχιτεκτονική Νέφους (Cloud Computing);

Υπολογιστική Νέφους (Cloud computing) είναι ένα μοντέλο που παρέχει τη δυνατότητα πρόσβασης μέσω δικτύου από κάθε μέσο, με ευκολία και κατ απαίτηση σε μια κοινόχρηστη δεξαμενή ρυθμιζόμενων υπολογιστικών πόρων (π.χ. δίκτυα, εξυπηρετητές, αποθηκευτικούς χώρους, εφαρμογές, και υπηρεσίες) που μπορούν με πολύ γρήγορο τρόπο να παραχθούν και να δοθούν προς χρήση με ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδρασης με τον πάροχο. Αυτό το μοντέλο υπολογιστικής νέφους αποτελείται από πέντε ουσιώδης χαρακτηριστικά, τρία

μοντέλα υπηρεσίας και τέσσερα μοντέλα ανάπτυξης. Η περιγραφή αυτή απεικονίζεται στο σχήμα 2.



Σχήμα 2. Γενική όψη της Υπολογιστικής Αρχιτεκτονικής Νέφους και των καταναλωτών. [4]

Σύμφωνα με το διεθνή οργανισμό NIST (National Institute of Standards and Technology) [4] τα χαρακτηριστικά αυτά είναι:

- Κατ' απαίτηση αυτοεξυπηρέτηση (On-demand self-service)

  Ο καταναλωτής μπορεί μονομερώς να κάνει χρήση υπολογιστικών δυνατοτήτων, όπως είναι χρήση για συγκεκριμένο χρονικό διάστημα ενός εξυπηρετητή και δικτυακού αποθηκευτικού χώρου, σύμφωνα με τις ανάγκες του, αυτόματα, χωρίς την απαίτηση ανθρώπινης αλληλεπίδρασης με τον πάροχο κάθε μιας από αυτές τις υπηρεσίες.

- Ευρεία πρόσβαση μέσω δικτύου (Broad network access)

  Οι δυνατότητες είναι διαθέσιμες μέσω δικτύου και προσπελάσιμες μέσω σταθερών μηχανισμών που προωθούν τη χρήση από ετερογενείς «ελαφριές» ή «βαριές» πλατφόρμες πελατών (π.χ. κινητά τηλέφωνα, ταμπλέτες, φορητούς Η/Υ και σταθμούς εργασίας).

- Δεξαμενή πόρων (Resource pooling)

Οι υπολογιστικοί πόροι του πάροχου ομαδοποιούνται για την εξυπηρέτηση πολλαπλών καταναλωτών με τη χρήση ενός μοντέλου πολλαπλής μίσθωσης, με διαφορετικούς φυσικούς και εικονικούς πόρους δυναμικά μοιρασμένους και αναδιαταγμένους σύμφωνα με τις απαιτήσεις των καταναλωτών. Υπάρχει μια αίσθηση ανεξαρτησίας από την τοποθεσία στην οποία ο πελάτης γενικά δεν έχει κανέναν έλεγχο ή γνώση για την ακριβή τοποθεσία των παρεχόμενων πόρων, αλλά ίσως έχει τη δυνατότητα να καθορίσει την τοποθεσία σε πιο υψηλό επίπεδο αφαιρετικού τύπου (π.χ. χώρα, πολιτεία ή κέντρο δεδομένων). Παραδείγματα των πόρων περιλαμβάνουν αποθηκευτικό χώρο, επεξεργαστική ισχύ, μνήμη και εύρος δικτύου.

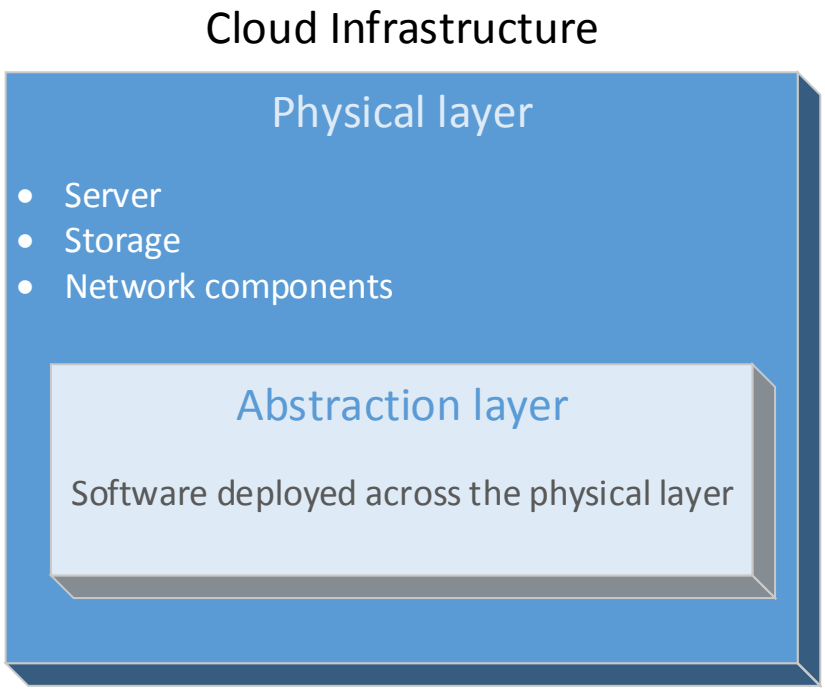
- Υψηλή ελαστικότητα (Rapid elasticity)

  Οι δυνατότητες μπορούν να διατεθούν ραγδαία και με ελαστικότητα, σε κάποιες περιπτώσεις αυτόματα, να κλιμακωθούν ραγδαία προς τα έξω και προς τα μέσα αναλόγως των απαιτήσεων. Στον καταναλωτή, συχνά αυτές οι ικανότητες φαίνεται να μπορούν να ιδιοποιηθούν σε οποιαδήποτε ποσότητα και οποιαδήποτε στιγμή.

- Μετρούμενη υπηρεσία (Measured service)

  Τα συστήματα υπολογιστικής νέφους αυτόματα ελέγχουν και βελτιστοποιούν τη χρήση των πόρων μοχλεύοντας μια ικανότητα μέτρησης σε κάποιο αφαιρετικό επίπεδο κατάλληλο του τύπου υπηρεσίας (π.χ. αποθηκευτικός χώρος, επεξεργαστική ισχύ, εύρος και ενεργούς λογαριασμούς χρηστών). Η χρήση των πόρων μπορεί να παρακολουθείται, να ελέγχεται και να αναφέρεται, παρέχοντας διαφάνεια τόσο στον πάροχο όσο και στον καταναλωτή της χρησιμοποιούμενης υπηρεσίας.

Στο σχήμα 3 βλέπουμε την υποδομή της υπολογιστικής αρχιτεκτονικής νέφους σύμφωνα με το NIST [5].



Σχήμα 3. Υποδομή Υπολογιστικής Αρχιτεκτονικής Νέφους. [5]

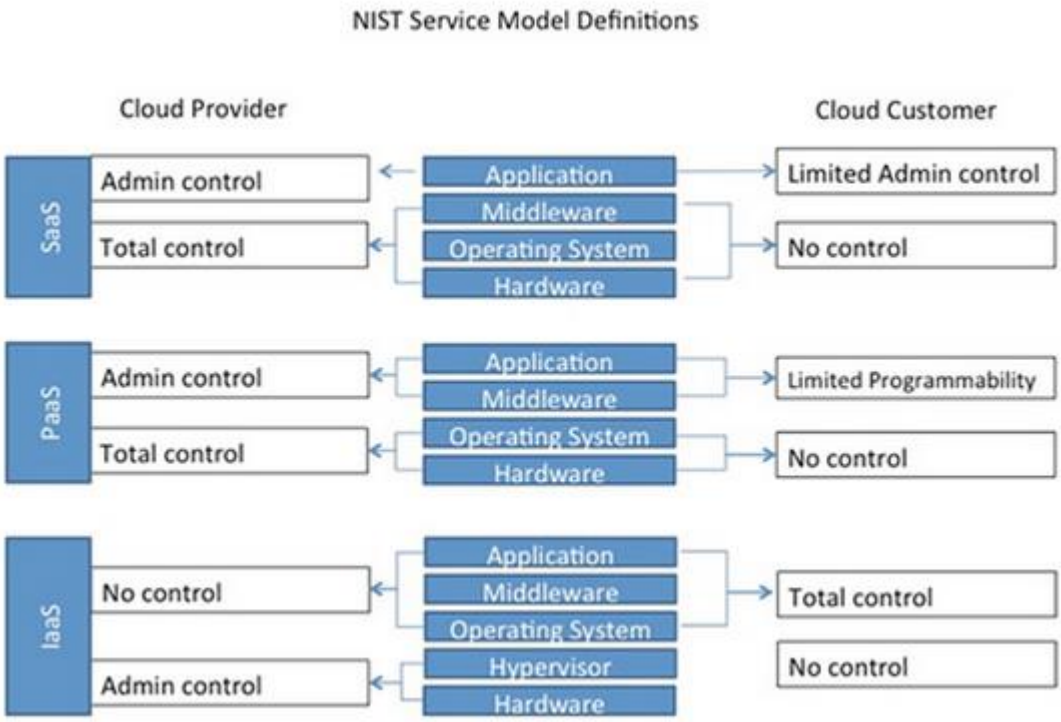
Σύμφωνα με το NIST τα μοντέλα υπηρεσίας είναι τρία, φαίνονται στο σχήμα 4 και περιγράφονται ως εξής:

- Λογισμικό ως μια Υπηρεσία (Software as a Service, SaaS)
  Ο τελικός καταναλωτής έχει τη δυνατότητα να χρησιμοποιεί μια εφαρμογή που εκτελείται σε μια υποδομή υπολογιστικής νέφους. Έχει πολύ περιορισμένες δυνατότητες ελέγχου και αφορούν ρυθμίσεις της εφαρμογής σε επίπεδο χρήστη και καμία άλλη όσον αφορά το φυσικό επίπεδο.

- Πλατφόρμα ως μια Υπηρεσία (Platform as a Service, PaaS)
  Ο τελικός καταναλωτής έχει τη δυνατότητα να αναπτύξει στην υποδομή αρχιτεκτονικής νέφους εφαρμογές που έφτιαξε ή επίκτητες εφαρμογές. Ο πάροχος του δίνει τη δυνατότητα να κάνει ρυθμίσεις εφαρμογής μόνο στο περιβάλλον φιλοξενίας της και όχι στο φυσικό επίπεδο.

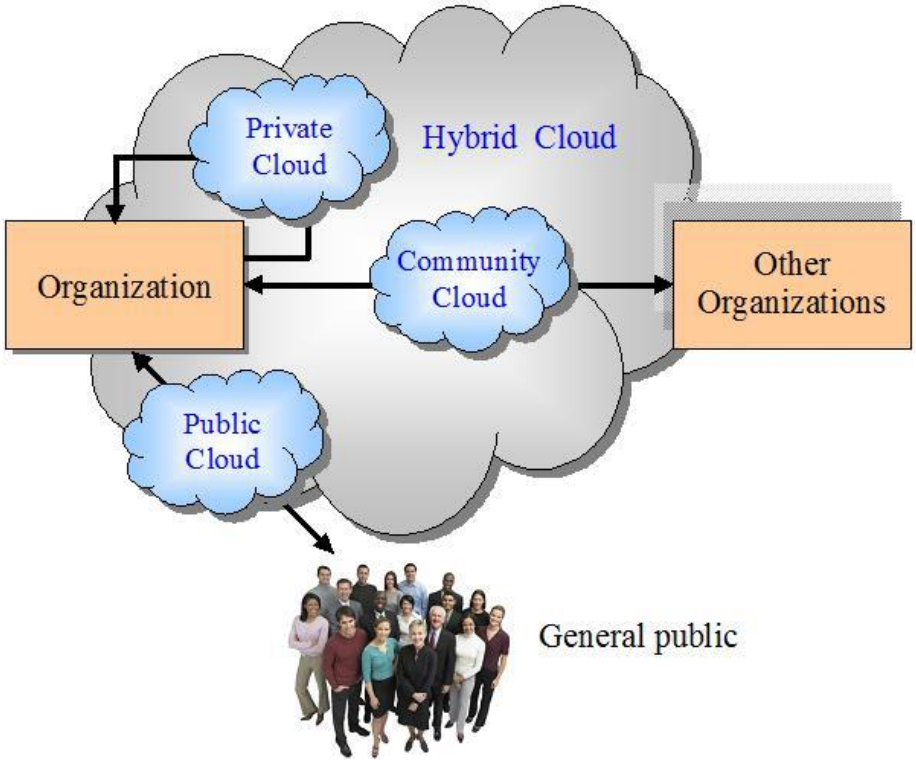
- Υποδομή ως μια Υπηρεσία (Infrastructure as a Service, IaaS)
  Ο τελικός καταναλωτής μπορεί να χρησιμοποιήσει πόρους όπως αποθηκευτικό χώρο, επεξεργαστική ισχύς, δίκτυα και άλλες θεμελιώδεις υπολογιστικούς

πόρους για να αναπτύξει και να εκτελέσει αυθαίρετα λειτουργικά συστήματα και εφαρμογές. Ο καταναλωτής δεν διαχειρίζεται ή ελέγχει την υποκείμενη υποδομή (π.χ. τον Hypervisor και τη φυσική υποδομή του δικτύου) αλλά έχει τον έλεγχο των λειτουργικών συστημάτων, του αποθηκευτικού χώρου, των εφαρμογών και πιθανώς περιορισμένο έλεγχο υψηλότερου επίπεδου κάποιων στοιχείων του δικτύου.



Σχήμα 4. Τα μοντέλα υπηρεσίας της Υπολογιστικής Αρχιτεκτονικής Νέφους. [5]

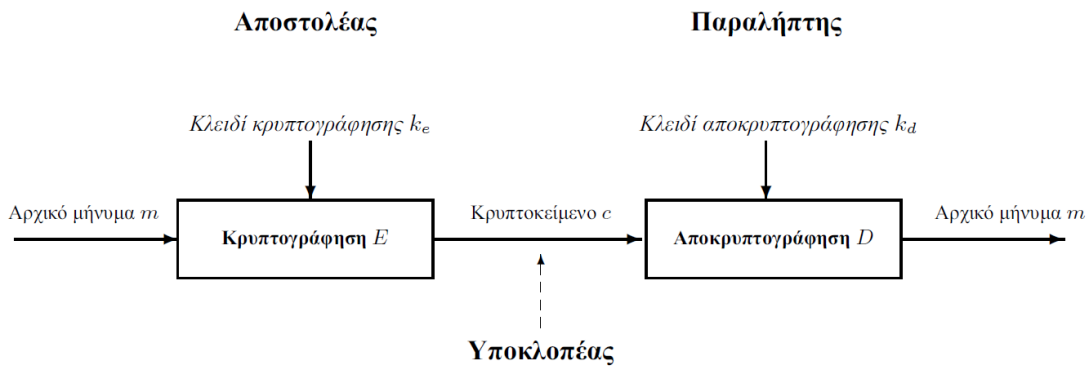
Τα μοντέλα ανάπτυξης της Υπολογιστικής Νέφους είναι τέσσερα και είναι ευδιάκριτα στο σχήμα 5. Το σημείο διαφοροποίησής τους είναι το κοινό από το οποίο είναι προσπελάσιμοι κάθε φορά οι πόροι και οι εφαρμογές της υπολογιστικής νέφους.



Σχήμα 5. : Τα μοντέλα ανάπτυξης της υπολογιστικής νέφους. [6]

1.3.2 Κρυπτογραφία

Πολύ συχνά ένα σύστημα πληροφορικής χρειάζεται να παρέχει υπηρεσίες ασφάλειας όπως εμπιστευτικότητα, ακεραιότητα δεδομένων, πιστοποίηση ταυτότητας χρήστη, εξουσιοδότηση και μη αποποίηση. Για να επιτευχθεί αυτό χρησιμοποιούνται μαθηματικές συναρτήσεις οι οποίες μετασχηματίζουν τα δεδομένα από αναγνώσιμα σε μη αναγνώσιμα ή αλλιώς κρυπτογραφημένα. Η μαθηματική συνάρτηση, που υλοποιείται από έναν αλγόριθμο, χρειάζεται για τον μετασχηματισμό αυτόν και μια ακόμα παράμετρο, δηλαδή ένα κλειδί. Το κλειδί είναι μια παράμετρος η οποία είναι γνωστή μόνο στον αποστολέα και ενίοτε, ανάλογα από τον τύπο της κρυπτογράφησης και στον παραλήπτη. Οι τρεις βασικές κατηγορίες των κρυπτογραφικών αλγορίθμων είναι : κρυπτογραφικές συναρτήσεις κατακερματισμού, αλγόριθμοι συμμετρικού κλειδιού και αλγόριθμοι ασύμμετρου κλειδιού (ή Δημοσίου κλειδιού) [7]. Στο σχήμα 6 φαίνεται το γενικό μοντέλο της κρυπτογράφησης δεδομένων.



Σχήμα 6. Τυπικό διάγραμμα κρυπτογραφικού συστήματος [4].

Οι συμμετρικοί αλγόριθμοι χρησιμοποιούνε το ίδιο κλειδί για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων. Επομένως ο αποστολέας και ο παραλήπτης κάνουν χρήση ακριβώς του ίδιου κλειδιού για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων αντίστοιχα. Αυτοί οι αλγόριθμοι λειτουργούνε σε κομμάτια δεδομένων κατά την κρυπτογράφηση και την αποκρυπτογράφηση. Εάν κάποιος τρίτος, που δεν γνωρίζει το κλειδί, υποκλέψει δεδομένα δεν είναι σε θέση να δει τα περιεχόμενα σε αναγνώσιμη μορφή ακόμη και αν γνωρίζει τον αλγόριθμο που έχει χρησιμοποιηθεί.

Οι αλγόριθμοι αυτοί χρησιμοποιούνται για να παρέχουν εμπιστευτικότητα μέσω της κρυπτογράφησης ή για την ανταλλαγή κλειδιών.

Οι έγκυροι αλγόριθμοι είναι ο Advanced Encryption Standard (AES) και ο Triple Data Encryption Algorithm (TDEA) ο οποίος βασίζεται στον Data Encryption Standard (DES). Αυτοί λειτουργούνε σε κομμάτια δεδομένων για την κρυπτογράφηση και την αποκρυπτογράφηση και ονομάζονται και κρυπταλγόριθμοι τμήματος.

Το αρχικό μήνυμα χωρίζεται σε ίσου μεγέθους τμήματα. Η κρυπτογράφηση γίνεται πάνω σε κάθε ένα από αυτά τα τμήματα και το αποτέλεσμα είναι ισάριθμα τμήματα κρυπτογραφημένων δεδομένων ίσου μεγέθους. Το ίδιο συμβαίνει και κατά την αποκρυπτογράφηση.

Στον κρυπταλγόριθμο τμήματος γίνονται αντικαταστάσεις τμημάτων από bit. Σήμερα έχει καθιερωθεί σαν μέγεθος τμήματος τα 128 bit, και αντίστοιχου

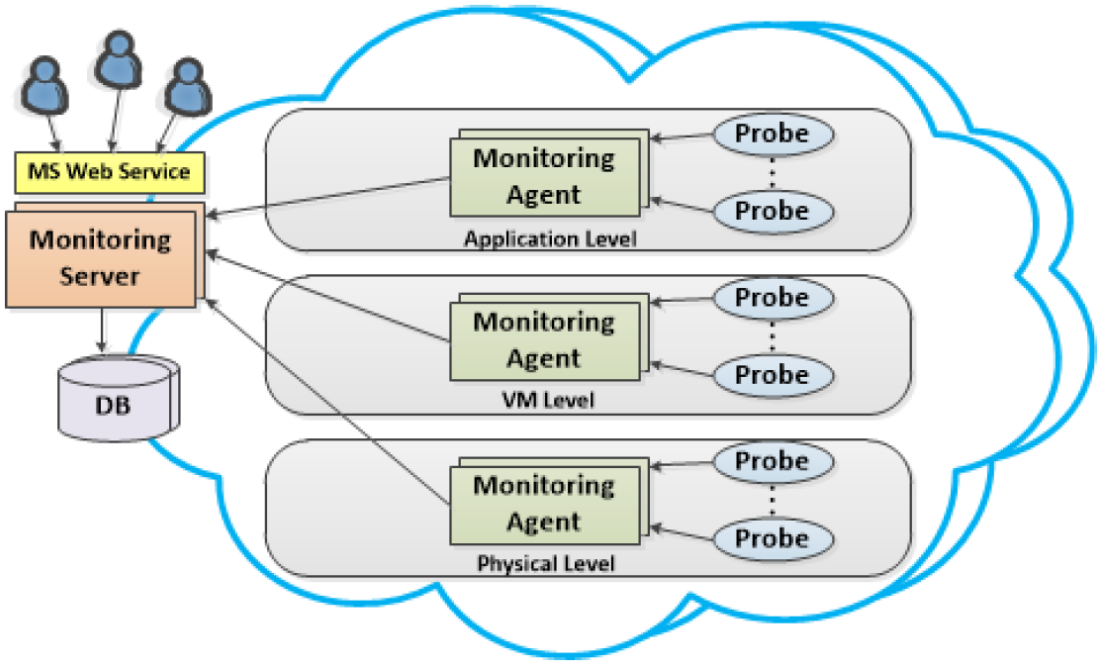
μεγέθους είναι και το κλειδί ώστε τα πιθανά κλειδιά να είναι 2^{128}. Στη λειτουργία του αλγορίθμου εισάγεται μια μονάδα αντικατάστασης που ονομάζεται S-Box και πραγματοποιεί αντικαταστάσεις bit ώστε να προσθέσει σύγχυση. Εισάγεται επίσης μια μονάδα αντιμετάθεσης bit (P-Box) η οποία χρησιμοποιείται σε πολλαπλά στάδια και εισάγει διάχυση στην κρυπτογράφηση των δεδομένων [4].

1.4 Το σύστημα παρακολούθησης υπολογιστικής νέφους JCatascopia

Οι περισσότεροι πάροχοι υπολογιστικής νέφους έχουν αναπτύξει εργαλεία παρακολούθησης τα οποία είναι προσαρμοσμένα στο λογισμικό και στο υλικό που χρησιμοποιεί και υποστηρίζει ο κάθε πάροχος ξεχωριστά. Οι εταιρείες που αναπτύσσουν το υλικό που απευθύνεται σε παρόχους υπολογιστικής νέφους προχωρούν με την ίδια λογική, αναπτύσσοντας λογισμικό παρακολούθησης μόνο για το δικό τους υλικό, στοχευόμενο συνήθως στο φυσικό επίπεδο. Το σύστημα παρακολούθηση JCatascopia προσφέρει το πλεονέκτημα του ανοικτού κώδικα που σε συνδυασμό με την ελαστικότητα και την προσαρμοστικότητα δίνει στον διαχειριστή της υποδομής εκείνο το εργαλείο που χρειάζεται για να μην αναλώνει χρόνο στη λήψη των μετρήσεων αλλά στη λήψη αποφάσεων με βάση αυτές τις μετρήσεις.

Λόγω της Java με την οποία έχει αναπτυχθεί λειτουργεί σε οποιοδήποτε συνδυασμό λειτουργικού συστήματος – υλικού υπολογιστικής νέφους. Συλλέγει πληροφορίες που αφορούν τις αποδόσεις του υλικού, του λογισμικού και του δικτύου σε όλα τα επίπεδα, δίνοντας έτοιμο API για την προσθήκη νέων σύμφωνα με τις ανάγκες του κάθε διαχειριστή. Η αρχιτεκτονική του αποτελείται από πράκτορες παρακολούθησης, από συλλέκτες μετρικών και από εξυπηρετητές παρακολούθησης, όπως βλέπουμε και στο σχήμα 7. Ο κάθε πράκτορας είναι η εφαρμογή που ελέγχει τους συλλέκτες μετρικών του συγκεκριμένου στιγμιότυπου και τις στέλνει στον εξυπηρετητή. Οι συλλέκτες μετρικών παρέχουν τα δεδομένα των μετρήσεων χαμηλού επιπέδου. Οι εξυπηρετητές συλλέγουν όλες τις μετρικές σε

μια βάση δεδομένων και μέσω μιας δικτυακής υπηρεσίας τύπου RESTful[2] παρέχουν τη δυνατότητα πρόσβασης στον ενδιαφερόμενο στα δεδομένα παρακολούθησης που έχουν συλλεχθεί [3].



Σχήμα 7: Αφαιρετική όψη της αρχιτεκτονικής του JCatascopia [3].

Η μετάδοση των δεδομένων μεταξύ των πρακτόρων και των εξυπηρετητών γίνεται χωρίς κρυπτογράφηση. Σε αυτήν την επικοινωνία ελλοχεύουν οι εξής κίνδυνοι:

- ένας τρίτος κακόβουλος πράκτορας να στέλνει ψεύτικες μετρήσεις στον εξυπηρετητή.
- ένας τρίτος κακόβουλος πράκτορας να στέλνει μεγάλο όγκο μετρήσεων με σκοπό να δημιουργήσει πρόβλημα υψηλού φόρτου επεξεργασίας στον εξυπηρετητή.
- Ένας τρίτος κακόβουλος χρήστης να κλέψει τα δεδομένα των μετρήσεων με αποτέλεσμα να βγάλει κάποια συμπεράσματα όπως π.χ. για τις εφαρμογές που τρέχουν σε κάποιον εξυπηρετητή και να αποφασίσει στοχευόμενη επίθεση στη συνέχεια.

[2] https://en.wikipedia.org/wiki/Representational_state_transfer

Για τους παραπάνω λόγους είναι σημαντικό να προστεθεί ασφάλεια στη μετάδοση των δεδομένων του συστήματος παρακολούθησης JCatascopia, μεταξύ πρακτόρων και εξυπηρετητών, ώστε να αποτρέπονται οι παραπάνω επιθέσεις. Σε αυτή τη μεταπτυχιακή διατριβή α. μέσω βιβλιογραφικής έρευνας επιλέξαμε τον αλγόριθμο κρυπτογράφησης που πληροί τις προδιαγραφές του συστήματος JCatascopia, όπως είναι η ασφάλεια, η ταχύτητα, η διαφάνεια β. υλοποιήσαμε τον αλγόριθμο για τη μετάδοση των δεδομένων μεταξύ του πράκτορα και του εξυπηρετητή γ. μετρήσαμε την απόδοση της υλοποίησης σε επεξεργαστική ισχύ, σε μνήμη και σε πόρους δικτύου, σε σχέση με τους πόρους που χρειάζονται χωρίς την κρυπτογράφηση και με αυτήν. Τέλος, δείξαμε ότι έχουμε πολύ μικρή επίπτωση σε δικτυακή κίνηση, και αμελητέα επίπτωση σε υπολογιστικούς πόρους τόσο στην πλευρά του πράκτορα, που είναι και το πιο σημαντικό όσο και στην πλευρά του εξυπηρετητή.

Στο κεφάλαιο 2 γίνεται αναφορά σε αντίστοιχες ερευνητικές εργασίες στις οποίες εξετάζονται άλλα συστήματα παρακολούθησης που προσφέρουν και ασφάλεια. Στο κεφάλαιο 3 αναφέρονται συνοπτικά κάποιες βασικές έννοιες και γίνεται αναλυτική περιγραφή της λειτουργίας του συστήματος JCatascopia. Στο κεφάλαιο 4 περιγράφεται αναλυτικά η μέθοδος κρυπτογράφησης που χρησιμοποιήθηκε, ο τρόπος που υλοποιήθηκε στο JCatascopia και οι έλεγχοι λογισμικού που έγιναν. Στο 5ο κεφάλαιο περιγράφονται τα σενάρια ανάλυσης απόδοσης. Στο ίδιο κεφάλαιο υπάρχουν οι μετρήσεις για την επιβάρυνση του αλγορίθμου στην κίνηση του δικτύου και τις απαιτήσει σε πόρους τόσο στην πλευρά του πράκτορα όσο και στην πλευρά του εξυπηρετητή. Στο κεφάλαιο 6 υπάρχει η σύνοψη των συμπερασμάτων, κάποια γνωστά κενά ασφαλείας καθώς και η προτεινόμενη μελλοντική ερευνητική εργασία.

Κεφάλαιο 2
Σχετική Έρευνα

2.1 Σχετική Ερευνητική Εργασία

Παρακάτω αναφέρονται συνοπτικά άλλες ερευνητικές εργασίες, συναφή αντικειμένου με την παρούσα μεταπτυχιακή διατριβή. Σε κάθε μια από αυτές εντοπίστηκε η καινοτομία, το πλεονέκτημα αλλά και το κενό, θεωρητικό ή πρακτικό, το οποίο συνέβαλε ώστε να οδηγηθεί η συγκεκριμένη διατριβή στα μετέπειτα αποτελέσματα.

Το σύστημα CryptVMI χρησιμοποιεί εργαλεία ανάλυσης απόδοσης της αρχιτεκτονικής VMI [8], [9] προσθέτοντας ασφάλεια στη μετάδοση των δεδομένων παρακολούθησης μεταξύ των εικονικών μηχανών, εξασφαλίζοντας εμπιστευτικότητα των δεδομένων ακόμα και από τους διαχειριστές της υπολογιστικής νέφους. Σε αυτό το σύστημα ενδοσκόπησης εντοπίζεται ότι ο αλγόριθμος κρυπτογράφησης AES είναι ο πλέον γρήγορος, ασφαλής και υποστηριζόμενος από την αρχιτεκτονική επεξεργαστών της Intel [10]. Εντοπίζεται επίσης το κενό ότι το συγκεκριμένο σύστημα μπορεί να εφαρμοστεί μόνο για την ενδοσκόπηση των εικονικών μηχανών και όχι τη λήψη αποφάσεων για την αναδιάταξη των πόρων σε μια δομή υπολογιστικής νέφους. Η μη ανάπτυξή του ως λογισμικό ανοικτού και ελεύθερου κώδικα το καταστεί δυσπρόσιτο για ερευνητικούς σκοπούς [11].

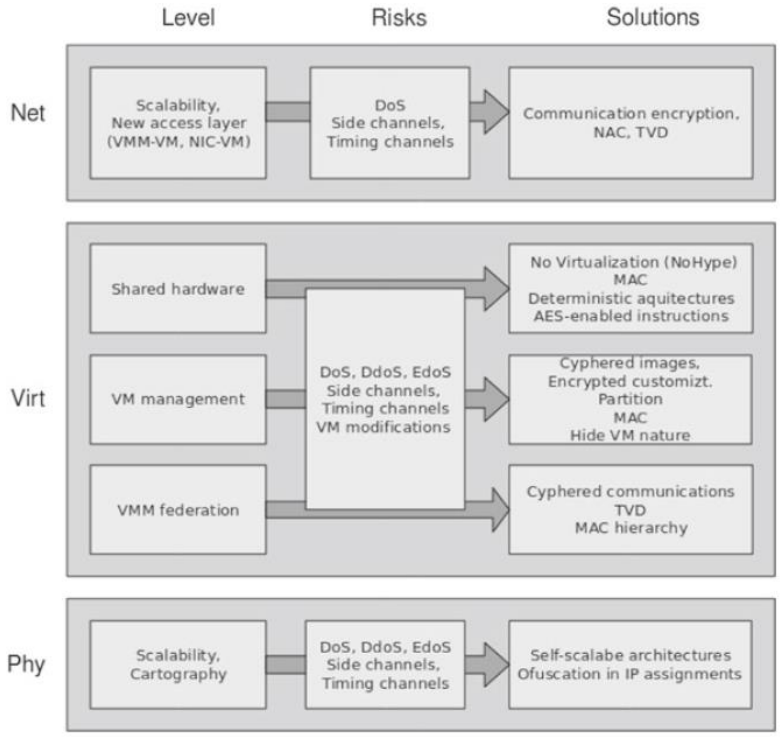
Το σύστημα DARGOS προσφέρει μια αρχιτεκτονική παρακολούθησης υψηλής προσαρμοστικότητας και επεκτασιμότητας για συστήματα υπολογιστικής νέφους πολλαπλής μίσθωσης. Κάνει χρήση της λειτουργίας «publish/subscribe» και της Υπηρεσίας Διανομής Δεδομένων «Data Distribution Service» (DDS) προσφέροντας δια-λειτουργικότητα και ποιότητα-στην-Υπηρεσία (QoS). Είναι εργαλείο ανοικτού

κώδικα βασισμένο στην πλατφόρμα OpenStack. Το συγκεκριμένο σύστημα για λόγους ασφάλειας δεν προτείνεται για χρήση και ανάγνωση των δεδομένων εκτός του δικτύου της υπολογιστικής νέφους [12].

Σύμφωνα με την «επισκόπηση των εργαλείων παρακολούθησης Υπολογιστικής Αρχιτεκτονικής Νέφους» διαπιστώνουμε ότι υπάρχει ιδιαίτερο κενό στον ασφαλή έλεγχο των συστημάτων υπολογιστικής νέφους και αυτό δημιουργεί ανασφάλεια και στους καταναλωτές [13].

Στην δημοσίευση του CloudSec βλέπουμε εύκολα το κενό που υπάρχει στον τομέα της παρακολούθησης υπολογιστικής νέφους αφού γίνεται μια υλοποίηση που αφορά καθαρά την υποδομή που βασίζεται σε λογισμικό της VMware και μόνο [14].

Σύμφωνα με τα αποτελέσματα της έρευνας του Luis Vaquero για την ασφάλεια στο μοντέλο Υποδομή ως μια Υπηρεσία (IaaS), στο σχήμα 8 βλέπουμε τους πιο πιθανούς κινδύνους που αφορούν την υπολογιστική νέφους [15].



Σχήμα 8: Σύνοψη των θεμάτων ασφαλείας στο μοντέλο IaaS [15].

Ο οργανισμός «Cloud Security Alliance» στο άρθρο [16] αναφέρει τους κινδύνους ασφαλείας της υπολογιστικής νέφους και εκπονεί ανάλυση ρίσκου για τον κάθε έναν από αυτούς. Συνοπτικά οι κίνδυνοι με σειρά κρισιμότητας είναι:

- Παραβίαση εταιρικών δεδομένων
- Απώλεια δεδομένων
- Πειρατεία λογαριασμού ή υπηρεσίας
- Μη ασφαλές σημείο διεπαφής και API
- Άρνηση υπηρεσίας
- Κακόβουλοι χρήστες εντός της υπολογιστικής νέφους
- Κατάχρηση των υπηρεσιών υπολογιστικής νέφους
- Ανεπάρκεια λόγο επιμέλειας
- Ευπάθειες διαμοιραζόμενων τεχνολογιών

Το σύστημα MonPaaS είναι μια προσαρμοστική Πλατφόρμα Παρακολούθησης ως μια Υπηρεσία (PaaS) για υποδομές και υπηρεσίες υπολογιστικής νέφους. Ο καταναλωτής της υπολογιστικής νέφους, με αυτό το σύστημα, μπορεί να ρυθμίσει και να προσαρμόσει ποιες πληροφορίες συλλέγονται για τους πόρους και τις υπηρεσίες που παρακολουθεί. Ο πάροχος υπολογιστικής νέφους και ο καταναλωτής βλέπουν διαφορετικά δεδομένα που αφορούν την παρακολούθηση [17]. Το μειονέκτημα του συστήματος είναι ότι ο διαχειριστής του MonPaaS πρέπει να έχει δικαιώματα διαχειριστή στις εικονικές μηχανές του πελάτη της υπολογιστικής νέφους σε αντίθεση με τον ίδιο τον πελάτη.

Στη δημοσίευση του Kanstren [18] γίνεται χρήση της τεχνολογίας Trusted Platform Modules (TPM) και vTPM [19], [20], [21] η οποία προσφέρει ασφάλεια στη μετάδοση των μετρικών, βασισμένη στο υλικό της φυσικής μηχανής και στο λογισμικό εικονικοποίησης των εικονικών μηχανών. Προϋπόθεση και περιορισμός είναι ότι το τελευταίο θα πρέπει να υποστηρίζει τις λειτουργίες vTPM και αντίστοιχα το υλικό της μηχανής που φιλοξενεί τις εικονικές μηχανές θα πρέπει να ενσωματώνει τη λειτουργία TPM. Αυτή η αρχιτεκτονική επηρεάζει την ταχύτητα μετάδοσης των δεδομένων ελάχιστα και την καθιστά αρκετά ασφαλή αλλά όχι ανεξάρτητη λειτουργικού συστήματος και υλικού.

Σε βιβλιογραφική έρευνα που έγινε για τη σύγκριση των αλγορίθμων AES (Advanced Encryption Standard), DES (Data Encryption Standard) και 3DES σύμφωνα με τα άρθρα [10], [22] και [23] το συμπέρασμα από τις μετρήσεις είναι ότι ο AES επιβαρύνει πολύ λιγότερο την απόδοση του συστήματος σε σχέση με τους άλλους δύο αλγόριθμους. Ο Najib A. [24] μελετάει ποιος από τους αλγόριθμους κρυπτογράφησης AES, 3DES, DES, Blowfish είναι πιο γρήγορος όταν υλοποιείται με τις βιβλιοθήκες JCE της Java και καταλήγει ότι ο Blowfish είναι ο γρηγορότερος και ο 3DES είναι ο ασφαλέστερος αλλά και ο πιο αργός. Από τις μετρήσεις που παρουσιάζονται, αλλά και σύμφωνα με την ιστορία κατά την οποία ο AES επιλέχθηκε από τον NIST ανάμεσα στους προτεινόμενους αλγόριθμους, καταλήξαμε να επιλέξουμε τον AES για την υλοποίησή μας.

Στο [25] ο Bogdanov συγκρίνοντας τις γνωστές μεθόδους κρυπτανάλυσης του AES αναπτύσσει μια νέα με την οποία αποδεικνύει για πρώτη φορά ότι το κλειδί του αλγόριθμου κρυπτογράφησης AES μπορεί να βρεθεί. Πρακτικά όμως είναι αδύνατον αφού ο χρόνος και η υπολογιστική ισχύς που απαιτείται για κάτι τέτοιο είναι υπερβολικά μεγάλος λόγο του μεγάλου αριθμού των επαναλήψεων που απαιτούνται.

Κεφάλαιο 3

Βασικές Έννοιες - JCatascopia

Οι παρακάτω έννοιες είναι χρήσιμο να αναφερθούνε, ώστε να γίνει κατανοητή η ανάλυση της λειτουργίας και της ασφαλούς μετάδοσης των δεδομένων του συστήματος JCatascopia. Στη συνέχεια αναλύεται η λειτουργία του συστήματος παρακολούθησης JCatascopia.

3.1 Βασικές έννοιες εργαλείων και δομών

3.1.1 Τι είναι η Java;

Είναι μια γλώσσα αντικειμενοστραφής προγραμματισμού. Ο κώδικας μεταγλωττίζεται σε κώδικα εικονικής μηχανής ή bytecode. Αυτό κάνει την πλατφόρμα της Java ανεξάρτητη λειτουργικού συστήματος [26].

3.1.2 Τι είναι μια εικονική μηχανή (Virtual Machine);

Εικονική μηχανή είναι μια εφαρμογή η οποία προσομοιώνει έναν Η/Υ κάνοντας χρήση των πόρων του φυσικού Η/Υ. Ο χρήστης έχει τη δυνατότητα αυτήν την εικονική μηχανή να την χρησιμοποιεί ως ένα τελείως ξεχωριστό σύστημα από το φυσικό. Μια εικονική μηχανή δίνει τη δυνατότητα στον τελικό χρήστη να αξιοποιήσει και να κάνει καλύτερη χρήση των φυσικών πόρων του Η/Υ [27].

3.1.3 Τι είναι ένα «Web Service»;

Ένα Web Service είναι ένα σύστημα λογισμικού που έχει σχεδιαστεί για την δια-λειτουργική επικοινωνία μεταξύ μηχανών μέσω του δικτύου. Τα συστήματα που χρησιμοποιούν ένα Web Service για την επικοινωνία τους μεταφέρουν τα δεδομένα μέσω HTTP[3] σε XML[4], συνδυάζοντας και άλλα πρότυπα [28].

3.1.4 Τι είναι μια Βάση Δεδομένων;

Μια Βάση Δεδομένων ενός Οργανισμού ή μίας Επιχείρησης είναι μια συλλογή αλληλοσχετιζόμενων, διαμοιραζόμενων (shared) λειτουργικών στοιχείων και μόνιμων δεδομένων που αποθηκεύονται μαζί (για μεγάλο χρονικό διάστημα) χωρίς άχρηστους πλεονασμούς για την ταυτόχρονη εξυπηρέτηση πολλών εφαρμογών [29].

3.2 Περιγραφή λειτουργίας του συστήματος JCatascopia

Το JCatascopia είναι ένα Σύστημα Παρακολούθησης Υπολογιστικής Νέφους πολλαπλών επιπέδων παρέχοντας δια-λειτουργικότητα. Είναι λογισμικό ανοικτού κώδικα και μπορεί να τρέξει με τρόπο διαφανή και μη οχληρό ως προς οποιαδήποτε υποδομή εικονικού περιβάλλοντος. Κατά την εκτέλεσή του, ανιχνεύει δυναμικά, χωρίς ανθρώπινη παρέμβαση ή την ανάγκη επανεκκίνησης του συστήματος παρακολούθησης οποιαδήποτε προσθήκη/αφαίρεση ενός παρακολουθούμενου στιγμιότυπου. Προσαρμόζεται κατά την εφαρμογή, καθιστώντας λιγότερο αυστηρή την ανάγκη για ανα-ταξινόμηση κάθε φορά που μια εφαρμογή ή/και μια παράμετρος των συσχετιζόμενων πόρων αλλάζει. Δημιουργεί δυναμικά υψηλού επιπέδου μετρικές κατά την εκτέλεση, συγκεντρώνοντας και ομαδοποιώντας μετρικές χαμηλού επιπέδου. Παρέχει δυνατότητες φιλτραρίσματος για να μειώσει

[3] https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
[4] https://en.wikipedia.org/wiki/XML

την επιφόρτιση στην κίνηση του δικτύου για την αποθήκευση και κατανομή των μετρικών [3].

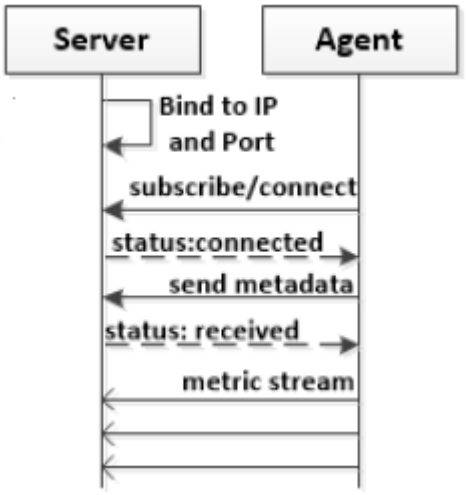
3.2.1 JCatascopia Συλλέκτης Μετρικών

Στο JCatascopia ένας συλλέκτης μετρικών α. μπορεί να επερωτά και να αποθηκεύει την τελευταία μέτρηση και την ακριβή ώρα της μέτρησης, β. επιτρέπει στον προγραμματιστή να προσθέσει τη δική του συλλογή μετρήσεων όπως την καταγραφή της χρήσης της μνήμης μόνο όταν εκτελείται μια συγκεκριμένη εφαρμογή, γ. έχει παραμέτρους όπως η χρονική περίοδος για την οποία θέλουμε να παίρνουμε μετρήσεις, δ. παρέχει φιλτράρισμα, ε. μπορεί να υλοποιηθεί δυναμικά σε έναν πράκτορα παρακολούθησης ενώ αυτός εκτελείται, ζ. διανέμει τις μετρικές με χρονική διάκριση, αρχικοποιώντας είτε έναν μηχανισμό παράδοσης ώθησης είτε έλξης. [3]

Οι χρήστες έχουν τη δυνατότητα να ρυθμίσουν τις παραμέτρους φιλτραρίσματος όπως είναι το N, το A, το minR, το maxR και το step. Με αυτές τις παραμέτρους και τη λειτουργία προσαρμοστικού φιλτραρίσματος αποφεύγεται η μετάδοση και η αποθήκευση περιττής πληροφορίας κάνοντας μεγάλη εξοικονόμηση σε πόρους. [3]

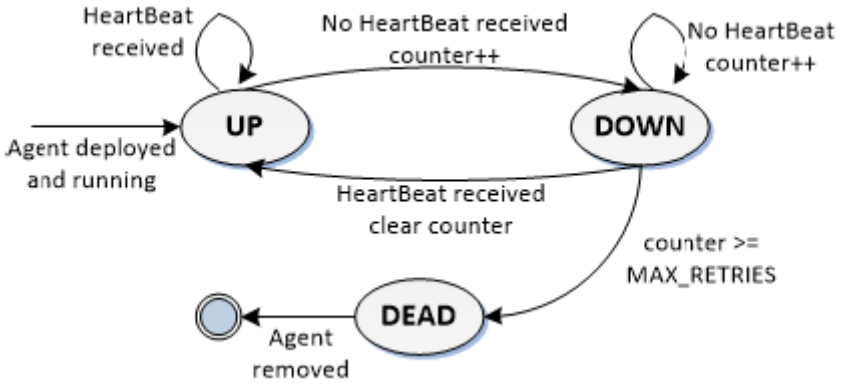
3.2.2 JCatascopia Πράκτορας Παρακολούθησης

Στο σχήμα 9 φαίνεται ο μηχανισμός με τον οποίο γίνεται η αρχική επικοινωνία ενός Πράκτορα παρακολούθησης με τον Εξυπηρετητή παρακολούθησης ώστε να λαμβάνονται οι μετρικές και να καταγράφονται στη Β.Δ. του εξυπηρετητή. [3]



Σχήμα 9: JCatascopia μηχανισμός pub/sub. [3]

Ο εξυπηρετητής ανά κάποιο χρονικό διάστημα δέχεται ένα σήμα από τον πράκτορα το οποίο δηλώνει ότι είναι ενεργός και υπάρχει επικοινωνία μεταξύ τους. Εάν δεν επικοινωνήσει ο πράκτορας για κάποιο χρονικό διάστημα με τον εξυπηρετητή τότε θεωρείται «νεκρός» και ο εξυπηρετητής τον αποσυνδέει. Η λειτουργία αυτή φαίνεται στο σχήμα 10. [3]



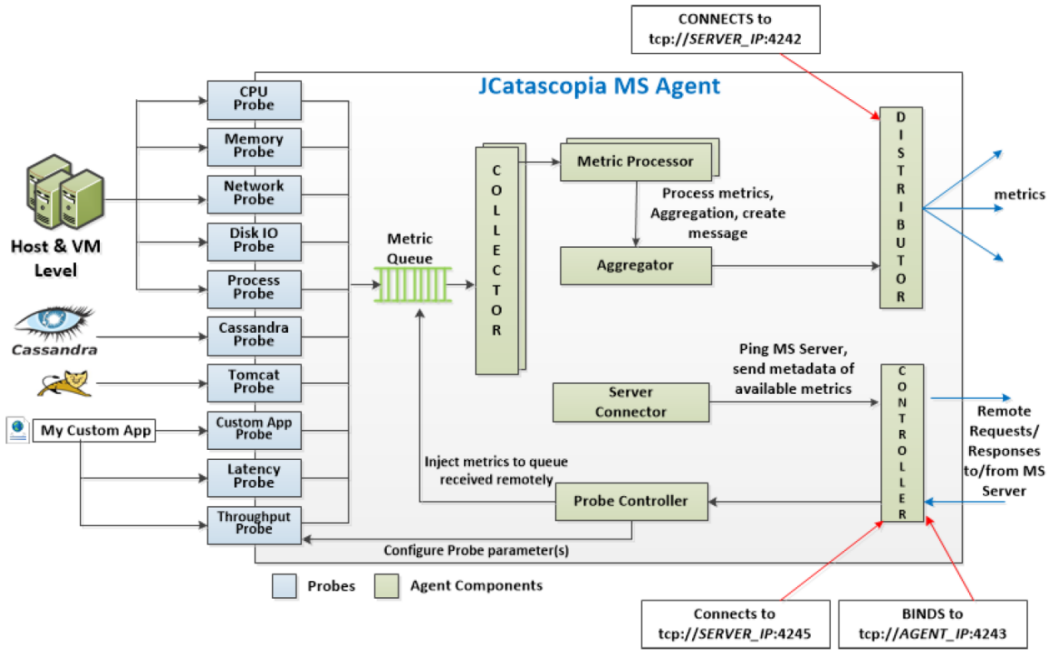
Σχήμα 10: Διάγραμμα κατάστασης πράκτορα παρακολούθησης. [3]

Ο πράκτορας παρακολούθησης αρχικοποιεί τους συλλέκτες μετρικών, επεξεργάζεται τις μετρήσεις, τις μετατρέπει ώστε να είναι αναγνώσιμες από τον εξυπηρετητή, συνδέεται στον εξυπηρετητή, στέλνει τις μετρήσεις που έχει συλλέξει, αποσυνδέει τους συλλέκτες και τελικά τερματίζει την επικοινωνία με τον εξυπηρετητή.

Ο πράκτορας για να συνδεθεί με τον εξυπηρετητή αρχικά εκτελεί τη λειτουργία ping προς αυτόν και εφόσον εκείνος απαντήσει τότε του στέλνει το ID του, την IP

διεύθυνση και τα μετα-δεδομένα τα οποία δηλώνουν τις μετρικές που θα συλλέγονται, και τον τύπο τους. Εφόσον ολοκληρωθεί αυτή η διαδικασία δημιουργείται ένα συνεχές ρεύμα μετρήσεων προς τον εξυπηρετητή. Ο μηχανισμός διανομής pub/sub υλοποιείται με το πλαίσιο ZMQ το οποίο είναι δομημένο πάνω στην επικοινωνία τύπου ZMQ socket. Λόγω του πλήρους ελέγχου της επικοινωνίας, σε τόσο χαμηλό επίπεδο, το JCatascopia ελέγχει πλήρως την επιτυχή αποστολή των μετρήσεων ώστε να μην χάνονται πακέτα αλλά σε περίπτωση αποτυχίας να αποστέλλονται πάλι στον εξυπηρετητή.

Έχει υλοποιηθεί ένα RESTful API μέσω του οποίου οι χρήστες ή ο εξυπηρετητής στέλνουν τις παραμέτρους της συλλογής μετρήσεων στον ελεγκτή του συλλέκτη. Οι μετρικές μετατρέπονται από τον επεξεργαστή μετρικών σε αναγνώσιμη μορφή και στη συνέχεια προστίθενται τα μεταδεδομένα. Ο αριθμός των ελεγκτών και των επεξεργαστών είναι προσαρμόσιμος μέσω ενός αρχείου που βρίσκεται στον φάκελο εγκατάστασης. Ο ελεγκτής συσσωμάτωσης κρατάει τις μετρικές μέχρι να ικανοποιηθούν κάποιες συνθήκες συσσωμάτωσης οι οποίες είναι προσαρμόσιμες μέσω του αρχείου που προαναφέραμε. Για παράδειγμα μια τέτοια συνθήκη είναι ότι το μέγεθος του μηνύματος μετρικών δεν πρέπει να ξεπερνάει τα 2KB. Στο σχήμα 11 βλέπουμε την αρχιτεκτονική του πράκτορα παρακολούθησης. [3]

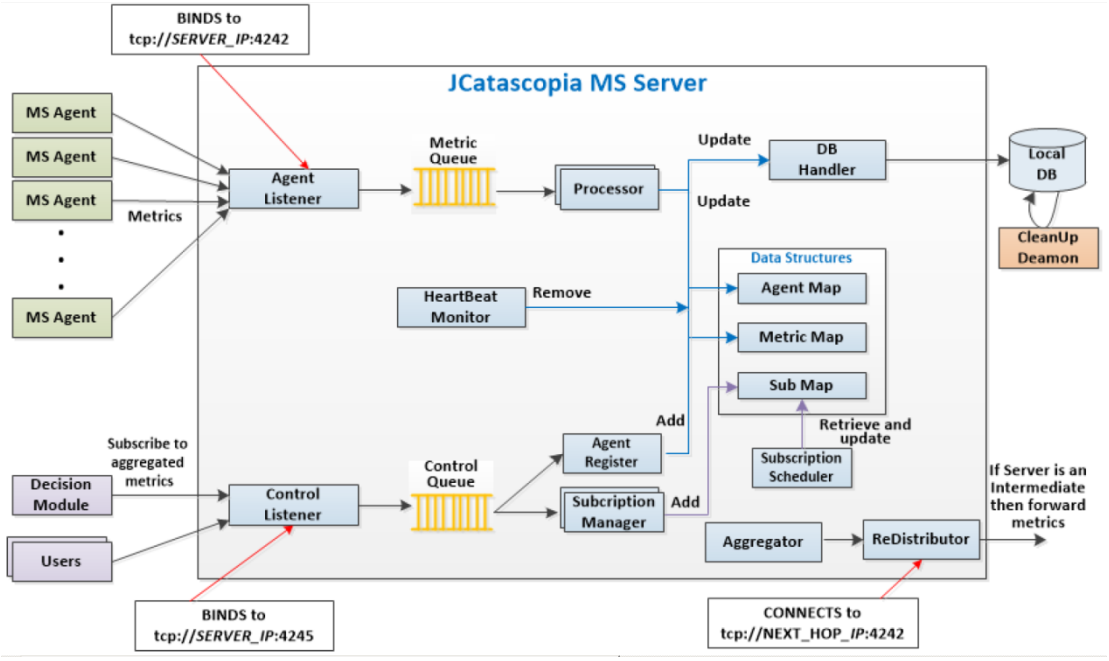


Σχήμα 11. Αρχιτεκτονική JCatascopia MS Agent [3].

3.2.3 JCatascopia Εξυπηρετητής Παρακολούθησης

Ο εξυπηρετητής παρακολούθησης διευθύνει τους πράκτορες και αποθηκεύει τις μετρικές στην Β.Δ.. Ο Control Listener αναμένει για αιτήσεις σύνδεσης από τους πράκτορες και ο Agent Listener αναμένει μετρικές που προστέθηκαν μετά την αρχική σύνδεση. Όταν τα μηνύματα με τις μετρικές φθάνουν στον εξυπηρετητή, τοποθετούνται σε ουρά. Οι επεξεργαστές μετρικών παίρνουν ένα-ένα τα μηνύματα από την ουρά, τα επεξεργάζονται και οι ελεγκτές της Β.Δ. τα αποθηκεύουν σε αυτήν.. Ο αριθμός των επεξεργαστών μετρικών είναι προσαρμόσιμος μέσω του αρχείου παραμετροποίησης στον φάκελο εγκατάστασης.

Μέσω του RESTful API ο μηχανισμός εγγραφής επιτρέπει στους χρήστες αλλά και σε εσωτερικές οντότητες να εφαρμόσουν φίλτρα συσσωμάτωσης και ομαδοποίησης στις μετρικές χαμηλού επιπέδου, ώστε να δημιουργήσουν νέες μετρικές υψηλού επιπέδου. Ο διαχειριστής εγγραφής ανακτά και επεξεργάζεται τις αιτήσεις σύνδεσης από την ουρά ελέγχου, ενημερώνοντας την Β.Δ. και τον Subscription Scheduler. Ο τελευταίος ανακτά τις συνδέσεις από τη Β.Δ. και ενημερώνει την τρέχουσα κατάσταση βασιζόμενος στην ελάχιστη περίοδο ενημέρωσης που έχει επιλεχθεί από τον χρήστη κατά την αίτηση σύνδεσης. Στο σχήμα 12 φαίνεται η δομή της αρχιτεκτονικής του εξυπηρετητή παρακολούθησης. [3]



Σχήμα 12. Αρχιτεκτονική JCatascopia MS Server [3].

3.2.4 JCatascopia Β.Δ.

Η Β.Δ. δεν είναι κλειστή στο σύστημα JCatascopia επιτρέποντας τον διαχειριστή να επιλέξει τον τύπο που τον βολεύει. Το σύστημα έχει ως αρχικές ρυθμίσεις τη σύνδεση με MySQL[5]. Για να μην αυξάνεται συνεχώς ο χρόνος απόκρισης ερωτημάτων SQL, έχει υλοποιηθεί μια διεργασία καθαρισμού η οποία διαγράφει παλιές εγγραφές, επεξεργάζεται τα δεδομένα φιλτράροντας τιμές που δεν χρειάζονται και εκτελώντας λειτουργίες συσσωμάτωσης σε μεγαλύτερα χρονικά διαστήματα. [3]

[5] https://www.mysql.com/

Κεφάλαιο 4
Σχεδιασμός - Υλοποίηση

Στο σύστημα JCatascopia είναι ανάγκη να προστεθεί εμπιστευτικότητα και ακεραιότητα δεδομένων κατά την μετάδοση των δεδομένων μεταξύ του εξυπηρετητή και του πράκτορα. Αυτό επιτυγχάνεται με κρυπτογράφηση των δεδομένων πριν την αποστολή τους από τον πράκτορα προς τον εξυπηρετητή. Χρειάζεται προσεκτική επιλογή του αλγορίθμου υλοποίησης της κρυπτογράφησης καθώς πρέπει το σύστημα να παραμείνει διαφανής ως προς την εκτέλεσή του, γρήγορο και ελαστικό. Η επιλογή του αλγορίθμου όσον αφορά την ασφάλεια και την ταχύτητα έγινε βασιζόμενη στην βιβλιογραφία και σε μετρήσεις άλλων ερευνητών καθώς είναι αδύνατον να μελετηθούν όλοι οι αλγόριθμοι κρυπτογράφησης. Οι βιβλιοθήκες που επιλέχτηκαν για την υλοποίηση βασίστηκαν στο δεδομένο του ανοικτού κώδικα και κυρίως στην συνεχή ανανέωσή τους. Οι ευπάθειες που εμφανίζονται καθημερινά στις βιβλιοθήκες αλλά και γενικότερα σε αλγορίθμους ασφάλειας είναι ανάγκη να υποστηρίζονται και να διορθώνονται από μια ομάδα ανάπτυξης ή οργανισμό που λαμβάνει υπόψη τα διεθνή πρότυπα και αυτό συμβαίνει και στην περίπτωση της Java από την Oracle.

Η κρυπτογράφηση στον AES λαμβάνει χώρα, όπως και σε κάθε κρυπταλγόριθμο τμήματος, μέσα από ένα πλήθος γύρων. Το τμήμα του αρχικού μηνύματος, το οποίο αποτελεί την είσοδο στον αλγόριθμο, θεωρείται ως πίνακας τεσσάρων γραμμών, αποτελούμενος από byte. Οι τέσσερις γραμμές αριθμούνται κατά σειρά 0, 1, 2, 3. Το πλήθος των στηλών του εν λόγω πίνακα καθορίζεται από το μέγεθος του τμήματος του μηνύματος. Στον AES, όπου το τμήμα του μηνύματος έχει μέγεθος 128 bit, το πλήθος των στηλών ενός τέτοιου πίνακα είναι 4 (αφού 4 X 4 X 8 = 128). Γενικότερα, το πλήθος των στηλών του πίνακα του μηνύματος συμβολίζεται με N_b. Ίδια θεώρηση ισχύει για κάθε ενδιάμεσο τμήμα που προκύπτει κατά τη διαδικασία της κρυπτογράφησης, καθώς και για το τελικό κρυπτογραφημένο τμήμα: όλα τα

τμήματα λογίζονται ως τετραγωνικοί πίνακες από byte, διάστασης 4. Κάθε τέτοιος πίνακας ονομάζεται κατάσταση (state). Αντίστοιχα, και το κλειδί αναπαρίσταται σαν πίνακας 4 γραμμών από byte, όπου το πλήθος των στηλών του N_k εξαρτάται από το μέγεθος του κλειδιού: για μεγέθη 128, 192 και 256 bit, ισχύει αντίστοιχα N_k = 4, 6, 8.

Η δομή που περιγράφηκε παραπάνω για το τμήμα των δεδομένων και το κλειδί αποτυπώνεται στους Πίνακες 9 και 10 αντίστοιχα, για $N_b = N_k = 4$.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Πίνακας 9. Πίνακας κατάστασης

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Πίνακας 10. Πίνακας κλειδιού

Τα byte των πινάκων κατάστασης και κλειδιού συμπληρώνονται κατά στήλη. Αυτό σημαίνει ότι τα πρώτα 8 bit του μηνύματος καταλαμβάνουν το byte $\alpha_{0,0}$ του αρχικού πίνακα κατάστασης, τα επόμενα 8 bit καταλαμβάνουν το byte $\alpha_{1,0}$ κ.ο.κ., ενώ αντίστοιχα συμπληρώνεται και ο πίνακας του κλειδιού. Κατά ανάλογο τρόπο, τα πρώτα 8 bit του κρυπτογραφημένου τμήματος λαμβάνονται από το byte $\alpha_{0,0}$ του τελικού πίνακα κατάστασης, τα επόμενα 8 bit από το byte $\alpha_{1,0}$ κ.ο.κ.

Ο αλγόριθμος Rijndael είναι εκείνος που επιλέχτηκε από τον οργανισμό NIST το 2001 ως ο AES. Οι διαφορές μεταξύ του Rijndael και του AES είναι α. ο Rijndael επιτρέπει το μέγεθος του κλειδιού να είναι πολλαπλάσιο του 32 και μεγαλύτερο των 128 bits και δεν απαιτείται να είναι ίδιο μεταξύ τους β. στον AES το μέγεθος του μπλοκ πρέπει να είναι 128 bits και το μέγεθος του κλειδιού πρέπει να είναι μεταξύ 128, 192 και 256 bits. Το πλήθος Nr των γύρων στον Rijndael εξαρτάται τόσο από το N_k όσο και από το N_b. Οι ακριβείς τιμές αναγράφονται στον Πίνακα 11.

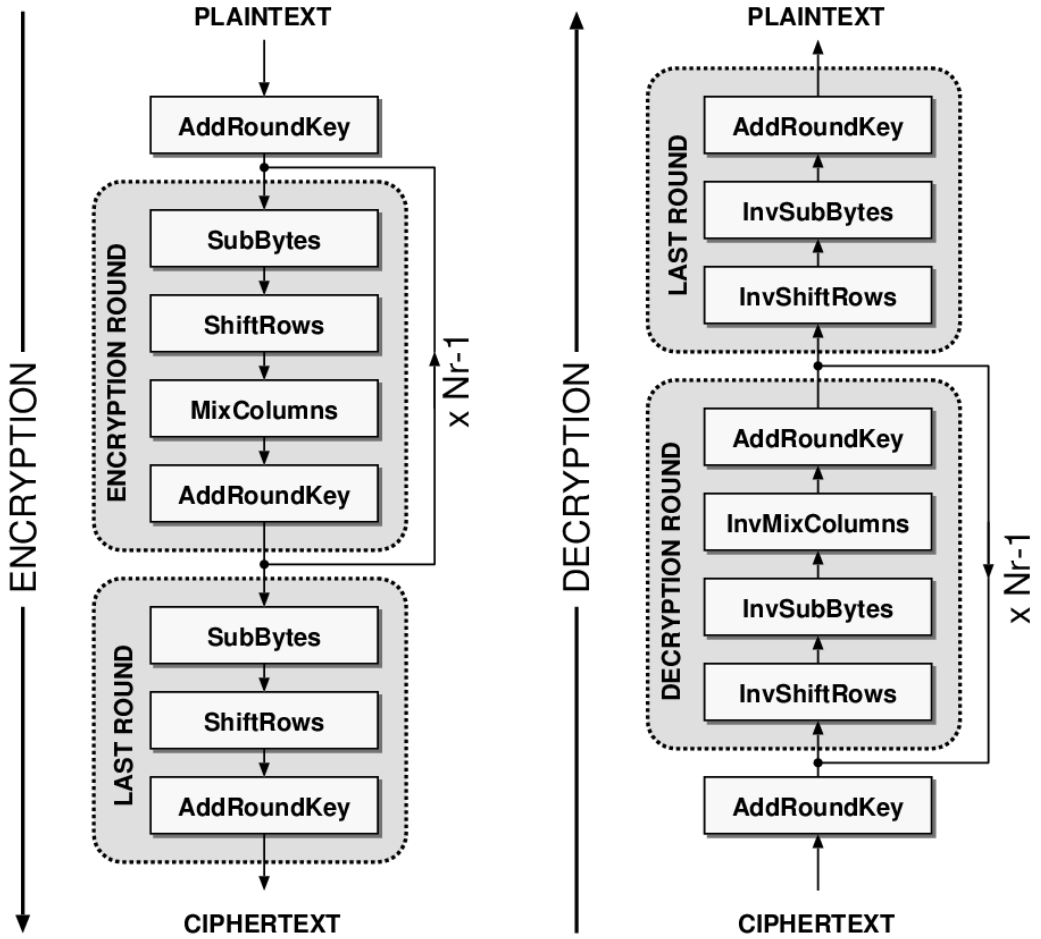
N_r	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	10	12	14
$N_k = 6$	12	12	14
$N_k = 8$	14	14	14

Πίνακας 11. Πλήθος γύρων στον Rijndael

Σε κάθε γύρο συντελούνται τέσσερις διαδικασίες με την εξής σειρά:

- Αντικατάσταση byte (SubBytes),
- Ολίσθηση γραμμών (ShiftRows),
- Ανάμειξη στηλών (MixColumns),
- Πρόσθεση κλειδιού για τον τρέχοντα γύρο (AddRoundKey)

όπου η έξοδος κάθε μίας από τις διαδικασίες αυτής αποτελεί είσοδο για την επόμενη. Μία εξαίρεση αποτελεί ο τελευταίος γύρος, στον οποίο δε συντελείται η διαδικασία MixColumns. Επίσης, πριν τον πρώτο γύρο γίνεται XOR πρόσθεση του τμήματος του μηνύματος με το κλειδί K. Η όλη διαδικασία της κρυπτογράφησης απεικονίζεται στο αριστερό τμήμα του Σχήματος 6, για την περίπτωση $N_r = 10$. Γίνεται φανερό ότι ο AES δεν είναι δομής Feistel, όπως ο DES. Η υλοποίηση του αλγόριθμου Feistel λειτουργεί στην κρυπτογράφηση όπως και στην αποκρυπτογράφηση, κάνοντας αντιστροφή της χρήσης του κλειδιού. Δηλαδή στην κρυπτογράφηση θα χρησιμοποιηθεί το K0 έως το Kn ενώ στην αποκρυπτογράφηση το Kn έως το K0. Αξίζει επίσης να σημειωθεί ότι στις πρώτες τρεις διαδικασίες κάθε γύρου (SubBytes, ShiftRows, MixColumns) δεν υπεισέρχεται καθόλου το κλειδί [30].



Σχήμα 13. Βασική δομή του αλγόριθμου AES: αριστερά κρυπτογράφηση και δεξιά αποκρυπτογράφηση. [31]

4.1 Η Βιβλιοθήκη προσθήκης κρυπτογραφίας στην Java, JCE.

Για την υλοποίηση του αλγορίθμου AES χρησιμοποιήθηκε η βιβλιοθήκη JCE της Sun που είναι ενσωματωμένη πλέον στην Java από την έκδοση JDK 1.4 και έπειτα [32].

4.2 Υλοποίηση και εφαρμογή κρυπτογραφίας AES στο σύστημα JCatascopia

Η ανάπτυξη του JCatascopia έχει γίνει στο περιβάλλον ανάπτυξης Eclipse, οπότε συνεχίστηκε στο ίδιο περιβάλλον και η προσθήκη του module ασφάλειας μετάδοσης δεδομένων.

Επιλέχθηκε να γίνει κρυπτογράφηση σε mode CBC/PKCS7PADDING [36]. Η παράμετρος PKCS7PADDING δηλώνει ότι μπορεί να γίνει κρυπτογράφηση και αποκρυπτογράφηση σε blocks δεδομένων από 1 έως 255 bytes. Η παράμετρος CBC δηλώνει την αλυσιδωτή κρυπτογράφηση, αφού μέρος του κρυπτογραφημένου μηνύματος χρησιμοποιείται για την κρυπτογράφηση του επόμενου block μηνύματος.

Και στον Agent και στον Server χρειάστηκε να συμπεριληφθούν κάποιες βιβλιοθήκες στην αρχή των κλάσεων οι οποίες δίνουν τη δυνατότητα χρήσης των εντολών κρυπτογράφησης και αποκρυπτογράφησης της JCE. Συνοπτικά είναι οι εξής:

- javax.crypto.*.
- java.security.*.
- sun.misc.BASE64Encoder
- org.apache.commons.codec.binary.Base64

Στον JCatascopia Agent, η κλάση TCPDistributor είναι υπεύθυνη για την αποστολή του τελικού μηνύματος στον Server. Σε αυτήν την κλάση ακριβώς πριν την αποστολή του μηνύματος προστέθηκε η μέθοδος encrypta η οποία κρυπτογραφεί το απλό κείμενο και το μετατρέπει σε κρυπτοκείμενο, δηλαδή σε μη αναγνώσιμη μορφή. Ως παραμέτρους παίρνει το κλειδί και το plaintext και ως έξοδο έχει το ciphertext.

Στον JCatascopia Server, υπάρχει η κλάση MetricProcessor η οποία δέχεται τα μηνύματα από τον Agent για περαιτέρω επεξεργασία και αποθήκευση. Σε αυτήν την κλάση αναπτύχθηκε η μέθοδος decrypta ή οποία εκτελεί την αποκρυπτογράφηση των μηνυμάτων. Αν τα μηνύματα δεν είναι κρυπτογραφημένα με το σωστό κλειδί

τότε απορρίπτονται από τον Server. Παράμετροι της μεθόδου αποκρυπτογράφησης είναι το κλειδί των 128 bit και το κρυπτοκείμενο, ενώ έξοδός της είναι το μήνυμα σε αναγνώσιμη μορφή, το οποίο και αποθηκεύει στη Β.Δ.

Στο σενάριο που ακολουθήθηκε στην υλοποίηση αυτή, θεωρείται ότι ο Server και ο Agent έχουν σαν παράμετρο, εξαρχής το ίδιο κλειδί κρυπτογράφησης των 128 bit.

Φυσικά και στον Server και στον Agent υπάρχουν αντίστοιχα οι κλάσεις εξαιρέσεων, ώστε σε περίπτωση προβλήματος να μην σταματάει η λειτουργία της εφαρμογής, αλλά να συνεχίζει απρόσκοπτα, δίνοντας τη δυνατότητα για ιχνηλασιμότητα στο πρόβλημα που προέκυψε.

4.3 Έλεγχοι λογισμικού ανοικτού και κλειστού κουτιού

Για τον έλεγχο της λειτουργίας της κρυπτογράφησης έγιναν έλεγχοι ανοικτού κουτιού. Ο κώδικας ελέγχτηκε γραμμή προς γραμμή και όπου εντοπίστηκαν λάθη και παραλήψεις διορθώθηκαν. Έλεγχοι κλειστού κουτιού έγιναν μέσω της βάσης δεδομένων του JCatascopia Server στην οποία είδαμε επανειλημμένως τις μετρικές του Agent αποκρυπτογραφημένες.

4.3.1 Έλεγχοι ανοικτού κουτιού

Για να γίνει διεξοδικός έλεγχος του κώδικα, κατά το debugging, προστέθηκαν συναρτήσεις εκτύπωσης κάθε μηνύματος πριν την κρυπτογράφηση και μετά την αποκρυπτογράφηση. Το κάθε μήνυμα εκτυπώθηκε στην οθόνη πριν να κρυπτογραφηθεί και αμέσως μετά την αποκρυπτογράφησή του από τον εξυπηρετητή, ώστε να είναι εύκολος ο εντοπισμός λαθών ή παραλείψεων. Χρησιμοποιήθηκε μόνο ένας Server και ένας Agent οι οποίοι μάλιστα εκτελέστηκαν στο ίδιο σύστημα για λόγους απλότητας. Αφού λύθηκαν τα πρωτογενή προβλήματα στον κώδικα, εκτελέστηκε η ίδια διαδικασία με τον Server σε διαφορετικό σύστημα από τον Agent.

Μια ακόμα περίπτωση που ελέγχτηκε είναι η αντίδραση του συστήματος όταν δεν είναι σωστό το κλειδί μεταξύ του Server και του Agent. Παρατηρήθηκε ότι ο Server λαμβάνει τα μηνύματα αλλά τα απορρίπτει απ' ευθείας.

4.3.2 Έλεγχοι κλειστού κουτιού

Σε αυτήν την περίπτωση επιλέχθηκε πάλι μόνο ένας Server και ένας Agent οι οποίοι εκτελούνται σε διαφορετικά συστήματα. Ο έλεγχος έγινε πάνω στις εγγραφές της Β.Δ. όπου ελέγχτηκε η ορθή καταγραφή των μετρικών που συλλέγονταν. Στη συνέχεια προστέθηκαν περισσότεροι Agents ώστε να ελεγχθεί ή ομαλή κρυπτογράφηση/αποκρυπτογράφηση με πολλούς Agents συνδεδεμένους στον Server.

Κεφάλαιο 5
Εκτίμηση

Η εκτίμηση της υλοποίησης χρειάζεται να γίνει σε πραγματικό περιβάλλον υπολογιστικής νέφους που είναι πλήρως ελεγχόμενο από τον διαχειριστή ώστε να μπορεί απομονωθεί η μέτρηση της απόδοσης και της ασφάλειας του συστήματος JCatascopia χωρίς την παρεμβολή αγνώστων παραμέτρων. Αυτό μπορεί να επιτευχθεί στο ιδιωτικό μοντέλο ανάπτυξης υπολογιστικής νέφους όπως και έγινε στην παρούσα διατριβή.

5.1 Υποδομή για τις πειραματικές μετρήσεις

Η υποδομή για τις μετρήσεις αποτελείται από έναν Η/Υ με Win 7 Pro 64Bit, Intel Core i5-3750K 3.4GHz, RAM 8GB, HDD 1TB. Για το ρόλο του Hypervisor[6] χρησιμοποιήθηκε το πακέτο ESXi V6. Το λειτουργικό σύστημα των εικονικών μηχανών είναι το Ubuntu 14.04 64 Bit.

Σε μια εικονική μηχανή έγινε εγκατάσταση και παραμετροποίηση του JCatascopia Server σύμφωνα με τις οδηγίες που βρίσκονται στο link https://github.com/CELAR/cloud-ms/tree/master/JCatascopia-Server. Σε άλλες πέντε εικονικές μηχανές, έγινε εγκατάσταση του JCatascopia Agent και παραμετροποίηση ώστε να βλέπει τον JCatascopia Server (). Ανάλογα με το πείραμα και τις μετρήσεις που θέλαμε να συλλέξουμε, γινόταν εκτέλεση του JCatascopia Server και Agent με κρυπτογράφηση ή όχι αντίστοιχα.

[6] https://www.vmware.com/products/vsphere-hypervisor

5.2 Σενάρια ανάλυσης απόδοσης

Τα ερωτήματα που πρέπει να απαντηθούν από αυτό το κεφάλαιο για την κρυπτογράφηση στο σύστημα JCatascopia κατά την μετάδοση των δεδομένων μεταξύ του εξυπηρετητή και του πράκτορα είναι:

- αν η κρυπτογράφηση είναι ασφαλής και τα δεδομένα είναι πραγματικά μη αναγνώσιμα από κάποιον που δεν γνωρίζει το κλειδί.
- Η επιφόρτιση του δικτύου τόσο στην πλευρά του εξυπηρετητή όσο και στην πλευρά του πράκτορα είναι αρκετά σημαντική σε βαθμό που να επηρεάζει την κίνηση του δικτύου όταν εισάγεται η κρυπτογράφηση;
- Η κρυπτογράφηση των δεδομένων του συστήματος JCatascopia απαιτεί τόσους υπολογιστικούς πόρους στην πλευρά του πράκτορα που να την καθιστά πολύ ακριβή;
- Οι απαιτήσεις σε υπολογιστικούς πόρους στην πλευρά του εξυπηρετητή είναι τόσο μεγάλες για ένα πραγματικό σύστημα με πολλούς πράκτορες ώστε να συνεπάγεται και αυξημένο κόστος υλοποίησης;

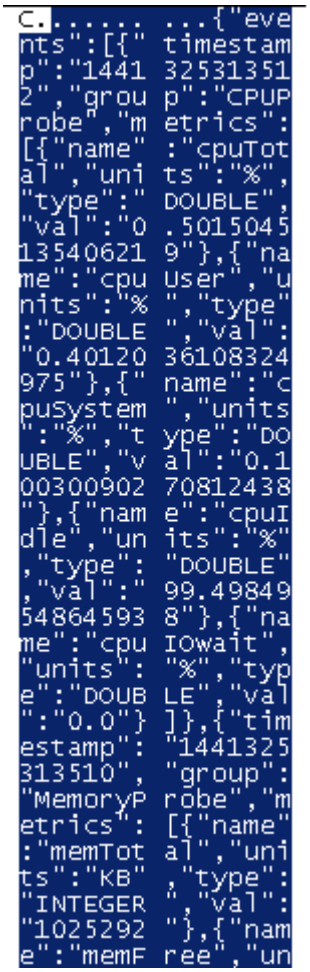
5.3 Ανάλυση μη κρυπτογραφημένων και κρυπτογραφημένων πακέτων

Το λογισμικό που χρησιμοποιήθηκε για την λήψη των πακέτων είναι το tcpdump του Linux και για την ανάλυσή τους το Wireshark [34]. Το Wireshark είναι ένα ευρέως χρησιμοποιούμενο εργαλείο για την ανάλυση της κίνησης του δικτύου καθώς είναι πολύ εύχρηστο, παραθυρικό, αλλά και με πολύ αυξημένες δυνατότητες τόσο σε χαμηλό επίπεδο όσο και σε γραφήματα. Το tcpdump είναι το καθιερωμένο εργαλείο ανάλυσης της κίνησης του δικτύου και δουλεύει σε γραμμή εντολών σε συστήματα Linux. Καταγράφει τα ζητούμενα δεδομένα σε αρχεία τύπου txt.

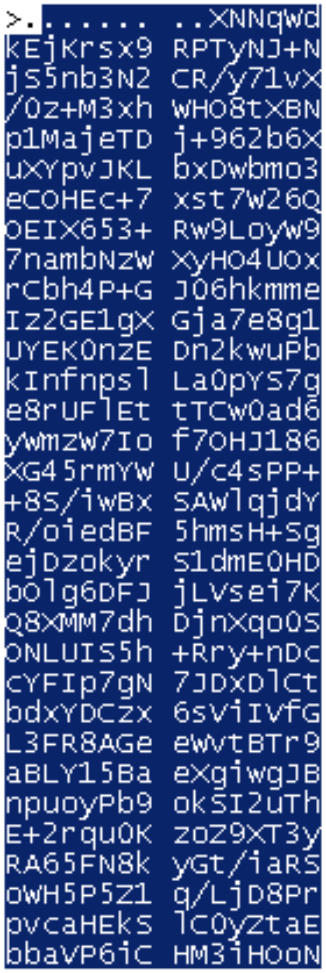
Στο στιγμιότυπο του σχήματος 14 βλέπουμε τα δεδομένα ενός πακέτου δεδομένων του JCatascopia Agent προς τον Server. Παρατηρούμε ότι τα δεδομένα είναι άμεσα αναγνώσιμα χωρίς καμιά επεξεργασία. Στο στιγμιότυπο του σχήματος 15 βλέπουμε

τα κρυπτογραφημένα δεδομένα ενός αντίστοιχου πακέτου. Παρατηρούμε ότι τα δεδομένα δεν είναι σε αναγνώσιμη μορφή.

Ο τρόπος με τον οποίο έγινε η παρακάτω αποτύπωση είναι α. εκτέλεση στην εικονική μηχανή Α του JCatascopia Server χωρίς κρυπτογράφηση, β. εκτέλεση στην εικονική μηχανή Β του JCatascopia Agent χωρίς κρυπτογράφηση, γ. εκτέλεση σε γραμμή εντολών μιας τρίτης εικονικής μηχανής που βρισκόταν στο ίδιο υπο-δίκτυο της εντολής «sudo tcpdump -i eth0 -w ~/capture.pcap». Η εντολή διακόπηκε μετά από χρόνο 10 λεπτών. Το αρχείο που δημιουργήθηκε, επεξεργάστηκε στο λογισμικό wireshark σε περιβάλλον Windows. Επειδή γνωρίζουμε ότι το σύστημα JCatascopia επικοινωνεί μέσω της πόρτας 4245, τοποθετήσαμε το φίλτρο «ip.src == 192.168.225.130 && tcp.port eq 4245» και απομονώσαμε τα πακέτα που μας αφορούνε. Η IP που βάλαμε είναι η IP του Server. Έπειτα επιλέξαμε ένα πακέτο και στο κάτω πλαίσιο του λογισμικού φάνηκε το παρακάτω αποτέλεσμα. Την ίδια διαδικασία εκτελέσαμε και για το Σχήμα 15 μόνο που στις αντίστοιχες εικονικές μηχανές JCatascopia Server και Agent εκτελέσαμε την έκδοση του JCatascopia με ενεργή την παράμετρο της κρυπτογράφησης. Η διάρκεια του κάθε πειράματος ήταν 10 λεπτά.



Σχήμα 14. Δεδ. μη κρυπτογραφημένα



Σχήμα 15. Δεδ. κρυπτογραφημένα

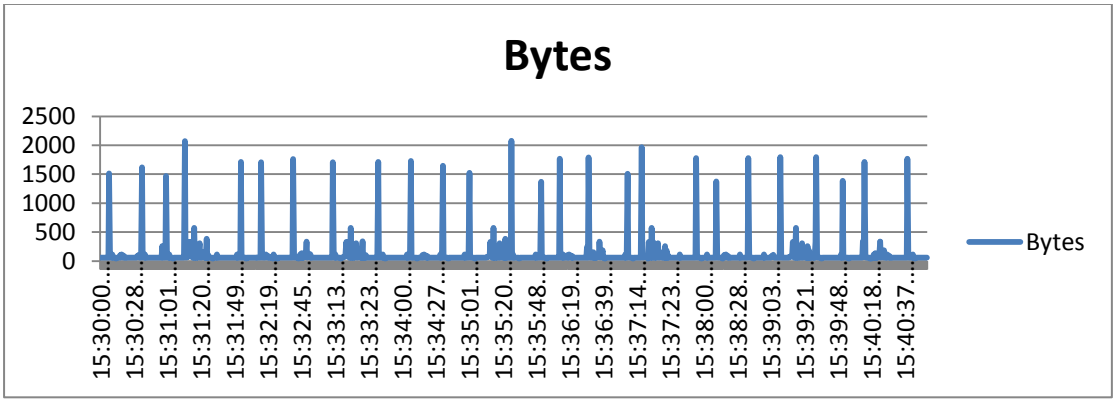
5.4 Επιβάρυνση του δικτύου

5.4.1 Μετρήσεις στην κίνηση του δικτύου στην πλευρά του Server

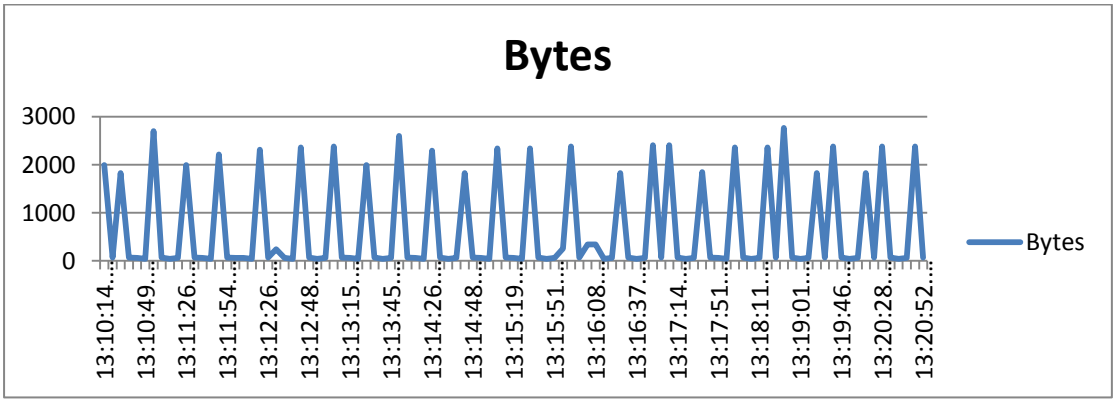
Τα παρακάτω γραφήματα αποτελούνε μετρήσεις που δείχνουν τον αριθμό των Bytes μεταξύ ενός JCatascopia Server και ενός JCatascopia Agent, για δέκα λεπτά, χωρίς κρυπτογράφηση AES, και με αυτήν, στην πλευρά του Server.

Η λήψη των μετρήσεων έγινε με τον τρόπο που περιγράφεται στην παράγραφο 5.3. Η μόνη διαφορά είναι ότι η εντολή «sudo tcpdump -i eth0 -w ~/capture.pcap» εκτελέστηκε σε γραμμή εντολών της εικονικής μηχανής του JCatascopia Server. Η ανάλυση των δεδομένων σε γραφήματα έγινε με τα εξής βήματα: α. εξαγωγή των

δεδομένων από το Wireshark σε csv αρχείο, β. άνοιγμα και επεξεργασία των δεδομένων σε Ms Excel.



Διάγραμμα 1. Κίνηση χωρίς κρυπτογράφηση.



Διάγραμμα 2.Κίνηση με κρυπτογράφηση.

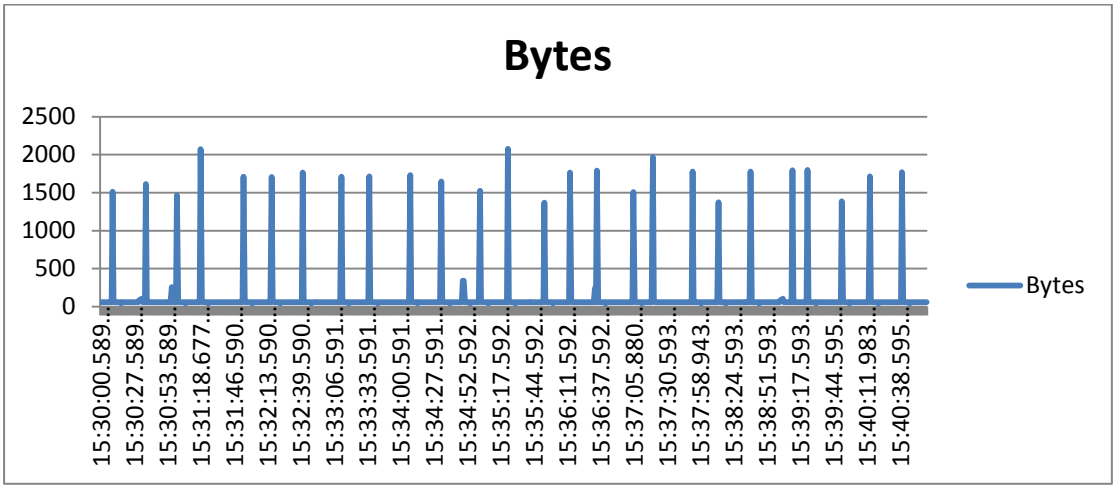
Παρατηρούμε ότι η αύξηση του αριθμού των Bytes είναι κατά μέσο όρο 500 bytes σε κάθε πακέτο, όταν γίνεται χρήση κρυπτογράφησης.

5.4.2 Μετρήσεις στην κίνηση του δικτύου στην πλευρά του Agent

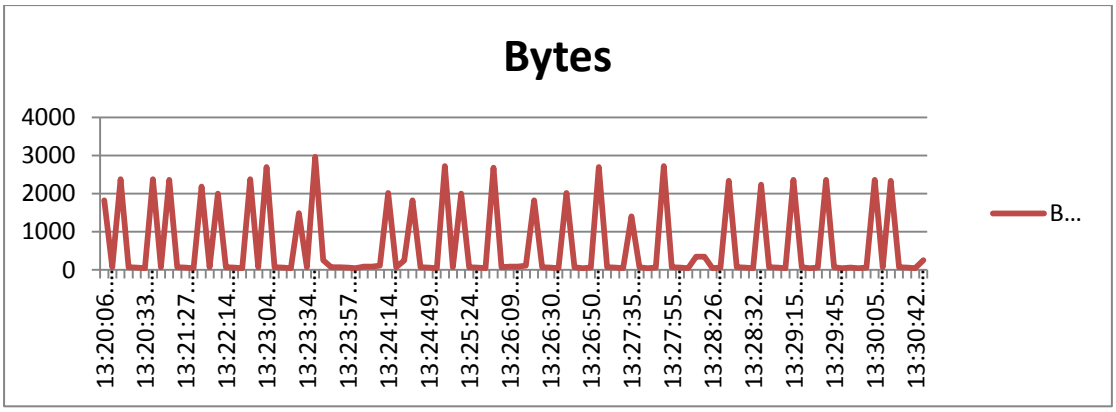
Τα παρακάτω γραφήματα αποτελούνε μετρήσεις που δείχνουν τον αριθμό των Bytes μεταξύ ενός JCatascopia Server και ενός JCatascopia Agent, για δέκα λεπτά, χωρίς κρυπτογράφηση AES, και με αυτήν, στην πλευρά του Agent.

Η λήψη των μετρήσεων έγινε με τον τρόπο που περιγράφεται στην παράγραφο 5.3. Η μόνη διαφορά είναι ότι η εντολή «sudo tcpdump -i eth0 -w ~/capture.pcap»

εκτελέστηκε σε γραμμή εντολών της εικονικής μηχανής του JCatascopia Agent. Η ανάλυση των δεδομένων σε γραφήματα έγινε με τα εξής βήματα: α. εξαγωγή των δεδομένων από το Wireshark σε csv αρχείο, β. άνοιγμα και επεξεργασία των δεδομένων σε Ms Excel.



Διάγραμμα 3. Κίνηση χωρίς κρυπτογράφηση.



Διάγραμμα 4. Κίνηση με κρυπτογράφηση.

Και σε αυτήν την μέτρηση παρατηρούμε ότι η αύξηση του αριθμού των Bytes είναι κατά μέσο όρο 500 bytes σε κάθε πακέτο, όταν γίνεται χρήση κρυπτογράφησης.

Ο λόγος που παρατηρούμε αυτό το γεγονός και στην παράγραφο 5.4.1 και στην παράγραφο 5.4.2 είναι ο εξής. Σύμφωνα με το RFC3602, Section 2.4 και το RFC 5652, Section 6.3, ο αλγόριθμος AES μπορεί να λειτουργήσει μόνο με μπλοκ δεδομένων των 128 bits. Εάν ένα πακέτο είναι μικρότερο τότε το συμπληρώνει

ώστε να μπορεί να λειτουργήσει ο αλγόριθμος σωστά. Ένα παράδειγμα padding είναι το εξής:

Το μέγεθος του παρακάτω μπλοκ είναι 8 bytes και χρειάζεται συμπλήρωμα για 4 bytes

... | DD DD DD DD DD DD DD DD | DD DD DD DD 04 04 04 04 |

5.5 Επιβάρυνση και απαιτήσεις σε πόρους

Τα εργαλεία που χρησιμοποιήθηκαν για τις μετρήσεις είναι το "top" πακέτο του Linux.

Η διαδικασία που ακολουθήθηκε για τη λήψη των μετρήσεων στις παραγράφους 5.5.1 και 5.5.2 είναι η εξής: α. εκτέλεση στην εικονική μηχανή Α του JCatascopia Server χωρίς κρυπτογράφηση και στην εικονική μηχανή Β του JCatascopia Agent χωρίς κρυπτογράφηση, β. εκτέλεση στην εικονική μηχανή Α του JCatascopia Server με κρυπτογράφηση και στην εικονική μηχανή Β του JCatascopia Agent με κρυπτογράφηση. Στην παράγραφο 5.5.3 προστέθηκαν στο πείραμα τέσσερις ακόμα εικονικές μηχανές στις οποίες εκτελέστηκε ο JCatascopia Agent. Η διάρκεια του κάθε πειράματος ήταν 10 λεπτά.

Σε γραμμή εντολών της κάθε εικονικής μηχανής, εκτελέσαμε την εντολή «pstree -A | less» και «ps –A» ώστε να εντοπίσουμε το ID των διεργασιών που μας ενδιαφέρουν. Στη συνέχεια εκτελούμε την εντολή «top» και διαδοχικά με το μενού ρυθμίζουμε την «εκτύπωση» της εντολής με τους διακόπτες f,W,E,l,t,m,W ώστε να λαμβάνουμε μετρήσεις μόνο για τη χρήση της μνήμης και του επεξεργαστή. Η εντολή που εκτελούμε στη συνέχεια στη γραμμή εντολών του Server και του Agent αντίστοιχα είναι «top -b -p 10326 -p 3135 -p 3138 > SrvAES_1Agent_10min.txt». Σε κάθε πείραμα εκείνο που αλλάζει είναι τα process id που στο συγκεκριμένο παράδειγμα είναι τα 10326, 3135, 3138. Τα process id τα βρίσκουμε κάθε φορά που εκκινούμε την εκτέλεση του πειράματος με τις εντολές «pstree» και «ps» όπως αναφέρθηκα και στην αρχή.

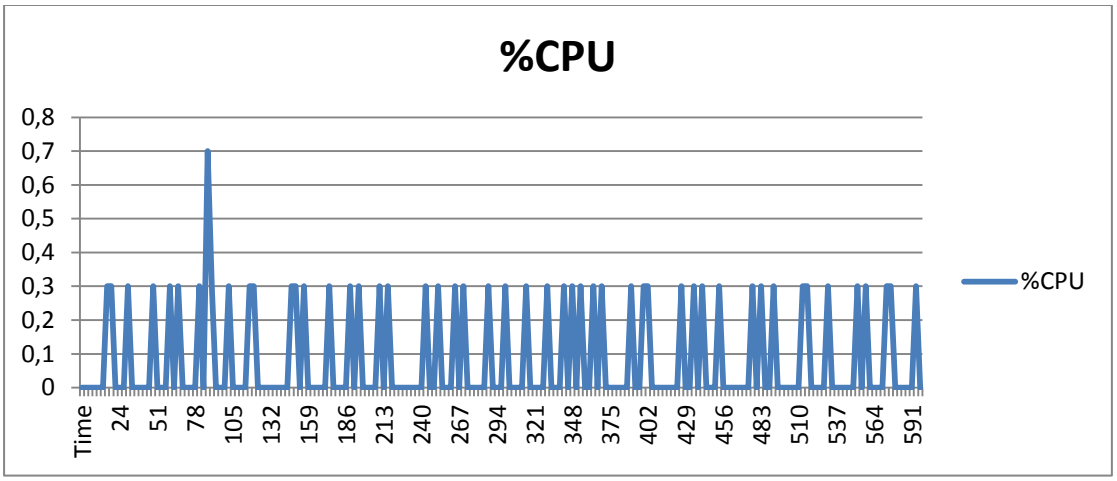
Στην παράγραφο 5.5.3 αλλάξαμε την παράμετρο aggregator.interval από 30 σε 5 που ήταν στις προηγούμενες μετρήσεις σε κάθε JCatascopia Agent. Με αυτόν τον

τρόπο οι μετρικές που στέλνονται στον JCatascopia Server είναι ανά 5 δευτερόλεπτα. Με αυτόν τον τρόπο αυξήσαμε αρκετά το φόρτο του πειράματος, καθώς από έναν Agent που στέλνει μετρικές κάθε 30 δευτερόλεπτα, το πείραμα είχε 5 Agents που έστελναν μετρικές κάθε 5 δευτερόλεπτα. Το ίδιο ακριβώς πείραμα εκτελέστηκε με κρυπτογράφηση και χωρίς κρυπτογράφηση.

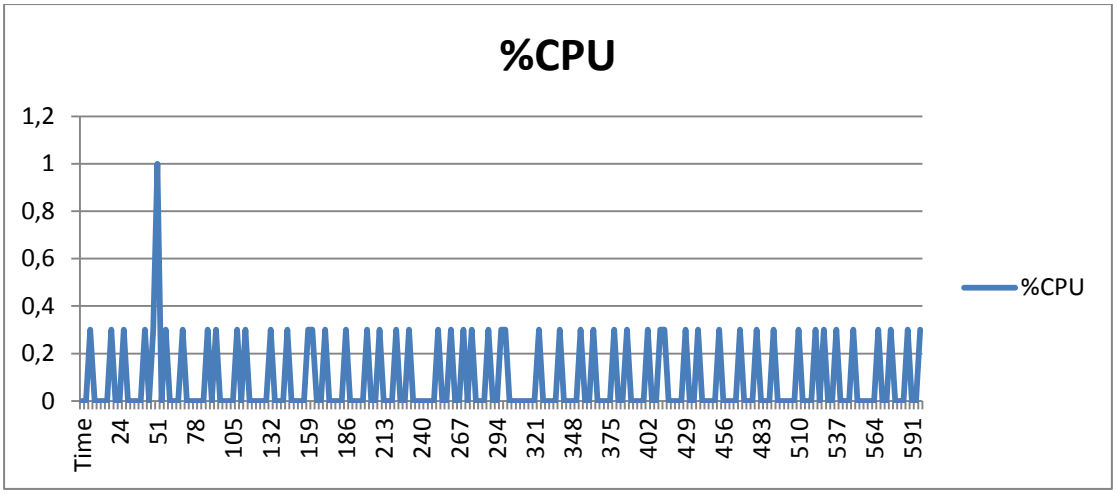
Η ανάλυση των δεδομένων σε γραφήματα έγινε σε όλες τις περιπτώσεις με τη χρήση του MS Excel.

5.5.1 Μετρήσεις για τις απαιτήσεις σε υπολογιστικούς πόρους στην πλευρά του Agent.

Τα παρακάτω γραφήματα αποτελούνε μετρήσεις που δείχνουν το ποσοστό της επεξεργαστικής ισχύς που χρειάζεται, για δέκα λεπτά, χωρίς κρυπτογράφηση AES, και με αυτήν, στην πλευρά του Agent. Τα πακέτα αποστέλλονται ανά 30 sec.

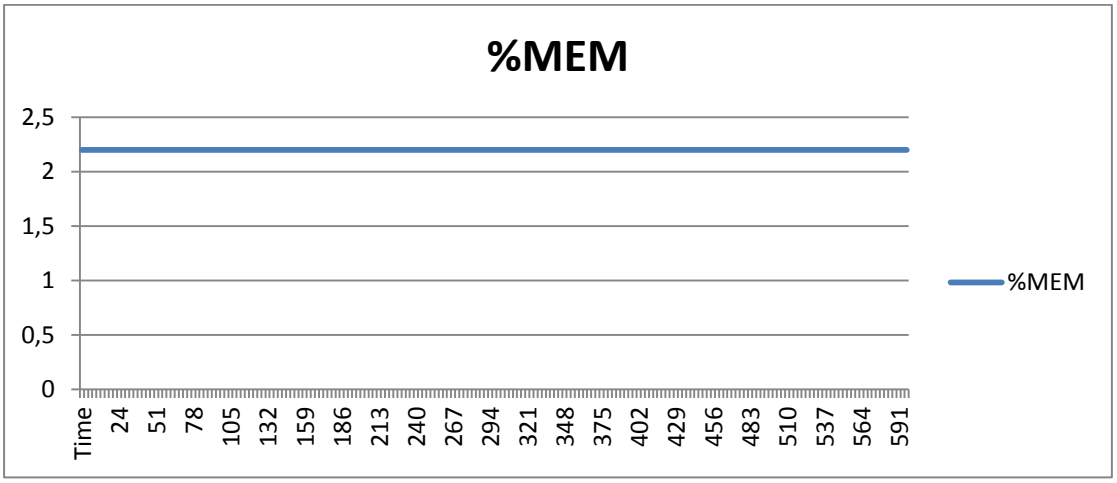


Διάγραμμα 5. Επεξεργαστική ισχύς χωρίς κρυπτογράφηση.

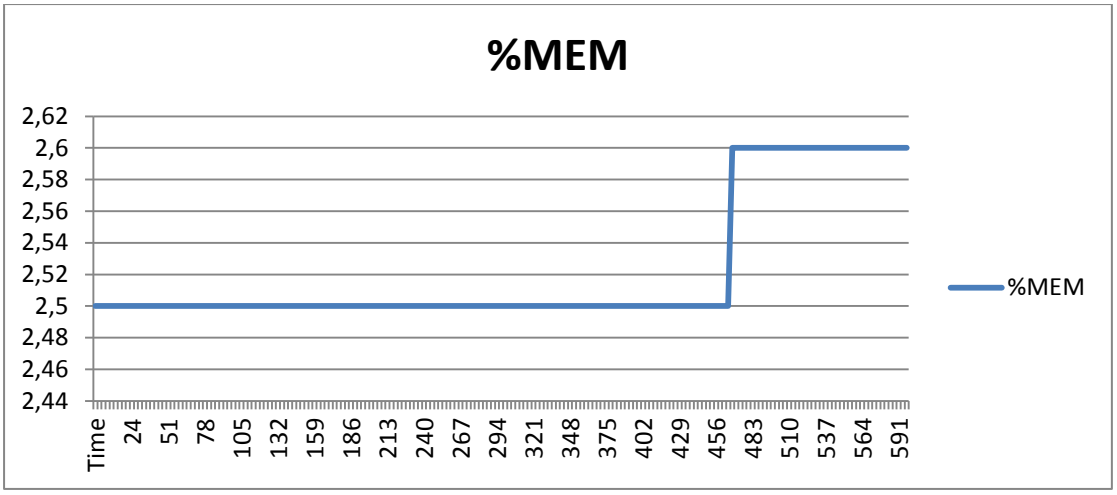


Διάγραμμα 6. Επεξεργαστική ισχύς με κρυπτογράφηση.

Τα παρακάτω γραφήματα αποτελούνε μετρήσεις που δείχνουν το ποσοστό της μνήμης RAM που χρειάζεται, για δέκα λεπτά, χωρίς κρυπτογράφηση AES, και με αυτήν, στην πλευρά του Agent. Τα πακέτα αποστέλλονται ανά 30 sec.



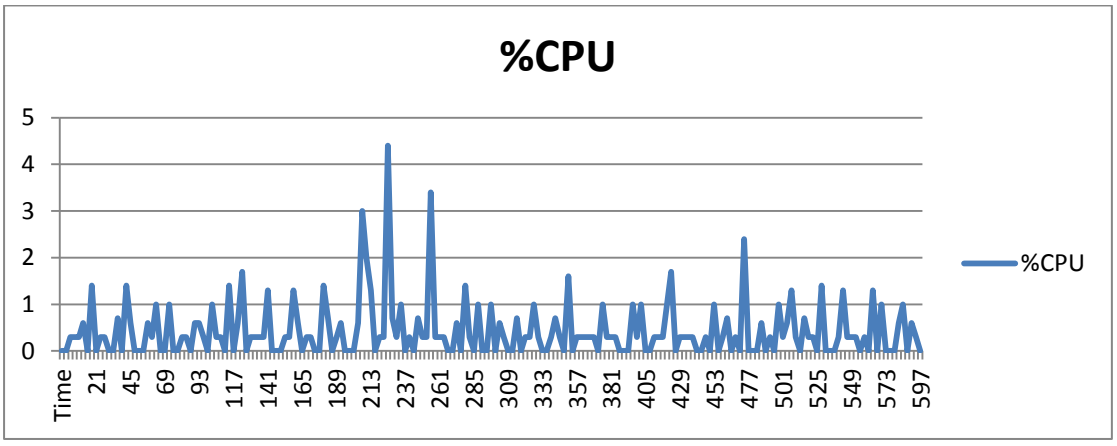
Διάγραμμα 7. Μνήμη χωρίς κρυπτογράφηση.



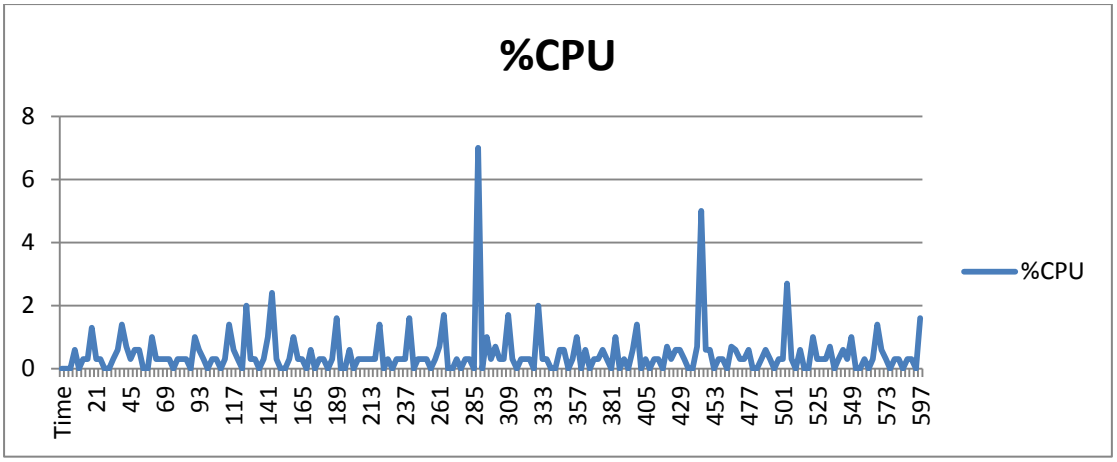
Διάγραμμα 8. Μνήμη με κρυπτογράφηση.

5.5.2 Μετρήσεις για τις απαιτήσεις σε υπολογιστικούς πόρους στην πλευρά του Server.

Τα παρακάτω γραφήματα αποτελούνε μετρήσεις που δείχνουν το ποσοστό της επεξεργαστικής ισχύς που χρειάζεται, για δέκα λεπτά, χωρίς κρυπτογράφηση AES, και με αυτήν, στην πλευρά του Server. Τα πακέτα αποστέλλονται ανά 30 sec.

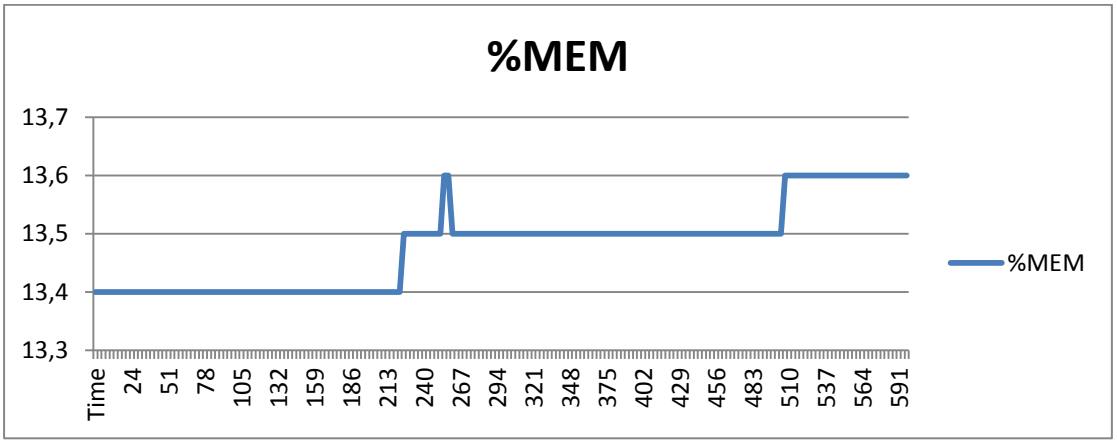


Διάγραμμα 9. Επεξεργαστική ισχύς χωρίς κρυπτογράφηση.

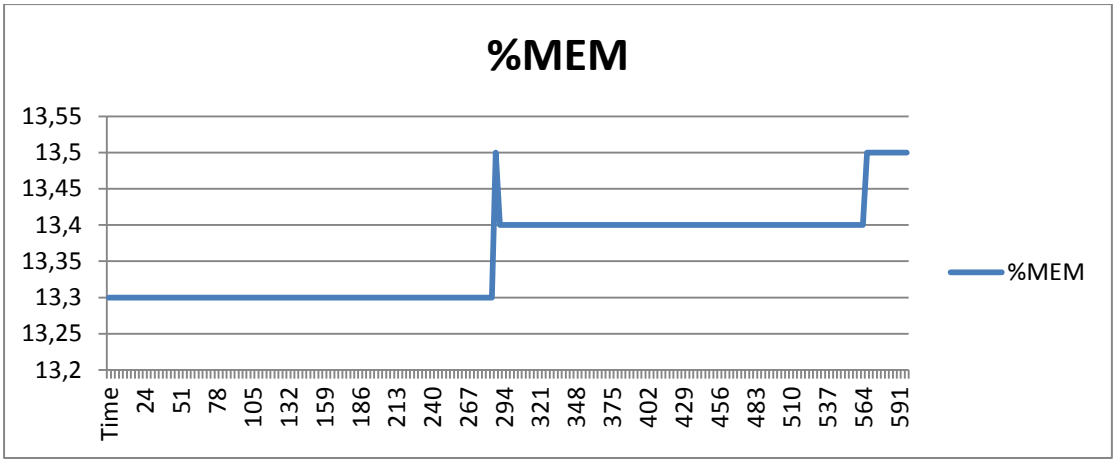


Διάγραμμα 10. Επεξεργαστική ισχύς με κρυπτογράφηση.

Τα παρακάτω γραφήματα αποτελούνε μετρήσεις που δείχνουν το ποσοστό της μνήμης RAM που χρειάζεται, για δέκα λεπτά, χωρίς κρυπτογράφηση AES, και με αυτήν, στην πλευρά του Server. Τα πακέτα αποστέλλονται ανά 30 sec.



Διάγραμμα 11. Μνήμη χωρίς κρυπτογράφηση.

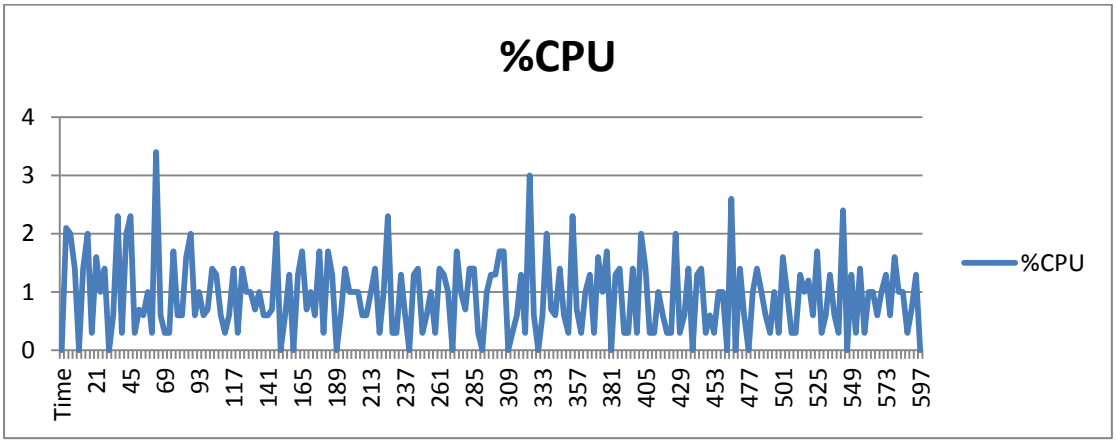


Διάγραμμα 12. Μνήμη με κρυπτογράφηση.

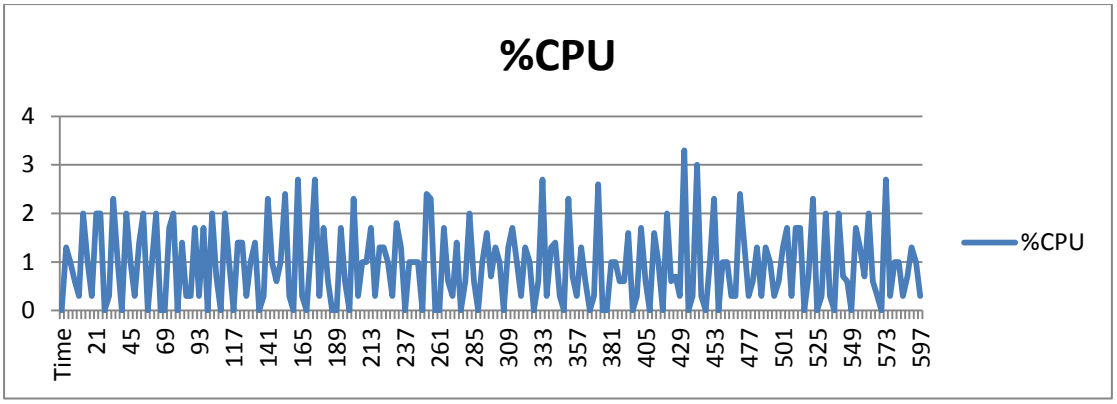
Στις προηγούμενες μετρήσεις δεν παρατηρούμε κάποια σημαντική διαφορά και δεν μας βοηθούνε πέραν του ότι η κρυπτογράφηση δεν επιβαρύνει το σύστημα όταν αυτό αποτελείται από έναν Server και έναν Agent.

5.5.3 Μετρήσεις για τις απαιτήσεις σε υπολογιστικούς πόρους στην πλευρά του Server αυξάνοντας τον φόρτο και τους clients.

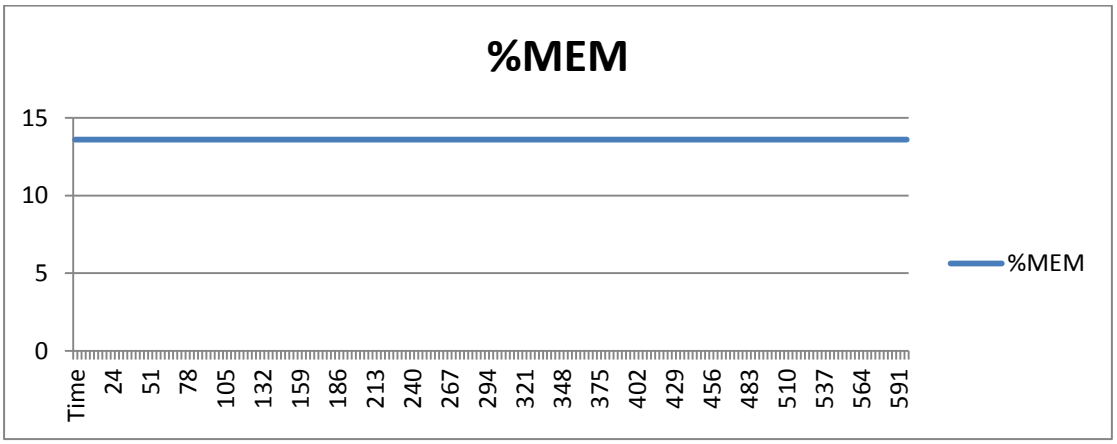
Επειδή από τις προηγούμενες δεν βγάλαμε κάποιο συμπέρασμα κάναμε μετρήσεις που αφορούνε έναν Server και 5 Agents, οι οποίοι στέλνουν μετρικές ανά 5 sec ο έκαστος.



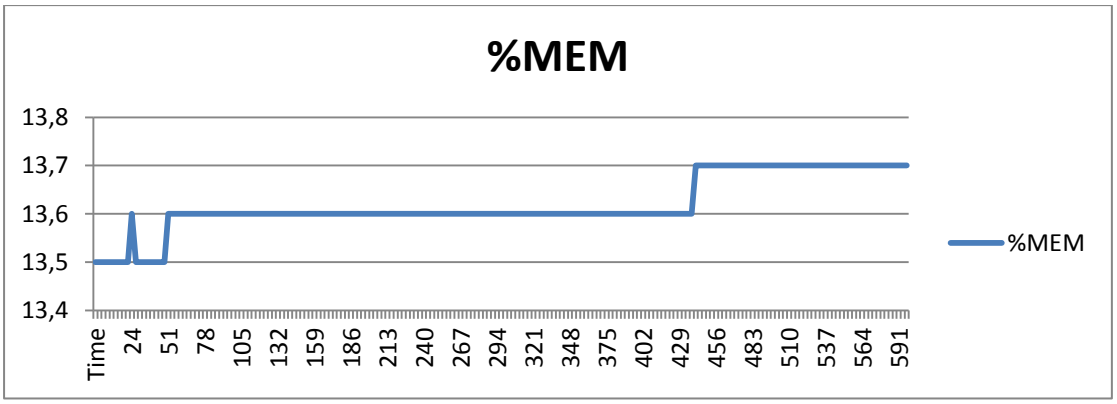
Διάγραμμα 13. Επεξεργαστική ισχύς χωρίς κρυπτογράφηση.



Διάγραμμα 14. Επεξεργαστική ισχύς με κρυπτογράφηση.



Διάγραμμα 15. Μνήμη χωρίς κρυπτογράφηση.



Διάγραμμα 16. Μνήμη με κρυπτογράφηση.

Από τις παραπάνω μετρήσεις συμπεραίνουμε εύκολα ότι αυξήθηκε η απαίτηση σε επεξεργαστική ισχύς, αλλά όχι σε απαγορευτικό βαθμό.

Συνοπτικά από την ανάλυση των πακέτων μεταξύ του πράκτορα και του εξυπηρετητή διαφαίνεται ότι είναι κρυπτογραφημένα και δεν μπορεί να γίνει η ανάγνωσή τους από κάποιον τρίτο που δεν γνωρίζει το κλειδί κρυπτογράφησης. Αυτό προσδίδει ακεραιτότητα των δεδομένων και εμπιστευτικότητα μεταξύ του πράκτορα και του εξυπηρετητή όπως ήταν και ο αρχικός στόχος της διατριβής.

Οι μετρήσεις που έγιναν με βάση την επεξεργαστική ισχύ και την μνήμη που καταναλώνεται κατά την εκτέλεση της κρυπτογράφησης/αποκρυπτογράφησης, αποδεικνύουν ότι οι απαιτήσεις του αλγορίθμου είναι αμελητέες και δεν προσθέτουν περαιτέρω κόστος στην υποδομή της υπολογιστικής νέφους και από τη μεριά του πράκτορα και από τη μεριά του εξυπηρετητή. Αυτό συμβαίνει γιατί, όπως προέκυψε και από την βιβλιογραφική έρευνα η Intel, έχει προσθέσει στο πακέτο εντολών των επεξεργαστών της (instruction set), εντολές που υλοποιούν αποκλειστικά τον αλγόριθμο κρυπτογράφησης AES.

Οι μετρήσεις που αφορούν την επιφόρτιση του δικτύου αποδεικνύουν επίσης ότι η επιβάρυνση στους δικτυακούς πόρους είναι μικρή και δίνει την δυναμική που απαιτείται για τη χρήση του συστήματος JCatascopia σε πραγματικές υποδομές υπολογιστικής νέφους χωρίς ενδοιασμούς.

Κεφάλαιο 6
Επίλογος

6.1 Συμπεράσματα

Τα συμπεράσματα που βγαίνουν από τις μετρήσεις που προηγήθηκαν είναι ότι τόσο η επιφόρτιση του δικτύου όσο και οι ανάγκες σε υπολογιστικούς πόρους είναι αρκετά μικρές σε σχέση με τα οφέλη που προκύπτουν. Πρακτικά, η χρήση κρυπτογράφησης με τον αλγόριθμο AES στο σύστημα JCatascopia δεν επιβαρύνει ούτε το δίκτυο ούτε την υπολογιστική ισχύ των συστημάτων που εμπλέκονται.

Σύμφωνα με τον οργανισμό «Common Vulnerabilities and Exposures» [35] αλλά και την Oracle [33] υπάρχουν γνωστές ευπάθειες στις συγκεκριμένες βιβλιοθήκες κρυπτογράφησης της Java, οι οποίες όμως έχουν επιδιορθωθεί από αντίστοιχες ενημερώσεις.

Μια ακόμη ευπάθεια του συστήματος είναι ότι οποιοσδήποτε κάνει από-μεταγλώττιση της εφαρμογής θα μπορεί να δει το κλειδί που χρησιμοποιείται οπότε θα μπορεί να αποκρυπτογραφήσει και τα δεδομένα που αποστέλλονται.

6.2 Μελλοντική εργασία

Μια κρίσιμη εργασία που θα μπορούσε να υλοποιηθεί είναι η ανταλλαγή του κλειδιού κρυπτογράφησης μέσω υποδομής δημοσίου κλειδιού RSA ώστε να είναι ασφαλή η ανταλλαγή του κλειδιού μεταξύ Server και Agent την πρώτη φορά. Έτσι και το κομμάτι της ασφαλούς αρχικοποίησης ενός πράκτορα θα αυτοματοποιηθεί πλήρως όπως είναι και η υπόλοιπη λειτουργία του JCatascopia.

Τα μοναδικά χαρακτηριστικά για κάθε εικονική μηχανή είναι ο συνδυασμός των τριών στοιχείων MAC Address, UUID (Universally Unique Identifier) [37] και CPU Info. Αυτά τα στοιχεία τα γνωρίζει ο πάροχος της υποδομής, ο οποίος θα μπορούσε να δημιουργεί τα κλειδιά με συνδυασμό αυτών των τριών στοιχείων. Αυτή είναι μια λειτουργία η οποία μπορεί να προστεθεί επίσης μελλοντικά στο JCatascopia.

Επίσης στην ευπάθεια που αναφέρω στην παράγραφο 6.1 θα μπορούσε να γίνει κλείδωμα των κλάσεων ώστε να μην είναι δυνατή η από-μεταγλώττισή τους. Υπάρχουν βιβλιοθήκες ανοικτού κώδικα που εκτελούν αυτήν την λειτουργία όπως είναι π.χ. η proguard. Ένα μικρό παράδειγμα κάνοντας χρήση αυτής της βιβλιοθήκης είναι το παρακάτω:

Απο-μεταγλωτισμένος κώδικας χωρίς τη χρήση της proguard:

btnNew = changeButtonLabel(btnNew, language.getText("new"));»

Απο-μεταγλωτισμένος κώδικας με τη χρήση της proguard:

d = a(d, n.a("new"));

Βιβλιογραφία

[1] Eurostat, «ICT usage in enterprises in 2014,» 2014.

[2] wikipedia.org, «Cloud Computing,» 2015.

[3] P. G. D. M. D. Trihinas D., «JCatascopia: Monitoring Elastically Adaptive Applications in the Cloud, Cluster, Cloud and Grid Computing (CCGrid),» 2014 14th IEEE/ACM International Symposium on, 2014, pp. 226-235.

[4] M. P., «The NIST Definition of Cloud Computing,» Τόμ. %1 από %2NIST Special Publication 800-145, 2011.

[5] NIST, «Cloud Computing Synopsis and Recommendations,» Τόμ. %1 από %2Special Publication 800-146, pp. 4-1, 2012.

[6] jimir.org, «Opportunities and Challenges of Cloud Computing to Improve Health Care Services,» 2011.

[7] NIST, «Special Publication 800-21,» p. 12, 2005.

[8] VMware, «Performance Analysis Methods,» 2008.

[9] VMware, «VMware Infrastructure Architecture Overview,» 2006.

[10] Intel, «Intel Advanced Encryption Standard (AES) New Instructions Set,» 2012.

[11] F. Y., «CryptVMI: a flexible and encrypted virtual machine introspection system in the cloud,» 2nd international workshop on Security in cloud computing, 2014, pp. 11-18.

[12] J. P., «DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds,» *Future Generation Computer Systems,* τόμ. 29, αρ. 8, pp. 2041-2056, 2013.

[13] K. F., «A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives,» *Journal of Parallel and Distributed Computing,* p. 2918–2933, 2014.

[14] I. A.S., «CloudSec: A security monitoring appliance for Virtual Machines in the IaaS cloud model,» *Network and System Security (NSS), 2011 5th International Conference on, IEEE,* pp. 113-120, 2011.

[15] V. L. M., «Locking the sky: a survey on IaaS cloud security,» *Computing,* τόμ. 91, αρ. 1, pp. 93-118, 2011.

[16] CSA.org, «The Notorious Nine Cloud Computing Top Threats in 2013,» 2013.

[17] J. M., «MonPaaS: An Adaptive Monitoring Platform as a Service for Cloud Computing Infrastructures and Services,» 2014.

[18] K. K., «Architecture for High Confidence Cloud Monitoring,» Cloud Engineering (IC2E), 2015 IEEE International Conference, 2015, pp. 195-200.

[19] T. C. Group, «Trusted Platform Module (TPM) Summary,» 2008.

[20] Intel, «Intel Trusted Platform Module (TPM module-AXXTPME3) Hardware User's Guide,» 2011.

[21] B. S., «vTPM: Virtualizing the Trusted Platform Module,» *Proceeding USENIX-SS'06 Proceedings of the 15th conference on USENIX Security Symposium,* τόμ. 15, αρ. 21, 2006.

[22] A. Mandal, «Performance Evaluation of Cryptographic Algorithms: DES and AES,» *Electrical, Electronics and Computer Science (SCEECS), 2012 IEEE Students' Conference,* pp. 1-5, 2012.

[23] Intel, «Intel AES-NI Performance Testing on Linux/Java Stack,» 2012.

[24] N. A. Kofahi, «Java Implementation and Performance Evaluation of Some Cryptographic Ciphers under WinXP and Linux Operating System Platforms,» τόμ. 3, αρ. 4, 2013.

[25] A. Bogdanov, «Biclique Cryptanalysis of the Full AES, Chapter Advances in Cryptology – ASIACRYPT,» τόμ. 7073, pp. 344-371, 2011.

[26] IEEE, «Java, Java, Java,» αρ. ISSN: 0278-6648, pp. 33-37, 1998.

[27] IEEE, «Computer,» τόμ. 38, αρ. 5, 2005.

[28] W3.org, «W3.org,» 11 02 2004. [Ηλεκτρονικό]. Available: http://www.w3.org/TR/ws-arch/#whatis.

[29] Σ. Χ., «Υλοποίηση εφαρμογών με γλώσσα SQL,» 2001, p. 25.

[30] Κ. Λιμνιώτης, «Κρυπταλγόριθμοι Τμήματος,» 2014, p. Κεφ.7:28.

[31] F. K. Gürkaynak, «GALS System Design: Side Channel Attack Secure Cryptographic Accelerators,» 2006, p. 24.

[32] owasp.org, «Using the Java Cryptographic Extensions,» 2014.

[33] oracle.com, «Oracle Critical Patch Update Advisory,» 2015.

[34] wireshark.org, 2015.

[35] cve.mitre.org, «CVE-2015-2601,» 2015. [Ηλεκτρονικό].

[36] oracle.com, «Java Cryptography Architecture Oracle Providers Documentation for Java Platform Standard Edition 7,» 2014.

[37] D. D. MCCRORY, «TECHNIQUES FOR IDENTIFYING AND COMPARING VIRTUAL MACHINES IN A VIRTUAL MACHINE SYSTEM». USA Ευρεσιτεχνία US 20100083251A1, 1 April 2010.