ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΜΕΣΟΛΟΓΓΙΟΥ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΓΩΝ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗ
ΔΙΟΙΚΗΣΗ ΚΑΙ ΤΗΝ ΟΙΚΟΝΟΜΙΑ

Πτυχιακή Εργασία

Υλοποίηση Συστήματος Ηλεκτρονικής Ψηφοφορίας

Φοιτητής: Λουνής Σταύρος Α.Μ.:7873

Εισηγητής:Δρ. Τριανταφύλλου Βασίλης

Περιεχόμενα

Πρόλογος

Με το πέρασμα του χρόνου, οι αλλαγές στον τρόπο ζωής των πολιτών που απαρτίζουν μια κοινωνία είναι δραματικές. Οι συνήθειες και οι συμπεριφορές που παλαιότερα αποτελούσαν Status quo, πλέον καταργούνται και δίνουν τη θέση τους σε νέες ιδέες και θεσμούς. Ένας νέος θεσμός που έχει πλέον παγιωθεί είναι και το Internet.

Η εποχή κατά την οποία το διαδίκτυο ήταν μια καθαρά ακαδημαϊκή υπόθεση και αυτό που κυρίως πρόσφερε στους χρήστες του ήταν η ανταλλαγή γραπτών μηνυμάτων ανάμεσα στους διάφορους επιστήμονες για θέματα του ενδιαφέροντος τους και τελευταία η περιήγηση στον παγκόσμιο ιστό μέσω του World Wide Web έχει περάσει οριστικά και αμετάκλητα.

Πλέον μέλη του παγκόσμιου κυβερνοχώρου, είναι άτομα που ποικίλουν από ανθρώπους που απλά περνούν το χρόνο τους σερφάροντας, σε ανθρώπους που δουλεύουν και ζουν τη εικονική τους ζωή καθημερινά. Ο αριθμός των ανθρώπων που πλέον έρχονται σε επαφή με το διαδίκτυο συνεχώς αυξάνεται Αυτή είναι η μια συνιστώσα της εξάπλωσης του διαδικτύου, η ποσοτική. Σαφώς υπάρχει και η ποιοτική και υποδηλώνει την μεγάλη αλλαγή και εμπλουτισμό των υπηρεσιών που παρέχει το διαδίκτυο στους χρήστες τους. Ένα μέσο που γεφυρώνει την υπόσταση του πολίτη του διαδικτύου με την υπόσταση του πολίτη στο κράτος, είναι η δυνατότητα που του παρέχεται να εκτελέσει ορισμένα από τα πολιτικά δικαιώματά του μέσω του διαδικτύου. Το σημαντικότερο δικαίωμα του κάθε πολίτη είναι η έκφραση των απόψεων σε θέματα που τον αφορούν. Η Ηλεκτρονική Ψηφοφορία είναι η διαδικτιακή απάντηση που προσφέρει αυτό το δικαίωμα.

Η εργασία αυτή έχει σαν σκοπό την δημιουργία ενός τέτοιου συστήματος. Ενός συστήματος ηλεκτρονικής ψηφοφορίας, που θα αποτελεί την προσπάθεια μεταφοράς της κλασσικής διαδικασίας σε ηλεκτρονική μορφή. Τα κεφάλαια 2 έως 5 πραγματεύονται την μετάβαση της δημοκρατίας σε ηλεκτρονική δημοκρατία, τις μορφές και τους τρόπους υλοποίησής της.

Τα κεφάλαια 6 – 8 περιγράφουν την Ηλεκτρονική Ψηφοφορία, όπως έχει μετασχηματιστεί στην παρούσα της μορφή, τα πλεονεκτήματα και τα μειωνεκτήματά της. Επιπλέον γίνεται μια περιγραφή των προγενέστερων συστημάτων ηλεκτρονικής ψηφοφορίας.

Τα κεφάλαια 9 – 14 περιγράφουν το προτεινόμενο σύστημα ηλεκτρονικής ψηφοφορίας IBB/PMA. Από τη σύλληψη της ιδέας, στον σχεδιασμό, την υλοποίηση και τις ανάγκες υποστήριξης σε υλικολογισμικό για την ορθή του λειτουργία.

Τέλος το κεφάλαιο 15 περιέχει τον κώδικα των προγραμμάτων που χρειάζονται για την εκτέλεση του IBB/PMA Electronic Voting System.

1. Εισαγωγή

Η ευρεία ανάπτυξη του Internet και η χρήση του Διαδικτύου έχει δημιουργήσει πολλές πρωτοβουλίες που σκοπό έχουν την εφαρμογή των καινοτομιών στις πληροφορίες και επικοινωνίες, για τη δημιουργία αυτού που καλείται «ψηφιακή μορφή» της Δημοκρατίας. Τα πρότυπα άμεσης Δημοκρατίας όπου οι πολίτες συμμετέχουν στην διαδικασία λήψης αποφάσεων θα είναι πάντοτε η κύρια προτεραιότητα για τις κοινωνικές μονάδες, όπως η Τοπική Αυτοδιοίκηση, Κρατική Διοίκηση ακόμα και Κοινωνίες Εθνών.

Οι ερευνητές της Ηλεκτρονικής Δημοκρατίας προτείνουν ότι οι δημοκρατικές κυβερνήσεις και οργανισμοί θα πρέπει να υλοποιήσουν στενότερους δεσμούς μεταξύ αυτών και των πολιτών, επηρεάζοντας τον τρόπο της λήψης αποφάσεων με τις επιδράσεις των πολιτών, επιτρέποντας στο κοινό να πάρει ενεργό μέρος την πολιτική διαδικασία. Οι εκλογές αποτελούν τον επίσημο μηχανισμό που χρησιμοποιούν οι άνθρωποι για να εκφράσουν τις απόψεις τους. Το γεγονός είναι ότι με την κλασσική μορφή ψηφοφορίας η προσέλευση των πολιτών στις κάλπες σημαδεύεται από χαμηλή συμμετοχή. Ο σκοπός της ψηφιοποίησης της διαδικασίας αυτής είναι πρωτίστως η αύξηση του επιπέδου συμμετοχής.

Ένα ηλεκτρονικό σύστημα ψηφοφορίας βασισμένο στο Διαδίκτυο, ορίζεται *ως ένα Ηλεκτρονικό Σύστημα που χρησιμοποιεί ηλεκτρονικές ψήφους και επιτρέπει στους ψηφοφόρους να μεταφέρουν την ψήφο τους στους υπεύθυνους της εκλογικής διαδικασίας μέσω του Διαδικτύου.* Είναι κατανοητό βέβαια ότι όταν αναφερόμαστε σε ένα σύστημα Ηλεκτρονικής Ψηφοφορίας τα τμήματα της τεχνολογίας που πρέπει να ενοποιηθούν προέρχονται πρακτικά από κάθε τομέα της Κοινωνίας της Πληροφορίας. Ακόμα υπάρχουν αναρίθμητες ερωτήσεις που αφορούν διαδικασίες ασφάλειας, διαύγειας, παρουσίασης και επικοινωνιών.

Ο σκοπός των επιστημόνων είναι να δώσουν απαντήσεις σ' αυτά τα ερωτήματα. Το IBB/PMA σύστημα Ηλεκτρονικής Ψηφοφορίας ολοκληρώνει τις δυνατότητες που υπάρχουν στα προγενέστερά τους συστήματα και προτείνει έναν αριθμό καινοτομιών που στόχο έχουν την επίτευξη υψηλότερου επιπέδου ασφάλειας και διαύγειας, σε ένα σύστημα Ηλεκτρονικής Ψηφοφορίας, στον τομέα της ψηφοφορίας και της καταμέτρησης των ψήφων.

Οι καινοτομίες αυτές περιλαμβάνουν την τεχνική των Physical Multiple Administrators (PMA) και την τεχνική Identical Ballot Boxes (IBBs). Οι δύο τεχνικές αυτές, σε συνδυασμό με την νέα προτεινόμενη διαδικασία, την "Verification Obligation", δίνουν στο σύστημα το απαραίτητο ηλεκτρονικό και διαδικαστικό κύρος, που απαιτείται να έχει ένα σύστημα ηλεκτρονικής ψηφοφορίας. Το IBB/PMA σύστημα ηλεκτρονικής ψηφοφορίας έχει δημιουργηθεί για να προσαρμόζεται στις ανάγκες και τους νομικούς κανόνες που ισχύουν στο Ελληνικό Κράτος.

2. Δημοκρατία

Η Δημοκρατία αποτελεί ένα από τους σημαντικότερους θεσμούς στην σημερινή κοινωνία, σχεδόν σε όλο τον κόσμο. Η ετυμολογική ανάλυση του όρου δημοκρατία έχει ως αποτέλεσμα ότι η λέξη προέρχεται από τις λέξεις "Δήμος" και "Κρατώ". Δημοκρατία λοιπόν είναι το πολίτευμα στο οποίο η εξουσία πηγάζει από τον ίδιο τον λαό.

Γενέτειρα της Δημοκρατίας είναι η Ελλάδα. Για να κατανοήσουμε λοιπόν το Δημοκρατικό Πολίτευμα θα ανατρέξουμε στην αρχαία Ελλάδα, από όπου μπορούμε και να πάρουμε την πληρέστερη εικόνα. Σαφώς η δημοκρατία από τα αρχικά στάδια δημιουργίας της, παρουσιάζει κάποιες αλλαγές, άμεσο αποτέλεσμα της εξελικτικής πορείας που ακολούθησε. Αλλά από την εποχή που βρισκόταν στην ακμή της μπορούμε να την χαρακτηρίσουμε ως ιδανική. Θα αναφερθούμε στον χρυσό αιώνα του Περικλή.

Οι βασικότερες αρχές, στις οποίες στηριζόταν η Δημοκρατία ήταν η ισονομία, η αξιοκρατία και η ελευθερία του λόγου. Ο αυτοσκοπός των προηγούμενων αρχών ήταν η προαγωγή του κοινού καλού. Ένα δημοκρατικό πολίτευμα είναι πετυχημένο όταν τοποθετεί πάνω από το ατομικό το κοινό καλό, δίνοντας συγχρόνως τη δυνατότητα στους ανθρώπους να αναπτυχθούν και να ευημερήσουν.

Οι νόμοι που ήταν υπεύθυνοι για την διατήρηση της δημοκρατίας ήταν ίσοι για όλους τους ανθρώπους. Όλοι είχαν τα ίδια δικαιώματα αλλά και τις ίδιες ευθύνες. Η ισονομία και η τήρηση των νόμων χαρακτηριστικά μπορεί να κατανοηθεί από την αντίληψη που επικρατούσε ότι ο πρώτος υπηρέτης των νόμων όφειλε να είναι ο κυβερνήτης της πόλης.

Στο πολίτευμα της δημοκρατίας, η αρχή της αξιοκρατίας έδινε σε όλους τους πολίτες το δικαίωμα να αναλάβουν κάποιο αξίωμα, ανεξαρτήτως της κοινωνικής θέσης ή της οικονομικής κατάστασης στην οποία βρισκόταν. Ο κάθε πολίτης έβρισκε την θέση που του αρμόζει καλύτερα και ολόκληρη η πολιτεία επωφελούταν επειδή ο κατάλληλος άνθρωπος βρισκόταν στην κατάλληλη θέση.

Μια ακόμα θεμελιώδης αρχή του δημοκρατικού πολιτεύματος είναι η ελευθερία του λόγου. Οι αρχαίοι Έλληνες χρησιμοποιούσαν τις λέξεις ισηγορία και παρρησία για να δηλώσουν την έννοια της ελευθερίας του λόγου. Ισηγορία είναι το δικαίωμα που είχαν όλοι οι πολίτες να αγορεύσουν, να μιλήσουν δηλαδή στην Εκκλησία του Δήμου. Η παρρησία είναι κάτι σημαντικότερο από την ισηγορία. Παρρησία είναι το δικαίωμα του κάθε ανθρώπου να μπορεί να πει την άποψή του ελεύθερα, ακόμη και αν είναι αντίθετη από την γνώμη των υπολοίπων. Το να εκφράσεις την γνώμη σου δεν έχει κάποιο νόημα εάν δεν μπορείς να πεις αυτό που πραγματικά αισθάνεσαι επειδή υπάρχει περίπτωση να αποδοκιμαστείς από τους άλλους. Η παρρησία προϋποθέτει ευρύτητα πνεύματος σε τέτοιο βαθμό που να μπορεί ο πολίτης να δεχθεί το αντίθετο από αυτό που ο ίδιος πιστεύει. Ο Βολτέρος έρχεται με τα λόγια του να ολοκληρώσει τις έννοιες ισηγορία και παρρησία. «Δεν πιστεύω ούτε μια λέξη από όσα λες αλλά θα υπερασπίζω ακόμη και με τη ζωή μου το δικαίωμά σου ελεύθερα να λες όσα πρεσβεύεις».

Βρισκόμαστε πλέον σε μια εποχή όπου τα ανθρώπινα δικαιώματα είναι αναγνωρισμένα σε όλες τις δημοκρατικές χώρες του κόσμου και πλέον το φιλοσοφικό πλαίσιο της δημοκρατίας έχει ολοκληρωθεί. Για να επιτευχθεί αυτό χρειάστηκαν ορισμένοι συνειδητοί πολίτες, συνειδητοί άνθρωποι. Χάρη σε αυτούς τους πολίτες ευδοκίμησε η δημοκρατία, με αυτούς τους πολίτες εξελίχθηκε και αυτοί είναι οι

πολίτες που θα την οδηγήσουν στην πραγματοποίηση στην σημερινή τεχνοκρατική κοινωνία.

χρόνου και τις συνιστώσες που συνέβαλλαν σε αυτήν την αλλαγή. Η μελέτη θα γίνει σαφώς με χρονολογική σειρά.

A. Teledemocracy

Η αρχαιότερη έννοια της ηλεκτρονικής ψηφοφορίας είναι η Teledemocracy, ελληνιστή Τηλεδημοκρατία. Αναπτύχθηκε το 1970 και αποτέλεσε την πρώτη ευρέως συζητημένη έννοια της ηλεκτρονικής δημοκρατίας το 1980. Παρόλο που είναι αδύνατο να εντοπιστεί ο άνθρωπος που εφεύρεσε τον όρο Teledemocracy, ο Ted Becker ήταν αυτός που το χρησιμοποιούσε σε πολλές έρευνές του. Ο Becker πειραματιζόταν στην χρήση καλωδιακής τηλεόρασης, για την υποστήριξη των πολιτικών στην λήψη αποφάσεων στα τέλη του 1970. Ο Christopher Arterton επίσης χρησιμοποιούσε αυτόν τον όρο στις κατά καιρούς ερευνητικές συνεισφορές του στο έργο του Becker κυρίως το 1987. Η Teledemocracy στοχεύει να καθιερώσει περισσότερες μορφές άμεσης Δημοκρατίας στα δημοκρατικά πολιτεύματα και προσπαθεί να χρησιμοποιήσει τις νέες τεχνολογίες επικοινωνιών για αυτόν το σκοπό. Οι οπαδοί της Teledemocracy στις Ηνωμένες Πολιτείες Αμερικής υποστηρίζουν ότι η σημερινή αλλαγή στην Αμερικάνικη Πολιτική παρακινείται από δύο παράγοντες: «Την διακονταετή πορεία προς την πολιτική ισότητα για όλους τους κατοίκους» και «Την εκρηκτική ανάπτυξη του τομέα των μέσων επικοινωνίας, όπως η τηλεόραση, το τηλέφωνο, οι δορυφορικές επικοινωνίες και οι προσωπικοί υπολογιστές».

Δυστυχώς για αυτούς όμως στα μέσα του 1980 έγινε ξεκάθαρο ότι η καλωδιακή τηλεόραση δεν είχε οδηγήσει ούτε σε περισσότερες μορφές άμεσης δημοκρατίας, ούτε σε μεγαλύτερο ποσοστό συμμετοχής στα κοινά από την πλευρά του γενικού συνόλου.

B. Cyberdemocracy

Ενώ η έννοια και η φιλοσοφία της Teledemocracy ήταν άμεση απόρροια της καλωδιακής τηλεόρασης, η Cyberdemocracy δημιουργήθηκε σαν άμεσο αποτέλεσμα της εξέλιξης των δικτύων υπολογιστών. Συγκεκριμένα οι εμπειρίες των χρηστών που ήταν συνδεδεμένοι από την απαρχή αυτών των δικτύων (on-line από την δεκαετία του 1970 σε δίκτυα όπως τα EIES, USENET, Bitnet και Internet), διαμόρφωσαν τις ιδέες και τις έννοιες του Cyberdemocracy.

Ο όρος Cyberspace με την έννοια "Spaceless Place" όπου οι λέξεις, οι ανθρώπινες σχέσεις, ο πλούτος, τα δεδομένα και η θέση ορίζονταν από ανθρώπους που χρησιμοποιούσαν ηλεκτρονικούς υπολογιστές και τεχνολογίες επικοινωνιών είναι ευρέως γνωστός σήμερα. Είναι ένας όρος κλειδί, που χρησιμοποιείται σε όλο τον κόσμο από επιστήμονες, καλλιτέχνες και άλλους ανθρώπους που ασχολούνται με τις επιπτώσεις των δικτύων στους ανθρώπους και στην ίδια την ζωή. Ο όρος είχε επινοηθεί από τον συγγραφέα βιβλίων επιστημονικής φαντασίας William Gibson, στο βιβλίο του Neuromancer.

Η έννοια της Cyberdemocracy αναπτύχθηκε σε μια εποχή όπου υπήρχε η ιδιόμορφη μίξη στην Αμερικάνικη κουλτούρα με το κίνημα των Yuppies και των Hippies. Σε αυτό το στάδιο αναπτύχθηκε το όνειρο μια εικονικής κοινότητας μεταξύ του Stanford University και της Silicon Valley. Αυτή η εικονική κοινότητα που ονειρευόταν θα ήταν μια κοινότητα που θα ικανοποιούσε το όνειρο για μια πραγματική δημοκρατία. Βέβαια σε εκείνους τους καιρούς η πραγματική δημοκρατία έβρισκε υπόσταση ως μια αυτό-δυναμωμένη διοίκηση πολιτών σε συνδυασμό με το όνειρο για υλικό πλούτο και το κυνήγι της ευτυχίας, που συνήθως μετριόταν με χρήμα. Αποτέλεσμα αυτής της φιλοσοφίας ήταν μια λανθασμένη άποψη της

πολιτικής. Πλέον το κράτος φάνταζε μια πιθανή απειλή στην ατομική ελευθερία και στην προσπάθεια μεγιστοποίησης του πλούτου.

Από αυτό το σύνολο των προβλημάτων, αναπτύχθηκαν δύο πλευρές της Cyberdemocracy. Η μια, πιο συντηρητική, τονίζει την σημασία μιας ελεύθερης αγοράς και υποστήριξης της καπιταλιστικής κίνησης. Η άλλη πιο φιλελεύθερη, επικεντρώνεται στις κοινωνικές αξίες. Η πρώτη είχε την υποστήριξη του Progress and Freedom Foundation (PFF), που σαν στόχο του είχε την δημιουργία μια πολιτικής θεωρίας για τον κυβερνοχώρο. Οι συντάκτες αυτής της θεωρίας ήταν ο Alvin Toffler, ο James Keyworth, ο Esther Dyson και ο George Gilder. Βέβαια δεν αποτελεί έκπληξη ότι ο PFF ήταν μαζί με τον Newt Gingrich, που ήταν από τα πρώτα μέλη του Αμερικανικού Κογκρέσου που εγκαθίδρυσε λίστα επικοινωνίας με τους πολίτες μέσω e-mail.

Η πιο προσανατολισμένη στην κοινωνία πλευρά της Cyberdemocracy σχηματίστηκε κυρίως από τον Howard Rheingold („The Virtual Community"), αλλά οι ιδέες που υποστήριξε ήταν ευρέως διαδομένες και αποδεκτές και από πολλούς άλλους. Σε αυτή τη μορφή της Κυβερνοδημοκρατίας έχουμε την κατασκευή κοινοτήτων με σκοπό την απόκτηση νέων φίλων πολιτικών και μη.

Ενώ η Κυβερνοδημοκρατία, όπως η Τηλεδημοκρατία, καλεί για πιο άμεσες μορφές διοίκησης, τονίζει διαφορετικές πλευρές της πολιτικής συμμετοχής. Η συζήτηση και η πολιτική δραστηριότητα σαν μορφές επικοινωνίας της πολιτικής συμμετοχής είναι πολύ σημαντικές για αυτούς. Το κύριο και σημαντικότερο τους ενδιαφέρον είναι να επαναδημιουργήσουν (Εικονικά και μη) τις κοινότητες σαν μια βάση αντίθεσης στην κεντρωποιημένη μορφή δημοκρατίας.

C. *Electronic Democratization*

Σε αντίθεση με την έννοια της Τηλεδημοκρατίας και της Κυβερνοδημοκρατίας, η έννοια του Electronic Democratization δεν θέλει να καθιερώσει άμεσες μορφές δημοκρατίας αλλά επιθυμεί να ενισχύσει και να βελτιώσει την αντιπροσωπευτική δημοκρατία. "Η μεγαλύτερη δυνατότητα στην νέα τεχνολογία της πληροφορίας είναι η βελτίωση της δημοκρατίας. Και η επιτυχία του έγκειται στην ικανότητά του να ενισχύσει την παρούσα μορφή δημοκρατίας των αντιπροσώπων." (Snider, 1994). Το κύριο ενδιαφέρον του είναι η όσον το δυνατόν μεγιστοποίηση των καναλιών επικοινωνίας. Παρόλο που υπάρχει διαφορά στα μέσα που χρησιμοποιούνται για την επίτευξη αυτού του σκοπού, οι στόχοι του Electronic Democratization είναι οι ίδιοι με αυτούς της ηλεκτρονικής δημοκρατίας. "Το Electronic Democratization ορίζεται ως η επέκταση της δημοκρατίας, που πρέπει να προϋπάρχει, με νέες τεχνολογίες επικοινωνιών έτσι ώστε να είναι εφικτή η αύξηση της πολιτικής δύναμης αυτών των ανθρώπων που έχουν ρόλους κλειδιά στις πολιτικές διαδικασίες. Υποθέτουμε ότι αυτή η μορφή ηλεκτρονικής δημοκρατίας φέρνει περισσότερους ανθρώπους στην απαιτούμενη ισχύ" (Hacker/ Todino 1996)

D. Ε-Δημοκρατία

Ο όρος ε-Δημοκρατία αποτελεί μια λέξη που δημιουργήθηκε σχετικά πρόσφατα για να περιγράψει την χρήση των τεχνολογιών ηλεκτρονικής επικοινωνίας, όπως το Internet στην ενδυνάμωση των δημοκρατικών διαδικασιών. Αποτελεί μια πολιτική εξέλιξη που ακόμα βρίσκεται σε νεαρή ηλικία και αποτελεί θέμα διαλόγου ανάμεσα σε κυβερνήσεις, ομάδες και κοινωνίες σε όλο τον κόσμο. Η Ε-Δημοκρατία, αποτελεί την ολοκλήρωση της εξέλιξης της ηλεκτρονικής δημοκρατίας στις μέρες μας.

Τυπικά, οι μορφές των ενισχύσεων που προτείνει η e-democracy είναι οροθετημένες στους στόχους, του να καταστήσουν τις πολιτικές διαδικασίες ευκολότερα προσβάσιμες, καθιστώντας την συμμετοχή των πολιτών στην λήψη των αποφάσεων πιο ευρεία και πιο άμεση από ότι είναι σήμερα. Με αυτή την ευρεία συμμετοχή, επιτυγχάνεται μεγαλύτερος επηρεασμός στα σημαντικά θέματα, αυξάνεται η διαύγεια και τα λοιπά. Μέρος της e-Democracy είναι και η ηλεκτρονική ψηφοφορία.

Η πρόκληση για τις κυβερνήσεις και τις κοινότητες είναι να κατασκευάσουν εργαλεία και να προσαρμόσουν διαδικασίες που να μπορούν να ολοκληρώσουν αυτούς τους στόχους. Υπάρχουν αναρίθμητα πρακτικά και θεωρητικά θέματα που ακόμα δεν έχουν μελετηθεί και λυθεί. Υπάρχουν ακόμα και θέματα δεν έχουν γίνει πλήρη κατανοητά και αυτά αποτελούν τροχοπέδη σε πολλές από τις μέχρι τώρα έρευνες. Αλλά αυτά τα προβλήματα είναι που έχουν δώσει το κίνητρο στους ερευνητές να δοκιμάζουν καινούργιες τεχνικές και προσεγγίσεις.

Το Internet μπορεί να παρατηρηθεί ως μια πλατφόρμα και ένα μέσο παράδοσης για τα εργαλεία που βοηθούν στην λύση των γεωγραφικών περιορισμών στην άμεση δημοκρατία και η χρήση του έχει σημάνει την εποχή της Internet Δημοκρατίας. Η χρήση λοιπόν του Διαδικτύου έχει οδηγήσει στην σημερινή της μορφή την ηλεκτρονική δημοκρατία και την έχει ονομάσει E-Democracy. Η E-Democracy είναι επίσης γνωστή ως "Digital Demcoracy" και "Techno-democracy".

4. Ηλεκτρονική Διακυβέρνηση

Το διαδίκτυο σαν μία νέα μορφή επικοινωνίας προκαλεί τις παραδοσιακές μορφές διακυβέρνησης. Η νέα γενιά Χ δικτυώνεται και στους ρυθμούς της παρασύρει την προηγούμενη της γενιά. Η σύγχρονη πολιτική είναι η e-πολιτική, η σύγχρονη δημοκρατία είναι e-δημοκρατία. Τέλος η σύγχρονη διακυβέρνηση έχει πλέον μετατραπεί σε e-government.

Καθολική και αυτοτελή πρόσβαση θα πρέπει να υπάρχει ισότιμη πρόσβαση για όλους τους «πελάτες» του συστήματος. Δεν είναι απαραίτητο μάλιστα να γνωρίζουν πώς είναι οργανωμένη η δημόσια διοίκηση, τι κάνει η κάθε συγκεκριμένη υπηρεσία ή ποιοι απ' τους υπαλλήλους χειρίζονται τα θέματα που τους αφορούν.

Α. Χαρακτηριστικά E-Government

i. Πολλαπλά κανάλια επικοινωνίας

Ο «πελάτης» θα πρέπει να μπορεί να επιλέξει αν θα χρησιμοποιήσει τις υπηρεσίες της ηλεκτρονικής διακυβέρνησης και, αν ναι, με ποιον από τους δυνατούς τρόπους (Διαδίκτυο, κέντρα τηλεφωνικής εξυπηρέτησης, αμφίδρομη τηλεόραση)

Εξυπηρέτηση ανεξαρτήτως τόπου και χρόνου – καθολική υπηρεσία

Όλες οι υπηρεσίες μπορούν να 'μετασχηματιστούν' σε ηλεκτρονικές θα πρέπει να είναι διαθέσιμες ανεξαρτήτως τόπου και χρόνου, στη μορφή καθολικής υπηρεσίας. Παράλληλα θα πρέπει να προωθηθεί το κατάλληλο νομικό πλαίσιο όπου είναι αναγκαίο ειδικά για την εξακρίβωση ταυτοπροσωπίας.

ii. Ολοκληρωμένες υπηρεσίες – Ομογενή υποδομή

Οι υπηρεσίες ηλεκτρονικής διακυβέρνησης θα πρέπει να βασίζονται σε μία ολοκληρωμένη πλατφόρμα παροχής ηλεκτρονικών υπηρεσιών. Η πλατφόρμα θα πρέπει να διαθέτει 'κοινές' διαδικασίες και τεχνολογίες που μπορούν να αλληλεπιδρούν μεταξύ τους με ασφαλή και ομογενοποιημένο τρόπο. Οι εφαρμογές λογισμικού και η επεξεργασία δεδομένων θα πρέπει να υποστηρίζουν την ενιαία πρόσβαση στις υπηρεσίες πάνω σε κοινά πρωτόκολλα επικοινωνίας και πρότυπα συστημάτων μεταξύ των διαφόρων υπηρεσιών. Σκοπός είναι η αποφυγή της πολλαπλής επεξεργασίας και συλλογής των ίδιων δεδομένων και η χρήση κοινών λύσεων, π.χ., σε θέματα ηλεκτρονικών πληρωμών και διαγωνισμών. Οι υπηρεσίες θα πρέπει επίσης να παρουσιάζονται με ένα ομογενοποιημένο τρόπο 'προς τα έξω'. Η διασταύρωση της πληροφορίας είναι βασικό σκέλος των ολοκληρωμένων υπηρεσιών και οδηγεί στην ελαχιστοποίηση του κόστους. Για το σκοπό αυτό, θα πρέπει να υπάρχει ενοποιημένη προσέγγιση, που να περιλαμβάνει και τα όποια νομικά και διοικητικά θέματα αφορούν στην επεξεργασία δεδομένων προσωπικού χαρακτήρα.

iii. Εύκολη πρόσβαση στην πληροφορία – αμεσότητα στην διαβίβαση παραπόνων

Οι νέες τεχνολογίες και η αξιοποίησή τους στις σχέσεις με την δημόσια διοίκηση θα προσφέρουν ευκολότερη, πιο αξιόπιστη φερέγγυα πρόσβαση στις διοικητικές πληροφορίες, καθώς και ευκολία στην αποστολή και εξυπηρέτηση παραπόνων. Προς το σκοπό αυτό είναι απαραίτητο ένα πολυεπίπεδο σύστημα επικοινωνίας που βασίζεται σε μία 'Πύλη Ηλεκτρονικής Διακυβέρνησης (Government Portal)' και σ' ένα ολοκληρωμένο πλαίσιο υπηρεσιών με συμμετοχή όλων των φορέων του ευρύτερου δημόσιου τομέα. Πιο συγκεκριμένα, θα πρέπει να αναπτυχθούν πρότυπα επικοινωνίας που θα επιτρέπουν στις ίδιες υπηρεσίες και πληροφορίες να είναι διαθέσιμες μέσα απ' τα ίδια κανάλια. Τα πρότυπα θα πρέπει να περιλαμβάνουν τρόπους ταξινόμησης των υπηρεσιών και πληροφοριών με τέτοιο τρόπο, ώστε να είναι προσπελάσιμες από άτομα με διαφορετικές δεξιότητες ή γνώσεις. Είναι απαραίτητο, να καταγράφονται συνεχώς οι ανάγκες των χρηστών των υπηρεσιών και η ικανοποίησή τους αναφορικά με την ευκολία πρόσβασης. Είναι επίσης σημαντικό, να καταγράφονται παράμετροι όπως χρόνος, τόπος, κανάλι επικοινωνίας, κόστος, δεξιότητες κ.τ.λ. ώστε να υπάρχει απόλυτη βεβαιότητα ότι οι υπηρεσίες καλύπτουν τις ανάγκες των χρηστών.

iv. Συμμετοχή – Ανοιχτή Επικοινωνία

Με τη βοήθεια της νέας τεχνολογίας θα επιτρέπεται συνεχής επικοινωνία και διαβούλευση μεταξύ κυβέρνησης, πολιτών και επιχειρήσεων. Έτσι οι μεν πολίτες θα είναι καλύτερα πληροφορημένοι και συνεπώς θα έχουν νέες δυνατότητες συμμετοχής στα κοινά, οι δε επιχειρήσεις θα μπορούν να αναπτυχθούν χάρη στη γρηγορότερη και αξιόπιστη ενημέρωση. Με τη δημοσιοποίηση των πληροφοριών που αφορούν στον πολίτη και τις επιχειρήσεις μέσω του διαδικτύου ή της αμφίδρομης τηλεόρασης, επιτυγχάνεται η ευκολία στην ανεύρεση των σχετικών πληροφοριών και παράλληλα υποστηρίζεται το βασικό δικαίωμα πρόσβασης στην πληροφόρηση. Επίσης θα μπορούν να αναπτυχθούν συστήματα που θα δέχονται και θα διεκπεραιώνουν αιτήματα και παράπονα για το κοινό, ενώ θα προσφέρουν συμβουλευτικές υπηρεσίες τόσο σε πολίτες και επιχειρήσεις όσο και ανάμεσα σε υπηρεσίες και τους χρήστες του συστήματος.

v. Ταυτοπροσωπία (Authentication) και προστασία των προσωπικών δεδομένων(privacy)

Με τη βοήθεια των σύγχρονων τεχνολογιών ασφαλείας είναι δυνατή η αυξημένη προστασία της επιχειρηματικής πληροφορίας και των προσωπικών δεδομένων. Υπάρχουν πολλές μέθοδοι όπως, π.χ., οι βιομετρικές, οι οποίες στηρίζονται σε βιολογικά χαρακτηριστικά που επιτρέπουν την ταυτοποίηση των χρηστών.

Αυτές οι ιδιότητες κληρονομήθηκαν από την ηλεκτρονική δημοκρατία στην ηλεκτρονική διακυβέρνηση. Μέρος της ηλεκτρονικής διακυβέρνησης αποτελεί φυσικά και η ηλεκτρονική ψηφοφορία. Με τα συστήματα που κάνουν πραγματικότητα την ηλεκτρονική ψηφοφορία θα ασχοληθούμε.

5. Ηλεκτρονική ψηφοφορία

Όπως αναφέραμε προηγουμένως η αρχαία Αθηναϊκή Δημοκρατία αποτέλεσε την πρώτη και καλύτερα εφαρμοσμένη μορφή δημοκρατίας. Σε εκείνη την δημοκρατία όλοι οι πολίτες συμμετείχαν στα κοινά με άμεσο τρόπο. Όλες οι αποφάσεις στην διοίκηση και την διακυβέρνηση της πολιτείας ήταν άμεσο αποτέλεσμα της θέλησής τους. Μπορεί βέβαια αυτή η μορφή δημοκρατίας να είναι ιδανική, αλλά μπορεί να είναι αποτελεσματική μόνο όταν το σύνολο των πολιτών είναι μικρό. Αυτό το μοντέλο της άμεσης δημοκρατίας παύει πλέον να είναι αποτελεσματικό όταν το πλήθος των πολιτών είναι μεγάλο και γεωγραφικά διασκορπισμένο. Η εξέλιξη αυτής της μορφής δημοκρατίας ως αποτέλεσμα της διεύρυνσης του πλήθους των πολιτών οδήγησε στο σύγχρονο πλέον μοντέλο της αντιπροσωπευτικής δημοκρατίας. Εδώ η συμμετοχή των πολιτών επιτυγχάνεται μέσω των εκλεγμένων αντιπροσώπων του. Το θέμα της εκλογής αυτής των αντιπροσώπων, αποτελεί σημαντικό παράγοντα στην ορθή εφαρμογή της δημοκρατίας. Παρατηρώντας το σύστημα της Αντιπροσωπευτικής Δημοκρατίας, μπορούμε να συμπεράνουμε ότι είναι αποτελεσματικό και ορθό στην λογική του. Παρ' όλα αυτά όμως πολλές φορές η προσφυγή στην κοινή γνώμη είναι απαραίτητη σε ορισμένα θέματα υψίστης σημασίας. Αυτή η ανάγκη είναι που καθιστά τα δημοψηφίσματα και τις δημοσκοπήσεις κρίσιμες λειτουργίες και εργαλεία στη σωστή λειτουργίας της σημερινής αντιπροσωπευτικής δημοκρατίας. Τα δημοψηφίσματα και οι δημοσκοπήσεις έχουν δύο ρόλους. Αρχικά αποτελούν το κύριο μέσο μεταβίβασης της εξουσίας από τους πολίτες στους αντιπροσώπους τους και επιπροσθέτως είναι ικανά να διασφαλίσουν την απαιτούμενη εμπιστοσύνη στην κυβέρνηση και τις επιλογές της. Οι δύο αυτοί ρόλοι επαγωγικά οδηγούν στο συμπέρασμα ότι τα δημοψηφίσματα και οι δημοσκοπήσεις οδηγούν στην ενδυνάμωση της δημοκρατίας.

Μέχρι τις πολιτικές εκλογές των ΗΠΑ το 2000, τα συστήματα ψηφοφορίας αποτελούσαν αντικείμενο ενδιαφέροντος των πολιτικών και υπηρεσιακών στελεχών που ήταν υπεύθυνοι για την διεξαγωγή των εκλογικών διαδικασιών. Από τότε όμως το ενδιαφέρον πέρασε σε όλο τον κόσμο και κυρίως στο θέμα της διεξαγωγής τους. Αυτή ήταν η πρώτη φορά που χρησιμοποιήθηκαν συστήματα ηλεκτρονικής ψηφοφορίας σε εκλογική διαδικασία κράτους.

Σύστημα ηλεκτρονικής ψηφοφορίας είναι το ψηφιακό σύστημα το οποίο χρησιμοποιείται για την διεκπεραίωση μιας ηλεκτρονικής ψηφοφορίας. Σκοπός του είναι η μετάδοση της ηλεκτρονικής ψήφου από τους ψηφοφόρους στους αρμόδιους εκλογικούς φορείς.

Και η χρήση των συστημάτων αυτών έδωσε το έναυσμα για μεγάλες και μακροχρόνιες συζητήσεις και έρευνες στις κοινότητες των επιστημόνων. Οι δυνατότητες αλλά και οι περιορισμοί που προκύπτουν από ένα τέτοιο σύστημα είναι το κύριο θέμα έρευνας στις μέρες μας. Οι ερευνητές πρέπει να αναπτύξουν συστήματα που να υπερπηδούν τα προβλήματα που εντάσσονται από την χρήση των σύγχρονων τεχνολογιών στον τομέα της ψηφοφορίας. Ένα από τα κύρια ζητήματα που απασχολούν την διεθνή επιστημονική κοινότητα είναι η ανάπτυξη πιο αξιόπιστων συστημάτων ηλεκτρονικής ψηφοφορίας, συστημάτων που θα είναι φιλικότερα στον χρήστη και όσον το δυνατόν λιγότερο δαπανηρά. Επιπλέον ασχολούνται με το ποιες είναι οι κύριες απαιτήσεις για αυτά τα συστήματα και ποιες είναι οι νομικές και συνταγματικές ρυθμίσεις που απορρέουν από την ανάπτυξη

τέτοιων εφαρμογών. Τέλος μελετώνται οι διάφορες τεχνικές που θα πρέπει να χρησιμοποιηθούν ώστε η χρήση των συστημάτων ηλεκτρονικής ψηφοφορίας να αυξήσει την συμμετοχή των πολιτών στις εκλογές.

Ο σκοπός της ηλεκτρονικής δημοκρατίας και κατ' επέκταση των συστημάτων ηλεκτρονικής ψηφοφορίας, όπως έχουμε αναφέρει είναι η επίτευξη της συμμετοχής των πολιτών στα κοινά σε μεγαλύτερο βαθμό από την παρούσα κατάσταση. Τα τελευταία χρόνια σε όλες σχεδόν τις σύγχρονες δημοκρατίες παρατηρείται μια πτωτική τάση στη συμμετοχή των πολιτών στην εκλογική διαδικασία. Χρόνο με τον χρόνο, όλο και λιγότεροι άνθρωποι προσέρχονται στις κάλπες κατά την διάρκεια της εκλογικής διαδικασίας των αντιπροσώπων. Αυτή η κατάσταση ήταν το κίνητρο που χρειαζόταν για να επέρθει η αύξηση του ενδιαφέροντος για την παροχή της δυνατότητας ψηφοφορίας μέσω του διαδικτύου.

Στην αρχή, η ηλεκτρονική ψηφοφορία μέσω Διαδικτύου θεωρήθηκε μια ακόμη εφαρμογή του Διαδικτύου, παρόμοια με αυτές του ηλεκτρονικού εμπορίου και της ηλεκτρονικής διακυβέρνησης. Η προσέγγιση όμως αυτή δεν είναι καθόλου σωστή, αφού τα συστήματα ηλεκτρονικής ψηφοφορίας μέσω Διαδικτύου πρέπει να ικανοποιούν υψηλές προδιαγραφές ασφάλειας.

Α. Πλεονεκτήματα Ηλεκτρονικής ψηφοφορίας

Πληθώρα επιστημονικών αλλά και δημοσιογραφικών κειμένων αναφέρονται στις δυνατότητες των συστημάτων ηλεκτρονικής ψηφοφορίας και τις προοπτικές που ανοίγουν για μια περισσότερο συμμετοχική δημοκρατία, αλλά ταυτόχρονα αναδεικνύουν τους περιορισμούς τους, τους κινδύνους και τις αδυναμίες που αντιμετωπίζουν, όπως επίσης και τους κοινωνικούς προβληματισμούς που εγείρουν. Αναφερόμενοι στα πλεονεκτήματα που προσφέρει η ηλεκτρονική ψηφοφορία σε σύγκριση με την παραδοσιακή μορφή ψηφοφορίας μπορούμε να αναφέρουμε τα ακόλουθα:

i. Συμμετοχή

Η έρευνα για τις βασισμένες στο Διαδίκτυο εκλογές τροφοδοτείται από την ανησυχία των δημοσίων λειτουργών ως προς τη συνεχή μείωση του αριθμού των συμμετεχόντων στις ψηφοφορίες – φαινόμενο ιδιαίτερα έντονο στις ΗΠΑ. Ειδικά στις μικρές ηλικίες η συμμετοχή στις ΗΠΑ κυμαίνεται στο 15%. Για τις ηλικίες αυτές οι ψηφοφορίες Online κρίνονται σαφώς πιο ελκυστικές. Η σκέψη λοιπόν που κυριαρχεί είναι ότι αν υπάρχει κάτι που μπορεί να ενθαρρύνει τη συμμετοχή των πολιτών στα κοινά, πρέπει να αξιοποιηθεί και για τη βασική αυτή λειτουργία της δημοκρατίας

ii. Ευκολία

Ίσως το σημαντικότερο επιχείρημα υπέρ των ηλεκτρονικών ψηφοφοριών είναι ο παράγοντας ευκολίας. Πράγματι, η ευκολία ενθαρρύνει τη συμμετοχή, η οποία μπορεί να οδηγήσει σε ένα ισχυρότερο εκλογικό σώμα. Είναι σαφές ότι με αυτό τον τρόπο λύνεται και το θέμα των ετεροδημοτών, το οποίο έχει απασχολήσει αρκετά την ελληνική κοινωνία, αφού δεν απαιτείται πλέον οποιαδήποτε μετακίνηση. Προφανώς, η λύση της ηλεκτρονικής ψηφοφορίας θα οδηγήσει στη μείωση των κρατικών δαπανών για τις εκλογές και στην εξοικονόμηση εργατοωρών λόγω των λιγότερων μετακινήσεων.

iii. Ενημέρωση

Στο κρισιμότερο σημείο της διαδικασίας των εκλογών, όταν η ψήφος ρίχνεται πραγματικά στην κάλπη, οι ψηφοφόροι έχουν παραδοσιακά ελάχιστες ή καθόλου πληροφορίες για τους υποψήφιους ή τα ζητήματα που τίθενται στην κρίση τους. Δεν είναι λίγοι εκείνοι που την ώρα της εκλογής ψηφίζουν ονόματα που απλώς θυμούνται ευκολότερα, ή πρόσωπα που έχουν προβληθεί περισσότερο από τα ΜΜΕ. Η ηλεκτρονική ψηφοφορία είναι σε θέση να προσφέρει στους ψηφοφόρους επίσημα εγκεκριμένες πληροφορίες για κάθε υποψήφιο, τη στιγμή ακριβώς που ψηφίζουν. Το γεγονός και μόνο αυτό θα απαλλάξει τον τρόπο διεξαγωγής των προεκλογικών εκστρατειών.

iv. Αποδοτικότητα διαδικασίας

Ένα ακόμα επιχείρημα υπέρ της ηλεκτρονικής ψηφοφορίας, που αφορά στο διαχειριστικό μέρος της όλης διαδικασίας, είναι ότι αυτή μπορεί να είναι ο γρηγορότερος, φθηνότερος και αποδοτικότερος τρόπος να διενεργηθούν οι εκλογές και να καταμετρηθούν οι ψήφοι. Για να φτάσουν όμως στο μέγιστο η αποδοτικότητα και η οικονομία, πρέπει να μπορεί να διεξαχθεί η ψηφοφορία χωρίς τη χρήση ιδιαίτερου υλικού και χωρίς να απαιτείται η οποιαδήποτε μετακίνηση των ψηφοφόρων.

v. Πρόσβαση

Ένα από τα μεγαλύτερα επιχειρήματα υπέρ της ηλεκτρονικής ψηφοφορίας είναι ότι επιτρέπει την πρόσβαση στη δημοκρατική διαδικασία σε πολύ περισσότερους πολίτες, ακόμη και σε ειδικές ομάδες του πληθυσμού που για διαφόρους λόγους τώρα αποκλείονται

Β. Προβλήματα εφαρμογής της Ηλεκτρονικής Ψηφοφορίας

Η μεγαλύτερη ανησυχία των υπευθύνων μιας ηλεκτρονικής ψηφοφορίας εντοπίζεται στον τομέα της ασφάλειας, λόγω των τρομερών επιπτώσεων που μπορεί να έχει μία επιτυχής επίθεση σε ένα τέτοιο σύστημα. Για παράδειγμα, η πιθανότητα κακόβουλης επίθεσης εναντίον υπολογιστικών συστημάτων συνδεδεμένων στο Διαδίκτυο που χρησιμοποιούνται για την ηλεκτρονική ψηφοφορία, δεν πρέπει να αγνοηθεί. Μια τέτοια επίθεση θα μπορούσε να οδηγήσει σε άρνηση εξυπηρέτησης, δηλαδή στην αδυναμία σύνδεσης στο δίκτυο, με τελικό αποτέλεσμα την αδυναμία άσκησης του εκλογικού δικαιώματος. Θα μπορούσε, επίσης, να οδηγήσει στη μαζική εισαγωγή στην ηλεκτρονική κάλπη τροποποιημένων ψηφοδελτίων ή στη μαζική αποκάλυψη της ψήφου ομάδων πολιτών. Αρκετοί πιστεύουν ότι παρά τις προκλήσεις αυτές, είναι τεχνολογικά δυνατό να κατασκευαστεί σύστημα ηλεκτρονικής ψηφοφορίας που να είναι τουλάχιστον όσο ασφαλές είναι και τα ήδη υπάρχοντα συστήματα επιστολικής ψήφου. Το σημαντικότερο βήμα προκειμένου ένα σύστημα ψηφοφορίας να καταστεί ασφαλές, είναι η επαλήθευση των μεμονωμένων ψηφοφόρων. Πρέπει να είμαστε βέβαιοι ότι οι ψηφοφόροι είναι πραγματικοί και ότι κάθε πρόσωπο δίνει μία ψήφο και αυτό είναι πράγματι ένα δύσκολο τεχνικό ζήτημα. Σύμφωνα με αρκετούς εμπειρογνώμονες η ανάπλαση της εξαιρετικά σύνθετης διαδικασίας εκλογής σε έναν υπολογιστή είναι ένα από τα δυσκολότερα προγραμματιστικά προβλήματα και μία από τις σημαντικότερες κρυπτογραφικές προκλήσεις. Σε κάθε περίπτωση, οι περισσότεροι ειδικοί στον τομέα συμφωνούν ότι πρέπει να επιτευχθεί η κατάλληλη ισορροπία μεταξύ ασφάλειας, προσβασιμότητας και ευκολίας χρήσης πριν ένα σύστημα ηλεκτρονικής ψηφοφορίας επιτραπεί να χρησιμοποιηθεί σε πραγματικές συνθήκες εκλογής. Πέρα όμως από τους τεχνολογικούς κινδύνους, κανείς δεν επιτρέπεται να αγνοεί κινδύνους κοινωνικής ή/και πολιτικής φύσης, όπως – για παράδειγμα – τον κίνδυνο ουσιαστικού αποκλεισμού από την εκλογική διαδικασία ομάδων πολιτών που αντιμετωπίζουν δυσχέρειες ή εμφανίζουν απροθυμία χρήσης της τεχνολογίας, αν οι εκλογές μέσω Διαδικτύου αποτελούν τη μοναδική επιλογή συμμετοχής.

Το κατά πόσο όλοι οι κίνδυνοι και οι αδυναμίες που σχετίζονται με τα συστήματα ηλεκτρονικής ψηφοφορίας μπορούν να αντιμετωπιστούν με την ήδη υπάρχουσα και την αναπτυσσόμενη τεχνολογία είναι συζητήσιμο. Άλλωστε, δεν θα ήταν δίκαιο να απαιτεί κανείς ένα σύστημα ηλεκτρονικής ψηφοφορίας να είναι περισσότερο ασφαλές από το χειρόγραφο του ισοδύναμο. Και βέβαια, η τελική απόφαση για τη χρήση ή μη μιας συγκεκριμένης τεχνολογίας θα πρέπει να συνυπολογίζει τόσο τους ενδεχόμενους κινδύνους, όσο και τα αναμενόμενα οφέλη.

Φαίνεται πάντως ότι, αν και η ψηφοφορία μέσω του Διαδικτύου αναμφίβολα δίνει στους ψηφοφόρους άνεση και ευκολία προσβασιμότητας, επιτρέποντας τους να ψηφίζουν από οποιοδήποτε σημείο υπάρχει πρόσβαση στο Διαδίκτυο, αυτός ο τρόπος ψηφοφορίας παρουσιάζει και σημαντικά προβλήματα ασφάλειας και εγείρει κοινωνικά ζητήματα που πρέπει να αντιμετωπιστούν αποτελεσματικά σε τεχνολογικό, οργανωτικό, επιχειρησιακό, αλλά και πολιτικό επίπεδο, πριν αποτελέσει εναλλακτική επιλογή για τη διεξαγωγή πραγματικών εκλογικών αναμετρήσεων.

Για τον λόγω αυτό ιδιωτικές και δημόσιες οργανώσεις έχουν στρέψει ένα τεράστιο ποσό ενέργειας και πόρων στην υπερνίκηση των τεχνικών προκλήσεων της

ασφαλούς ηλεκτρονικής ψηφοφορίας. Διάφορες επιχειρήσεις έχουν αναπτύξει ήδη πειραματικά συστήματα και τα εξετάζουν για χρήση στις δημόσιες εκλογές.

Η ασφάλεια, λόγω της πολυπλοκότητας αλλά και της σοβαρότητας του εγχειρήματος, απαιτεί ιδιαίτερη προσοχή. Οι χρήστες-ψηφοφόροι θα ψηφίζουν μέσω Διαδικτύου μόνο όταν είναι βέβαιοι ότι οι προσωπικές πληροφορίες τους είναι ασφαλείς. Η λύση είναι να εφαρμοστεί μια πλήρης «υποδομή εμπιστοσύνης», παρόμοια με αυτήν που χρησιμοποιείται ήδη σε κάποιο βαθμό στις εφαρμογές ηλεκτρονικού εμπορίου.

Το σύστημα κρυπτογραφίας PKI και η τεχνολογία ηλεκτρονικών υπογραφών, που εφαρμόζονται στο SSL (Secure Sockets Layer), παρέχουν την επικύρωση, την εγγύηση της ακεραιότητας των δεδομένων και την απαραίτητη μυστικότητα. Απαιτείται όμως περαιτέρω έρευνα, έτσι ώστε να είμαστε όσο το δυνατόν πιο σίγουροι πριν φτάσουμε στο σημείο να εφαρμόσουμε ένα τέτοιο σύστημα. Σε κάθε περίπτωση, η Κοινωνία της Πληροφορίας επηρεάζει κάθε πλευρά της καθημερινής μας ζωής και αλλάζει τον τρόπο που ζούμε, που εργαζόμαστε, που επικοινωνούμε, που διασκεδάζουμε, που διοικούμε και διοικούμαστε.

Είναι, λοιπόν, καθαρά θέμα χρόνου πριν αλλάξει και τον τρόπο που συμμετέχουμε στα κοινά, που διαμορφώνουμε τις πολιτικές μας απόψεις, που συμβάλλουμε στη διαμόρφωση των απόψεων των άλλων, που επηρεάζουμε τις πολιτικές αποφάσεις που μας αφορούν, που συμμετέχουμε στα κοινά, που εκλέγουμε τους αντιπροσώπους μας και, τελικά, τον τρόπο που βιώνουμε τη δημοκρατία.

C. Ιδιότητες Συστήματος Ηλεκτρονικής ψηφοφορίας

i. Αναφορά και περιγραφή των ιδιοτήτων

Ένα σύστημα ηλεκτρονικής ψηφοφορίας είναι ένα εξαιρετικά πολύπλοκο σύστημα, με αρκετές παραμέτρους, σε όλους σχεδόν τους τομείς της τεχνολογίας των υπολογιστών. Επαγωγικά λοιπόν όταν σχεδιάζεται ένα τέτοιο σύστημα, πρέπει να λαμβάνονται υπ' όψιν όλες οι ιδιότητες που πρέπει να έχει για να μπορεί να χαρακτηριστεί εφαρμόσιμο. Οι ιδιότητες αυτές, μπορούν να ταξινομηθούν σε κατηγορίες, ανάλογα με την ανάγκη που εξυπηρετούν. Οι έξι σημαντικότερες ιδιότητες που πρέπει να διασφαλίζει ένα σύστημα είναι :

Ακρίβεια (Accuracy): Ένα σύστημα έχει διασφαλίσει την ιδιότητα της Ακρίβειας όταν :

1. Δεν είναι εφικτό για μια ψήφο που έχει εισαχθεί στο σύστημα να τροποποιηθεί.
2. Δεν είναι εφικτό για μια ψήφο που έχει εισαχθεί στο σύστημα να διαγραφεί από την τελική καταμέτρηση των ψήφων και
3. Δεν είναι εφικτό για μια άκυρη ψήφο να καταμετρηθεί στην τελική ψηφοφορία.

Δημοκρατία (Democratic): Ένα σύστημα μπορεί να χαρακτηριστεί δημοκρατικό όταν:

1. Μπορεί και διαχωρίζει τους πολίτες πρό της ψηφοφορίας και επιτρέπει σε αυτούς που είναι νόμιμοι ψηφοφόροι να προβούν στην ακόλουθη διαδικασία ψηφοφορίας, αποκλείοντας τους άλλους.
2. Έχει ενσωματωμένο μηχανισμό που διασφαλίζει ότι όλοι οι νόμιμοι ψηφοφόροι μπορούν να ψηφίσουν μόνο μια φορά. Αυτό προλαμβάνει, τις πιθανές προσπάθειες κάποιων να ψηφίσουν δύο και περισσότερες φορές.

Μυστικότητα (Privacy): Η ιδιότητα της μυστικότητας προκύπτει όταν

1. Κανένας απολύτως παράγοντας, συμπεριλαμβανομένης της εκλεκτικής αρχής και του ιδίου του ψηφοφόρου δεν έχει την δυνατότητα να συνδέσει κάποια ψήφο με το άτομο που την έκανε.
2. Κανένας ψηφοφόρος δεν μπορεί να αποδείξει με κανέναν τρόπο ότι ψήφισε με συγκεκριμένο τρόπο
3. Όλες οι ψήφοι παραμένουν μυστικές από όλους μέχρι την λήξη της ψηφοφορίας και το στάδιο της δημοσίευσης των αποτελεσμάτων.

Επαληθεσιμότητα (Verifiability): Ένα σύστημα ηλεκτρονικής ψηφοφορίας είναι επαληθεύσιμο όταν

1. Ο κάθε ψηφοφόρος μεμονωμένα μπορεί να επαληθεύσει ότι η ψήφος του έχει καταμετρηθεί σωστά.

Ευκολία (Convenient): Ένα σύστημα χαρακτηρίζεται εύκολο όταν είναι σχεδιασμένο με τέτοιο τρόπο ώστε:

1. Να είναι εύκολη η χρήση του από όλους τους ψηφοφόρους, ανεξαρτήτως των προσωπικών τους γνώσεων και ικανοτήτων
2. Να μην απαιτεί από τους ψηφοφόρους να κατέχουν ειδικά, ούτε καν ισχυρά μηχανήματα για να το χρησιμοποιήσουν

Ευελιξία (Flexibility): Ένα σύστημα είναι ευέλικτο όταν επιτρέπει περισσότερες από μια ψηφοφορίες.

Πέραν των πέντε αυτών ιδιοτήτων, που χαρακτηρίζονται ως οι σημαντικότερες στην υλοποίηση ενός συστήματος ηλεκτρονικής ψηφοφορίας, υπάρχουν και ορισμένες ακόμα που βρίσκονται υπό συζήτηση. Αυτές οι ιδιότητες έρχονται να προσθέσουν ή τροποποιήσουν ορισμένα θέματα που καθημερινά προκύπτουν στην ηλεκτρονική ψηφοφορία. Οι ιδιότητες αυτές είναι:

Παραλληλισμός με το παραδοσιακό σύστημα: Ένα σύστημα ηλεκτρονικής ψηφοφορίας είναι συνταγματικά ισότιμο με το παραδοσιακό σύστημα, όταν σε κάθε ψηφοφορία διασφαλίζεται ότι το σύστημα χρησιμοποιεί τις ίδιες διαδικασίες που θα χρησιμοποιούνταν στην ψηφοφορία εάν γινόταν, σε εκείνο τον τόπο, με τον παραδοσιακό τρόπο.

Ισοτιμία των υποψηφίων: Ένα σύστημα ηλεκτρονικής ψηφοφορίας θα πρέπει να είναι σχεδιασμένο έτσι ώστε να μην επιτρέπει κάποια διαφορετική μεταχείριση μεταξύ των υποψηφίων κατά τη διάρκεια της ψηφοφορίας.

Μη ύπαρξη προπαγάνδας στην ηλεκτρονική ψηφοφορία: Το σύστημα θα πρέπει να μην επιτρέπει κάποια προβολή κάποιας εισόδου από τους υποψήφιους προς τους ψηφοφόρους.

Μη ύπαρξη καταγραφής- καταμέτρησης κατά την ώρα διεξαγωγής της ψηφοφορίας:

Το σύστημα θα πρέπει να διαχωρίζει τις φάσεις της ψηφοφορίας και της καταμέτρησης.

ii. Οι ιδιότητες ως μηχανισμοί διασφάλισης της ηλεκτρονικής ψηφοφορίας

Όπως έχουμε αναφέρει πρωτίστως οι ιδιότητες σε ένα τέτοιο σύστημα είναι η βάση πάνω στην οποία στηρίζεται η ύπαρξη της απαιτούμενης εμπιστοσύνης στο σύστημα. Για να μπορέσει να χρησιμοποιηθεί ένα σύστημα ηλεκτρονικής ψηφοφορίας πρέπει πρώτα να κερδίσει την εμπιστοσύνη των ανθρώπων που θα το χρησιμοποιήσουν.

Ο σχεδιασμός και η υλοποίησή του θα πρέπει να έχουν γίνει με τη χρήση των κανόνων και λεπτομερειών που θα καθιστούν αυτή την ύπαρξη εμπιστοσύνης απλά θέμα χρόνου, μέχρι την πρώτη του χρήση. Οι ιδιότητες αυτές, όταν έχουν εφαρμοστεί στο έπακρο τους δίνουν στο σύστημα, τέτοια μορφή που είναι αποδεκτή από τους μελλοντικούς χρήστες του, καθιστώντας το φυσική εξέλιξη του παραδοσιακού μηχανισμού ψηφοφορίας. Επαγωγικά, επειδή αυτός ο μηχανισμός είναι δοκιμασμένος, και το καινούργιο σύστημα θα ξεκινήσει από μια γερή βάση.

Για να κατανοήσουμε τον τρόπο εφαρμογής των ιδιοτήτων στο καινούργιο σύστημα ηλεκτρονικής ψηφοφορίας θα πρέπει να προβούμε σε μια ανάλυση της κάθε ιδιότητας ξεχωριστά και στον αποτέλεσμα που αποσκοπούμε, εντάσσοντας την στο σύστημά μας. Αρχικά θα αναφερθούμε στις έξι βασικές ιδιότητες.

Αρχικά έχουμε την ιδιότητα της Ακρίβειας. Αυτή ορίζει ότι δεν πρέπει κάποια ψήφος που έχει εισαχθεί να τροποποιηθεί. Είναι βασικό, να έχει σχεδιαστεί το σύστημα με τέτοιο τρόπο που εξ' ορισμού, με την εισαγωγή της ψήφου, αυτή να είναι απροσπέλαστη από όλα τα μέλη που εμπλέκονται άμεσα και έμμεσα με την διαδικασία της ψηφοφορίας. Εάν η ψήφος μπορεί να παραποιηθεί, το αποτέλεσμα της ψηφοφορίας θα είναι ψευδές και το σύστημα θα έχει αποτύχει.

Επιπλέον η ιδιότητα της Ακρίβειας θέλει να μην εφικτό μια ψήφος που έχει εισαχθεί στο σύστημα να μπορεί να αποκλειστεί από την τελική καταμέτρηση. Η ύπαρξη αυτής της ιδιότητας έχει τον ίδιο βαθμό σημαντικότητας με την προηγούμενη. Ο λόγος είναι ότι η μη καταμέτρηση κάποιας νόμιμης ψήφου στο τελικό αποτέλεσμα οδηγεί σε εσφαλμένο αποτέλεσμα. Και σε ένα σύστημα ηλεκτρονικής ψηφοφορίας όταν το αποτέλεσμα είναι εσφαλμένο ή κατά προσέγγιση, τότε το σύστημα έχει αποτύχει.

Το τρίτο και τελευταίο τμήμα που περιλαμβάνεται στην ιδιότητα της Ακρίβειας είναι η ιδιότητα του συστήματος να είναι σχεδιασμένο με τέτοιο τρόπο ώστε να μην είναι εφικτό μια άκυρη ψήφος να καταμετρηθεί στην τελική ψηφοφορία. Εάν μια ψήφος δεν έχει νόμιμη μορφή, σύμφωνα με αυτή που μπορεί να εισαχθεί στο σύστημα, τότε η προσπάθεια καταμέτρησής της θα οδηγήσεις σε εξαίρεση. Επιπλέον αν κάποια άκυρη ψήφος καταμετρηθεί στην τελική ψηφοφορία, τα αποτέλεσμα θα είναι η αλλοίωση του αποτελέσματος, γεγονός που πρέπει να αποφεύγεται.

Ακολούθως της ιδιότητας της Ακρίβειας έρχεται η ιδιότητα της Δημοκρατίας. Ένα σύστημα ηλεκτρονικής ψηφοφορίας πρέπει να είναι δημοκρατικό. Αυτό επιτυγχάνεται με δύο ιδιότητες που πρέπει να έχει το σύστημα. Κατ' αρχήν θα πρέπει το σύστημα να επιτρέπει στους νόμιμους ψηφοφόρους μόνο να ψηφίσουν. Σε κάθε εκλογή υπάρχει ένα σύνολο ανθρώπων που έχουν δικαίωμα να ψηφίσουν, ανάλογα πάντα με την περίπτωση. Ένα σύστημα ηλεκτρονικής ψηφοφορίας θα πρέπει να τηρεί μια λίστα εξουσιοδότησης ψήφου με όλους τους ψηφοφόρους, ειδάλλως όλοι όσοι επιθυμούν θα ψηφίσουν την επιλογή της αρεσκείας τους ασχέτως εάν έχουν δικαίωμα να ψηφίσουν. Και όπως είναι κατανοητό το αποτέλεσμα θα είναι διαφορετικό από το ορθό.

Σε προσθήκη με την προηγούμενη ιδιότητα ένα σύστημα ηλεκτρονικής ψηφοφορίας θεωρείται δημοκρατικό όταν πληρεί μια ακόμα ιδιότητα. Ένα σύστημα θα πρέπει να επιτρέπει στους νόμιμους ψηφοφόρους να ψηφίσουν την επιλογή τους μόνο μια φορά. Είναι δεδομένο ότι εάν δεν διασφαλίζεται αυτή η ιδιότητα, όλοι όσοι επιθυμούν την εκλογή της επιλογής τους, θα εισέλθουν στο σύστημα και θα ψηφίσουν πολλές φορές. Αυτό αψηφάει την ανάγκη για ισοτιμία σε τέτοιες σημαντικές κινήσεις όπως είναι οι εκλογές προσώπων. Έτσι καταλαβαίνουμε την αναγκαιότητα για ένα σύστημα να μπορεί να ελέγξει τους ψηφοφόρους στο θέμα του εάν ψήφισαν ή όχι σε κάθε δεδομένη στιγμή της ψηφοφορίας.

Έχοντας εξετάσει και την ιδιότητας της δημοκρατίας σε ένα σύστημα ηλεκτρονικής ψηφοφορίας θα εξετάσουμε την ιδιότητα της μυστικότητας. Αυτή η ιδιότητα αποτελείται από τρεις μηχανισμούς που πρέπει να ολοκληρώνονται σε ένα σύστημα ηλεκτρονικής ψηφοφορίας, έτσι ώστε να καθίστασαι χρησιμοποιήσιμο.

Σε πρώτο επίπεδο βρίσκεται η ιδιότητα που θέλει ένα σύστημα να μην υπάρχει τρόπος να συνδεθεί κάποια ψήφος με τον άνθρωπο που την επέλεξε. Αυτή είναι μια από τις πρωταρχικές απαιτήσεις σε ένα σύστημα όπου επιθυμούμε να υπάρχει ανωνυμία. Η ψήφος ως γνωστό είναι προσωπική επιλογή του κάθε ανθρώπου και πρέπει να είναι μυστική. Κανένας δεν πρέπει να γνωρίζει την ψήφο του άλλου, χωρίς την θέλησή του.

Δεύτερος σημαντικός παράγοντας είναι το να μην μπορεί να αποδείξει ο ψηφοφόρος με τον οποιοδήποτε τρόπο ότι ψήφισε κάτι συγκεκριμένο. Εάν αυτή η ιδιότητα δεν τηρηθεί σε ένα σύστημα ηλεκτρονικής ψηφοφορίας, τότε είναι δεδομένο ότι μια μεγάλη μερίδα ψηφοφόρων δεν θα ψηφίσει κατά συνείδηση αλλά όπως θα τους υποδειχθεί. Εφόσον θα υπάρχει η επιλογή να αποδείξει με κάποιον τρόπο ο ψηφοφόρος τι ψήφισε, θα οδηγηθούμε αναπόφευκτα σε μια κατάσταση τύπου «Ψήφισέ με και..» και το χειρότερο είναι ότι το σύστημα θα το ενθαρρύνει.

Τέλος η τεχνική που ολοκληρώνει την ιδιότητα της μυστικότητας, θέλει τις ψήφους να παραμένουν μυστικές μέχρι το πέρας της διαδικασίας ψηφοφορίας. Στην περίπτωση που δεν διασφαλίζεται αυτό θα υπάρχει μια αρχική εντύπωση, που μακροπρόθεσμα θα επηρεάζει το αποτέλεσμα. Με το να είναι φανερές οι ψήφοι σε ένα σύστημα ηλεκτρονικής ψηφοφορίας, είναι δυνατόν να αλλοιώνεται το τελικό αποτέλεσμα άδικα για μια πλευρά, εξαιτίας της εν' μέρη έκβασης της ψηφοφορίας, ή ακόμα και το αντίθετο.

Για να διασφαλίσουμε λοιπόν την ύπαρξη των ίδιων δικαιωμάτων και από τις δύο πλευρές, ως επίσης και για να προασπίσουμε μια δίκαια αναμέτρηση, είμαστε υποχρεωμένοι να διατηρήσουμε τις ψήφους μυστικές μέχρι το τέλος της ψηφοφορίας, όπου και το αποτέλεσμα θα έχει πλέον κριθεί.

Ακολούθως της μυστικότητας σε μια ψηφοφορία, έχουμε την ιδιότητα της επαληθευσιμότητας. Αυτή η ιδιότητα θέλει ανεξάρτητα τον κάθε ψηφοφόρο να μπορεί να επαληθεύσει ότι ψήφισε, και ότι η ψήφος του έχει νόμιμα καταμετρηθεί στην τελική ψηφοφορία. Εδώ έχουμε μια ειδική περίπτωση ιδιότητας όπου ο κάθε ψηφοφόρος, με κάποια τεχνική μπορεί να τοποθετεί μια ξεχωριστή σφραγίδα στην ψήφο του, ώστε αυτό μόνο να μπορεί να την αναγνωρίσει ανάμεσα σε όλες τις άλλες.

Ο ψηφοφόρος, υπογράφοντας την ψήφο του, μετά το πέρας της διαδικασίας ψηφοφορίας μπορεί να ελέγξει ότι η ψήφος του υπάρχει μέσα στις καταμετρηθήσες ψήφους. Έτσι διασφαλίζεται η σωστή καταμέτρηση των ψήφων ως επίσης και ο βαθμός εγκυρότητας της διαδικασίας, μιας και ο κάθε άνθρωπος μπορεί να επαληθεύσει την ψήφο του μετά από κάποια σχετική διαδικασία.

Μετά και από την μυστικότητα έχουμε μια ακόμα σημαντική ιδιότητα. Την ιδιότητα της ευκολίας. Σε κάθε ψηφοφορία οι άνθρωποι που έχουν δικαίωμα να

ψηφίσουν είναι συνήθως από όλες τις κοινωνικές τάξεις, από όλες τις ηλικίες και από όλα τα επίπεδα εκπαίδευσης. Είναι λογικό λοιπόν ότι δεν πρέπει να ζητήσουμε από τους ανθρώπους αυτούς να έχουν κάποια ειδική γνώση για να ασκήσουν αυτό το δικαίωμα.

Σε μια ψηφοφορία θα πρέπει να μπορούν να συμμετέχουν όλοι όσοι έχουν δικαίωμα να συμμετέχουν και να μην αποκλείονται εμμέσως άτομα, επειδή δεν μπορούν να χρησιμοποιήσουν το σύστημα. Έτσι η ιδιότητα αυτή διασφαλίζει ότι ένα σύστημα ηλεκτρονικής ψηφοφορίας θα πρέπει να έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι εύκολο στην χρήση, ανεξαρτήτως προσωπικών γνώσεων και δεξιοτήτων. Θα πρέπει να έχει δημιουργηθεί με τέτοιο τρόπο ώστε να μπορεί να χρησιμοποιηθεί και από άτομα που κατεξοχήν είναι μακριά από την τεχνολογία των υπολογιστών. Θα πρέπει να έχει γίνει πρόληψη σε τέτοιο σημείο που να υπάρχει υποστήριξη για ηλικιωμένους και άτομα με ειδικές ανάγκες. Επιπλέον στον σχεδιασμό του συστήματος θα πρέπει να έχει ληφθεί υπ' όψη ότι οι περισσότεροι από τους εν' δυνάμει ψηφοφόρους δεν θα έχουν κάποιο δυνατό υπολογιστικό σύστημα και κατά συνέπεια θα πρέπει να σχεδιάσουν ένα σύστημα, το οποίο θα απαιτεί λίγους πόρους και όχι κάτι που ορισμένοι δεν θα μπορούν καν να εκτελέσουν.

Η τελευταία από τις σημαντικότερες ιδιότητες που θα πρέπει να έχει διασφαλίσει ένα σύστημα είναι η δυνατότητα να υποστηρίζει περισσότερες από μια ψηφοφορίες τη φορά. Βέβαια, αυτό εάν και είναι επιθυμητό, η ανάγκη ύπαρξής του προκύπτει άμεσα από το περιβάλλον στο οποίο διεξάγεται η οποιαδήποτε ψηφοφορία. Αυτή η ιδιότητα βρίσκει μεγαλύτερη απήχηση σε συστήματα που διαχειρίζονται δημοσκοπήσεις και μαζικές ψηφοφορίες, συστήματα που αποτελούν κατηγορία των συστημάτων ψηφοφορίας. Σε αυτά τα συστήματα, επιθυμούμε ο ψηφοφόρος να μπορεί να επιλέξει ανάμεσα σε έναν αριθμό διαφορετικών ψηφοφοριών έτσι ώστε να μπορεί να εκφράσει την άποψή του σε μια ενιαία συνεδρία και να μην πρέπει να αναπτυχθεί μια σειρά από το ίδιο σύστημα για την κάθε μια από τις ψηφοφορίες.

Έχοντας αναλύσει τις σημαντικότερες από τις ιδιότητες που θα πρέπει να διασφαλίζει ένα σύστημα ηλεκτρονικής ψηφοφορίας, για να θεωρείται αξιόπιστο θα περάσουμε στην συζήτηση των παρελκόμενων ιδιοτήτων που και αυτές παρέχουν αξιοπιστία σε ένα τέτοιο σύστημα. Η μόνη τους διαφορά από τις λεγόμενες θεμελιακές είναι ότι αυτές τελούν ακόμα σε θεωρητικό επίπεδο και βρίσκονται υπό συζήτηση. Βέβαια σε ένα σύστημα, τέτοιων δυνατοτήτων και προσφοράς είναι κατανοητό ότι κάθε νέα ιδιότητα είναι απαραίτητη για την καθιέρωση του επιθυμητού και πολλές φορές αναγκαίου επιπέδου εμπιστοσύνης ανάμεσα σε κατασκευαστές και χρήστες.

Η πρώτη από αυτές τις ιδιότητες είναι η ιδιότητα του παραλληλισμού με το παραδοσιακό σύστημα. Ένα σύστημα ηλεκτρονικής ψηφοφορίας ανάλογα με το μέρος και την περίσταση, την οποία καλείται να υποστηρίξει, έχει κάποιες ειδικές τροποποιήσεις που το καθιστούν κατάλληλο για τη δεδομένη περίσταση. Το παραδοσιακό σύστημα ψηφοφορίας σε κάθε μέρος και για κάθε περίσταση αποτελεί μια λύση δοκιμασμένη και στερεότυπη. Η προσπάθεια μεταφοράς αυτής της μεθόδου σε ηλεκτρονική μορφή δεν είναι η καλύτερη δυνατή λύση και πιθανότατα δεν θα ευδοκιμήσει μιας και ορισμένα θέματα που θεωρούνται αυτονόητα και η λύση τους είναι εφικτή στα παραδοσιακά συστήματα αποτελούν ανυπέρβλητα εμπόδια στην ηλεκτρονική απόδοση του ίδιου συστήματος. Βέβαια ανάλογα με την περίσταση η διατήρηση ορισμένων από τις διαδικασίες αποτελεί μονόδρομο σαν επιλογή, μιας και είναι δοκιμασμένες και επιτυχημένες.

Ακολούθως έρχεται η ιδιότητα της ισοτιμίας των υποψηφίων. Ένα σύστημα ηλεκτρονικής ψηφοφορίας θα πρέπει να είναι ανεξάρτητο των προσωπικών απόψεων

των κατασκευαστών και των υπευθύνων του. Θα πρέπει να υπάρχουν μηχανισμοί που θα διασφαλίζουν ότι όλοι οι υποψήφιοι θα έχουν ίσα δικαιώματα, ως προς την παρουσίασή τους στους ψηφοφόρους κατά την διάρκεια της ψηφοφορίας. Παραδειγματική εφαρμογή αυτής της ιδιότητας, αποτελεί η επιλογή να παρουσιάζονται όλοι οι υποψήφιοι με αλφαβητική ιεραρχία, με την ίδια γραμματοσειρά και το ίδιο μέγεθος γραμμάτων. Η ιδιότητα αυτή, εκτός από την ισότητα στους υποψηφίους, έχει στόχο να αναβαθμίσει και το επίπεδο εμπιστοσύνης ανάμεσα στο σύστημα και στους ψηφοφόρους. Όταν οι ψηφοφόροι βλέπουν ότι όλοι οι υποψήφιοι ανεξαιρέτως χρίουν κοινής μεταχείρισης από το σύστημα, τότε δημιουργείται ένα καλύτερο κλίμα στην όλη διαδικασία της ψηφοφορίας.

Μετά την ιδιότητα της ισότητας των υποψηφίων, έρχεται η ιδιότητα που θέλει το σύστημα ψηφοφορίας να είναι ελεύθερο από κάθε μορφής προπαγάνδας, άμεσης ή έμμεσης. Οι διαδικασίες που το διασφαλίζουν αυτό βασίζονται εν μέρει και στην προηγούμενη ιδιότητα, αλλά στην στερητική της απόδοση. Με αυτό εννοούμε ότι κανένας υποψήφιος δεν θα πρέπει να έχει το δικαίωμα να εισάγει κάποιο μήνυμα που θα μπορούν να διαβάσουν, δουν ή ακούσουν οι ψηφοφόροι. Η μη ύπαρξη μηνυμάτων βασίζεται στο γεγονός ότι κάποιος υποψήφιος μπορεί να εισάγει κατευθυνόμενα μηνύματα που να έχουν σκοπό και ίσως και αποτέλεσμα το να προσελκύσουν και άλλους ψηφοφόρους την τελευταία στιγμή. Καταλαβαίνουμε λοιπόν ότι κατά την διαδικασία της επιλογής του υποψηφίου ο ψηφοφόρος θα πρέπει να βλέπει κάτι πολύ απλό και κατανοητό και να μην τοποθετείται σε σκέψη τη στιγμή εκείνη. Θα πρέπει η ψήφος να γίνεται με «καθαρό μυαλό».

Τελευταία από αυτές τις ιδιότητες έρχεται η ιδιότητα που θέλει τα συστήματα ηλεκτρονικής ψηφοφορίας να μην καταγράφουν τα αποτελέσματα κατά τη διάρκεια της ψηφοφορίας, αλλά στο τέλος της ψηφοφορίας. Θέλουν δηλαδή τα συστήματα να παρουσιάζουν μια ξεκάθαρη διαφοροποίηση των φάσεων της ψηφοφορίας. Μπορούμε να κατανοήσουμε την ανάγκη για αυτόν τον ξεκάθαρο διαχωρισμό. Εάν η καταμέτρηση γίνεται κατά τη διάρκεια της ψηφοφορίας, τότε το αποτέλεσμα θα σχηματίζεται σταδιακά. Το αποτέλεσμα αυτής της σταδιακής δημιουργίας του αποτελέσματος μπορεί να έχει αρνητικό ή θετικό αντίκτυπο σε κάποιον από τους υποψηφίους ή τους ψηφοφόρους, οδηγώντας στην δημιουργία ψευδών εντυπώσεων. Το αποτέλεσμα θα πρέπει να δημιουργείται και να φαίνεται μόνο όταν η διαδικασία της ψηφοφορίας έχει τελειώσει.

6. Φάσεις ηλεκτρονικής Ψηφοφορίας

Κάθε εκλογική διαδικασία, στην κλασική και μη της προσέγγιση μπορεί να χωριστεί σε φάσεις. Οι φάσεις αυτές οριοθετούν τα διακριτά μέρη στην εκλογική διαδικασία που πρέπει να εκτελεστούν με κάποια ιεραρχική σειρά. Η σημαντικότητα του διαχωρισμού αυτού έγκειται στην ανάγκη για διαύγεια εντός των διαδικασιών. Ο τυπικός διαχωρισμός των φάσεων, οδηγεί σε τέσσερις φάσεις.

1. Εγγραφή στο σύστημα
2. Ψηφοφορία
3. Καταμέτρηση Ψήφων
4. Δημοσίευση αποτελεσμάτων

Εγγραφή στο σύστημα: Σε αυτήν την φάση η εκλογική αρχή δημιουργεί την λίστα με τους πολίτες που έχουν το δικαίωμα να ψηφίσουν. Παραδειγματικά, σε μια ψηφοφορία για την ανάδειξη του Πρωθυπουργού, η λίστα αυτή περιέχει όλους τους πολίτες του κράτους που έχουν συμπληρώσει το δέκατο όγδοο έτος της ηλικίας του.

Ψηφοφορία: Σε αυτή τη φάση, οι νόμιμοι ψηφοφόροι ασκούν το εκλογικό τους δικαίωμα. Ανάλογα πάντα με τους εκλογικούς νόμους που έχουν οριστεί από την εκλογική αρχή, οι ψηφοφόροι καλούνται να εκδηλώσουν την προτίμησή ή προτιμήσεις τους

Καταμέτρηση των ψήφων: Μετά την ολοκλήρωση της φάσης της ψηφοφορίας, ακολουθεί η καταμέτρηση των ψήφων. Είναι σημαντικό, η καταμέτρηση να είναι διαφορετική φάση και να μην γίνεται ταυτόχρονα με την ψηφοφορία, για να μην είναι δυνατός ο έμμεσος επηρεασμός των ψηφοφόρων από την προσωρινή έκβαση των αποτελεσμάτων. Στο τέλος της καταμέτρησης, τα αποτελέσματα αναδεικνύουν τον νικητή της εκλογικής διαδικασίας.

Δημοσίευση των αποτελεσμάτων: Η τελευταία αυτή φάση έρχεται μετά την καταμέτρηση των ψήφων. Πλέον η εκλογική διαδικασία έχει τελειώσει και ο νικητής της αναμέτρησης έχει αποφασιστεί από τους ψηφοφόρους. Σε αυτήν την φάση υπάρχουν οι διαδικασίες που ακολουθούνται για την δημοσίευση των αποτελεσμάτων της ψηφοφορίας.

Η κλασσική μέθοδος ψηφοφορίας έχει καθιερωθεί και εφαρμόζεται ένα αρκετά μεγάλο διάστημα. Είναι κατανοητό ότι μια προσπάθεια μετάβασης της κλασσικής ψηφοφορίας σε ηλεκτρονική θα πρέπει να διατηρεί ένα μεγάλο μέρος της μορφής της κλασσικής ψηφοφορίας, γιατί η λειτουργικότητά της είναι δεδομένη.

Συστήματα ηλεκτρονικής ψηφοφορίας

Τα συστήματα ηλεκτρονικής ψηφοφορίας μπορούν να καταταχθούν από λειτουργική άποψη σε γενικές κατηγορίες, ανάλογα με την μέθοδο που ακολουθείται για την ψηφοφορία.

A. Direct Recording Electronic (DRE)

Τα συστήματα ηλεκτρονικής ψηφοφορίας Direct Recording Electronic (DRE) είναι συστήματα τα οποία καταμετρούν τα αποτελέσματα της ψηφοφορίας απ' ευθείας στην μνήμη του υπολογιστή. Τα μηχανήματα έχουν τη μορφή αυτόνομης συσκευής ψηφοφορίας ή φορητού υπολογιστή. Τέτοια συστήματα έχουν χρησιμοποιηθεί σε εκλογικές διαδικασίες στην Ινδία. Τα συστήματα που χρησιμοποιήθηκαν εκεί αποτελούνται από δύο τμήματα, τον υπολογιστή και την οθόνη με τους διακόπτες επιλογής. Από εκεί ο ψηφοφόρος μπορεί να κάνει τις επιλογές του. Κάθε διακόπτης επιλογής συνδέεται με την ηλεκτρονική κάλπη με ξεχωριστή καλωδίωση και η ψήφος καταχωρείται ανάλογα.

Τα συστήματα DRE είναι διαδεδομένα και επιλέγονται συχνά σε εκλογικές διαδικασίες επειδή ενσωματώνουν συστήματα που προορίζονται για χρήση από άτομα με ειδικές ανάγκες, επιτρέποντας τους να ψηφίσουν χωρίς την βοήθεια ενός τρίτου προσώπου. Επιπλέον μπορούν να παρέχουν άμεσα αποτελέσματα εγκυρότητας κάθε ψήφου έτσι ώστε να παρέχεται η δυνατότητα στον ψηφοφόρο να διορθώσει την ψήφο του σε περίπτωση λάθους.

B. Mark-sense voting

Τα συστήματα mark – sense παρέχουν στον ψηφοφόρο μια συγκεκριμένη φόρμα ψηφοφορίας, την οποία συμπληρώνει και τοποθετεί στο μηχάνημα. Οι ψήφοι καταμετρούνται με ειδικούς ανιχνευτές στην ίδια τοποθεσία ή στα κεντρικά γραφεία της εκλογικής διαδικασίας, από τα αντίστοιχα μηχανήματα. Το πρόβλημα που μπορεί να προκύψει σε αυτήν τη μορφή ψηφοφορίας είναι ότι μπορεί η ψήφος να μην έχει τοποθετηθεί σωστά ανάλογα με τις δυνατότητες ανάγνωσης της μηχανής καταμέτρησης, ή να μην έχει συμπληρωθεί καθόλου.

C. Punch card voting

Με τις ψήφους διάτρητων καρτών οι ψηφοφόροι κάνουν τρύπες σε κάποιες κάρτες ψηφοφορίας και μετά τις εισάγουν στο μηχάνημα καταμέτρησης. Οι τρύπες γίνονται σε συγκεκριμένες τοποθεσίες στις κάρτες ανάλογα με την επιλογή τους. Υπάρχουν δύο κύριοι αντιπρόσωποι τέτοιων μηχανημάτων, η Datavote και η Votomatic. Τα συστήματα και των δύο χρησιμοποιούν ένα συγκεκριμένο εργαλείο που δημιουργεί τις τρύπες στις κάρτες, ανάλογα με τις επιλογές τους και ένα σύστημα για να μαζεύει τα υπολείμματα του χαρτιού. Η διαφορά των δύο συστημάτων έγκειται κυρίως σε αλλαγές εμφάνισης που θέλουν το Datavote σύστημα να αναγνωρίζει κάρτες με κόψιμο μορφής κύκλου εκεί που το Votomatic σύστημα αναγνωρίζει τρίγωνα. Αυτά τα συστήματα χρησιμοποιήθηκαν στις Αμερικάνικες προεδρικές

εκλογές το 2000 παρέχοντας την δυνατότητα ψηφοφορίας σε άτομα που ήταν εκτός της Αμερικής.

D. Telephone Voting

Η ηλεκτρονική ψηφοφορία μέσω τηλεφώνου επιτρέπει στους ψηφοφόρους να ασκήσουν τα εκλογικά τους δικαιώματα, τηλεφωνώντας σε κάποιους προκαθορισμένους τηλεφωνικούς αριθμούς. Αυτό μπορεί να γίνει με δύο τρόπους. Ο πρώτος είναι η χρήση συστημάτων που θα επιτρέπουν την επιλογή της ψήφου με αναγνώριση της φωνής και ο δεύτερος με την χρήση των πλήκτρων του τηλεφώνου. Το κύριο μειονέκτημα που παρουσιάζεται σε αυτή τη μορφή ψηφοφορίας είναι η δυσκολία που υπάρχει στην πιστοποίηση των ψηφοφόρων και στην διασφάλιση της αντιστοιχίας μιας ψήφου ανά ψηφοφόρο.

E. Internet Voting

Με την διαδικτυακή ψηφοφορία, οι ψηφοφόροι καταθέτουν την ψήφο τους ηλεκτρονικά με την χρήση του διαδικτύου. Τα συστήματα που υποστηρίζουν την διαδικτυακή ψηφοφορία, μπορούν να χωριστούν σε τρία μοντέλα ανάλογα κυρίως με την μορφή που παίρνει η ηλεκτρονική ψήφος και την τοποθεσία ψηφοφορίας.

- Ψηφοφορία από το σπίτι μέσω ηλεκτρονικού ταχυδρομείου
- Ψηφοφορία από το σπίτι online σε συγκεκριμένους δικτυακούς τόπους
- Ψηφοφορία από ένα περιφερειακό κέντρο που έχει τον κατάλληλο εξοπλισμό και μία σύνδεση στο Διαδίκτυο.

Το πρώτο μοντέλο απαιτεί να λειτουργεί η ηλεκτρονική ψηφοφορία ως εικονική μίμηση της λεγόμενης διαδικασίας ψηφοφορίας. Η μόνη διαφορά είναι ότι οι ψήφοι ζητούνται και αποστέλλονται μέσω ασφαλούς ηλεκτρονικού ταχυδρομείου. Οι ψήφος θα μπορεί έπειτα να συμπληρωθεί από τον ψηφοφόρο και, εκτυπωμένη και υπογεγραμμένη να επιτρέψει με τη χρήση της ταχυδρομικής υπηρεσίας ή ηλεκτρονικά στο γραφείο εκλογής μέσω ηλεκτρονικής υπογραφής. Τα κύρια πλεονεκτήματα αυτής της μεθόδου, σε σχέση με μία που χρησιμοποιεί δικτυακό τόπο είναι η μεγαλύτερη ευκολία για τους ψηφοφόρους που δεν έχουν ιδιαίτερη εμπειρία στο Διαδίκτυο ότι είναι λιγότερο ευαίσθητη στις δυσχέρειες μετάδοσης κατά την ώρα αιχμής.

Το δεύτερο μοντέλο πρεσβεύει την χρησιμοποίηση δικτυακών τόπων, στους οποίους οι ψηφοφόροι εισέρχονται μέσω ασφαλούς συνδέσεων, επιβεβαιώνουν της ταυτότητά τους και συμμετέχουν στην ηλεκτρονική ψηφοφορία. Αυτού του είδους η ψηφοφορία θα μπορούσε να ολοκληρωθεί από το σπίτι, το γραφείο την βιβλιοθήκη, το σχολείο ή οποιοδήποτε άλλο σημείο με πρόσβαση στο Διαδίκτυο και να πραγματοποιηθεί στο διάστημα αρκετών ημερών.

Ο τελευταίος μπορεί να χρησιμοποιήσει οποιοδήποτε κέντρο ψηφοφορίας μέσα στην γεωγραφική περιοχή, κάτι που επιτρέπεται εφόσον υπάρχει κεντρικό σύστημα βάσεων δεδομένων με τα στοιχεία των ψηφοφόρων και τις ψήφους τους. Η ύπαρξη της τεχνολογίας αυτό το πρότυπο πιθανό αν λάβουμε υπόψη τη βελτίωση της ταχύτητας που παρέχει σε σχέση με την παραδοσιακή λύση αλλά και την σχετικά

απλή υλοποίηση του λόγου των περιορισμένων εκλογικών κέντρων. Για τους πιο σημαντικούς, τα παραδοσιακά εκλογικά κέντρα μπορούν να χρησιμοποιήσουν την τεχνολογία του Διαδικτύου. Έτσι, η ψηφοφορία θα διεξάγεται σε εξοπλισμένα με υπολογιστές εκλογικά κέντρα., όπου - αφού επιβεβαιωθεί η ταυτότητα του ψηφοφόρου και συμπληρωθεί το ψηφοδέλτιο – η ψήφος θα αποστέλλεται μέσω Διαδικτύου στο σταθμό καταμέτρησης. Αυτή η μέθοδος επιταχύνει το υπάρχον σύστημα, ενώ παρέχει ασφάλεια και ευκολία στον ψηφοφόρο.

Βέβαια στην περίπτωση των παραδοσιακών εκλογικών κέντρων, τόσο η τεχνολογική πλατφόρμα όσο και η φυσική εγκατάσταση πρέπει να βρίσκονται κάτω από τον έλεγχο των αρμόδιων για τη διεξαγωγή της διαδικασίας της εκλογής και πρέπει επίσης να παρακολουθούνται κατάλληλα, ώστε να πληρούν τις απαιτήσεις ασφάλειας και ιδιωτικότητας και να εμποδίζουν τις μη εξουσιοδοτημένες παρεμβάσεις.

7. Υπάρχοντα συστήματα ηλεκτρονικής ψηφοφορίας

Η ηλεκτρονική δημοκρατία βρίσκει την ολοκλήρωσή της μέσα από τα συστήματα ηλεκτρονικής ψηφοφορίας. Ο προσπάθειες που έχουν γίνει μέχρι σήμερα για την κατασκευή ενός συστήματος που θα υποστηρίζει την διαδικασία της ηλεκτρονικής ψηφοφορίας είναι αξιοθαύμαστες. Οι προσπάθειες αυτές ποικίλουν, από θεωρητικά πρότυπα και μοντέλα, μέχρι ολοκληρωμένα συστήματα, τα οποία είτε τελούν υπό δοκιμή είτε έχουν χρησιμοποιηθεί σε διάφορα σημεία στον κόσμο, για διάφορες εκλογικές διαδικασίες.

Τα συστήματα αυτά, έχουν αναπτυχθεί, όχι για να καλύπτουν κάθε πιθανή μορφή εκλογικής διαδικασίας, αλλά για να παρέχουν υπηρεσίες σε κάποιο τμήμα των δυνατών μορφών ηλεκτρονικής ψηφοφορίας που μπορεί να υπάρχουν και να χρειάζονται. Αυτά τα συστήματα, μοιράζονται πολλές φορές ίδιες αρχιτεκτονικές και προσεγγίσεις υλοποίησης.

Υπάρχουν συστήματα για ηλεκτρονική ψηφοφορία μέσω εικονικής κάλπης, ηλεκτρονικού ταχυδρομείου, διαδικτύου ακόμα και αυτοματοποιημένα συστήματα, που επιτρέπουν την ψηφοφορία μέσω του τηλεφώνου. Η ανάλυση, ο σχεδιασμός και η υλοποίησης τέτοιων συστημάτων είναι χρονοβόρος και απαιτητικός στο θέμα των ατόμων που τα υποστηρίζουν. Κάθε ένα από αυτά τα συστήματα, χρειάστηκε χρόνια μελετών και σχεδιασμού για να μπορέσει πλέον να φτάσει στο σημείο της υλοποίησης.

Και αφού πλέον υλοποιήθηκαν και παραδόθηκαν για κάποια ψηφοφορία, παρουσιάστηκαν σε κάθε περίπτωση κάποια προβλήματα και ελλείψεις οι οποίες και έπρεπε να διορθωθούν. Έτσι οδηγήθηκαν πάλι στο στάδιο του σχεδιασμού και υλοποίησης αρκετές φορές μέχρι να φτάσουν στη σημερινή τους μορφή.

Τα συστήματα ηλεκτρονικής ψηφοφορίας που υπάρχουν αυτή τη στιγμή είναι:

- *VoteHere*
- *HART intercivic – eSlate3000*
- *Votations.com – Polls , surveys and direct marketing tools*
- *SAFEVOTE - The leader in Voting Technology*
- *SENSUS*
- *TrueBallot – WebVote, TouchVote, ScanVote , TeleVote*
- *Votia Empowerment*
- *EVOX*
- *EVS – The Association of Electronic Voting Systems*
- *Infopoll*
- *VoteNET*

Η πλήρης κατανόηση του συστήματος ηλεκτρονικής ψηφοφορίας, ως επίσης και η κατανόηση των δυνατοτήτων του, μπορεί να επιτευχθεί μέσω της γνώσης των δυνατοτήτων των αντίστοιχων συστημάτων στην κατηγορία του. Θα παρουσιάσουμε λοιπόν τα προαναφερθέντα συστήματα ηλεκτρονικής ψηφοφορίας και θα μπορέσουμε να παρουσιάσουμε τις παρούσες ανάγκες και δυνατότητες των μέχρι στιγμής υπαρκτών συστημάτων.

A. Vote Here, Inc.

Η επιχείρηση VoteHere, ιδρύθηκε το 1998 και έχει τα γραφεία της στην Washington. Αποτελεί μια από τις πρωτοπόρες επιχειρήσεις στην κατασκευή συστημάτων ηλεκτρονικής ψηφοφορίας. Η εταιρία, στα συστήματα που παρέχει δίνει ιδιαίτερη βαρύτητα στο θέμα της ασφάλειας. Η τεχνολογία VoteHere έχει μέχρι στιγμής χρησιμοποιηθεί σε περισσότερες από 90 εκλογικές διαδικασίες στις Ηνωμένες Πολιτείες Αμερικής και στην Ευρώπη, αγγίζοντας το αστρονομικό ποσό των 13 εκατομμυρίων ψηφοφόρων. Από το 1999 μέχρι το 2002, η VoteHere είχε κατοχυρώσει τις πατέντες που χρησιμοποιούνται στο σύστημά της και τον Σεπτέμβριο του 2003 δημοσίευσε μια καινοτομία που ενσωμάτωσε στα συστήματά της με την ονομασία VHTi.

Το VHTi είναι μια τεχνολογία διαπίστευσης ψήφων. Το κοινό – στόχος της είναι οι ψηφοφόροι και το σημαντικό της πλεονέκτημα είναι ότι μπορεί να ολοκληρωθεί σε διάφορα συστήματα ηλεκτρονικής ψηφοφορίας. Αυτή η τεχνολογία έχει δύο παραδοτέα κατά τη διαδικασία της ψηφοφορίας. Πρώτον, δίνει τη δυνατότητα στους ψηφοφόρους να πάρουν μια «Ιδιωτική Απόδειξη» της ψήφου τους και δεύτερον, δημιουργεί ένα Election Transcript το οποίο παρέχει μια διαφάνεια στην διαδικασία καταμέτρησης των αποτελεσμάτων. Με αυτό το αρχείο, μπορεί ο κάθε ψηφοφόρος ξεχωριστά να επαληθεύσει την ψήφο του μετά τη λήξη της καταμέτρησης.

Η υλοποίηση αυτής της τεχνολογίας είναι σχετικά απλή. Κατά τη διάρκεια της ψηφοφορίας, ο ψηφοφόρους επιλέγει και παίρνει την απόδειξη του σχήματος 1 και στην λήξη της ψηφοφορίας μπορεί να επαληθεύσει την ψήφο του.

VOTE VERIFICATION RECEIPT

BALLOT #48763

---Begin Ballot Record---

[illegible]

---End Ballot Record---

James	HLM 22
Smith	RSO 22
Martinez	ZVJ 22
Yes	PKT 22
No	OPU 22

---Begin Signature---

[illegible]

---End Signature---

PLEASE KEEP THIS RECEIPT TO VERIFY YOUR BALLOT ONLINE.

Σχήμα 8.A.1: Απόδειξη Ψηφοφορίας VHTi

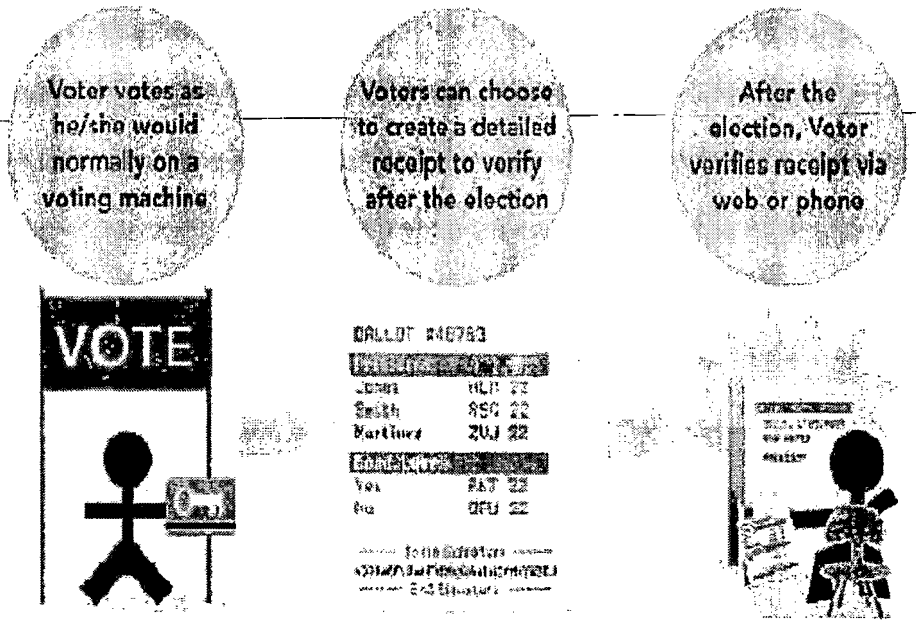
Η απόδειξη ψήφου που λαμβάνει ο πολίτης μετά την ψήφο του έχει ένα σύνολο από κωδικούς που χρησιμοποιούνται κατά την επαλήθευση της ψήφου. Αρχικά έχει ένα κωδικό που δείχνει την ψήφο του κάθε ψηφοφόρου. Αυτός ο αριθμός είναι τυχαίος και δεν καταγράφεται το άτομο που ψήφισε στο σύστημα.

Ανάμεσα στα πεδία Begin/End Ballot Record υπάρχει η ψήφος του πολίτη σε κρυπτογραφημένη μορφή. Αυτό το πεδίο θα χρησιμοποιηθεί κατά την επαλήθευση της ψήφου του πολίτη.

Ακολούθως υπάρχουν τα πεδία που δείχνουν τους υποψηφίους μεταξύ των οποίον ψήφισε ο ψηφοφόρος καθώς και ένα τριψήφιο κωδικό που δείχνει το μηχάνημα στο οποίο ψήφισε ο ψηφοφόρος.

Τέλος ανάμεσα στα πεδία Begin / End Signature υπάρχει ένας κωδικός που δείχνει ότι η συγκεκριμένη απόδειξη είναι γνήσια και δεν έχει πλαστογραφηθεί.

Μετά την λήξη της ψηφοφορίας, ο πολίτης- ψηφοφόρος μπορεί να ελέγξει την ψήφο του μέσω του διαδικτύου μιας αντίστοιχης τηλεφωνικής ψηφοφορίας. Ο πολίτης με την απόδειξη αυτή, συνδέεται στο σύστημα ελέγχου και εισάγει τον κωδικό που βρίσκεται στην ψήφο του και μετά το κρυπτογραφημένο μέρος της ψήφος του. Το σύστημα ψάχνει στην βάση του και βρίσκει την εγγραφή που ταιριάζει με την είσοδο του ψηφοφόρου. Αφού βρει την ψήφο του πολίτη, τον ενημερώνει, ότι υπάρχει στα αρχεία του και έχει καταμετρηθεί στην φάση της καταμέτρησης των ψήφων. Η διαδικασία που ακολουθείται μπορεί να αποδοθεί σχηματικά με το σχήμα 2.



Σχήμα 8.Α.2: Διαδικασία συστήματος VHTi

Το σύστημα αυτό όπως προαναφέρθηκε, εισάγει μια τεχνολογία στην κοινωνία της ηλεκτρονικής ψηφοφορίας, που διασφαλίζει ότι η διαδικασία της ψηφοφορίας, είναι απρόσβλητη, στο θέμα της αλλαγής ψήφων. Δημιουργεί ένα κλίμα εμπιστοσύνης ανάμεσα στους πολίτες και στην εκλογική διαδικασία και μπορεί να χρησιμοποιηθεί για να στηρίξει μικρής, μεσαίας και μεγάλης έκτασης εκλογικές διαδικασίες. Επιπλέον η δυνατότητά της να ολοκληρώνεται με άλλα συστήματα ηλεκτρονικής ψηφοφορίας, διευρύνει τις δυνατότητες χρήσης της.

B. HART intercivic – eSlate3000

Η Hart InterCivic γνωστή παλιότερα ως Hart Information Services, είναι μια ακόμα από τις επικρατούσες επιχειρήσεις στον τομέα της ηλεκτρονικής διακυβέρνησης και ηλεκτρονικής ψηφοφορίας. Έχει προϊόντα και υπηρεσίες που υποστηρίζουν περισσότερους από 5000 πελάτες σε 13 πολιτείες. Η επιχείρηση έχει στο δυναμικό της περισσότερους από 250 υπαλλήλους, με γραφεία στο Texas, το Colorado και την Alabama. Τα προϊόντα τους έχουν ολοκληρωθεί με συστήματα που υποστηρίζουν ψηφιακές κάρτες και βιομετρικές μεθόδους πιστοποίησης των χρηστών. Το τελευταίο προϊόν της επιχείρησης στον τομέα της ηλεκτρονικής ψηφοφορίας είναι το eSlate 3000 σύστημα ηλεκτρονικής ψηφοφορίας.

Σχήμα 8.B.1: eSlate3000

Το eSlate3000 είναι το πιο σύγχρονο και προσιτό σύστημα Άμεσης Εγγραφής Ψήφου που είναι διαθέσιμο την παρούσα χρονική στιγμή. Οι σχεδιαστές του eSlate3000 έδωσαν έμφαση στην ευχρηστία του από μεγάλο αριθμό ανθρώπων. Η ψήφος αναπαριστάται με τέτοιο τρόπο που να είναι ευδιάκριτη και να μην αφήνει περιθώρια λάθους επιλογής. Επίσης έχει προβλεφθεί στο σχεδιασμό η κατασκευή του προϊόντος και για χρήση από Άτομα με Ειδικές Ανάγκες.

Με κάποια μικρή τροποποίηση, λοιπόν το eSlate3000 γίνεται DAU 5000, έτσι ώστε να μπορεί να υποστηρίξει τις συσκευές που μπορούν να υποστηρίξουν ψηφοφορία για τους ανάπηρους. Οι αλλαγές που γίνονται είναι η υποστήριξη ειδικών διεπαφών επικοινωνίας. Αυτές οι διεπαφές χρησιμοποιούν όργανα που κατανοούν την κίνηση του κεφαλιού και τις εισπνοές και εκπνοές του πολίτη. Επιπλέον υποστηρίζεται η δυνατότητα ηχητικής παρουσίασης των ψήφων, για τους ανθρώπους με πρόβλημα στην όραση και η ηχητική επιλογή ψήφου. Τέλος, το σύνολο της πλοήγησης στο σύστημα DAU 5000 γίνεται με τον συνήθη σχεδιασμό σε προγράμματα για τα AMEA.

Το eSlate 3000 χρησιμοποιεί μια PCMCIA κάρτα μνήμης, για την αποθήκευση των ψήφων. Αυτές οι Mobile Ballot Boxes μεταφέρονται μετά στα κεντρικά γραφεία που χρησιμοποιούνται για την ψηφοφορία και γίνεται η καταμέτρησή τους και η εξαγωγή των αποτελεσμάτων.

Αυτό το σύστημα ηλεκτρονικής ψηφοφορίας, είχε ως σκοπό του την παροχή μιας ασφαλούς και διαυγής ψηφοφορίας, δίνοντας την ευκαιρία ψήφου και στα άτομα που, λόγο κάποιου προβλήματος στην υγεία τους, δεν θα μπορούσαν να χρησιμοποιήσουν ένα τέτοιο σύστημα.

C. Votations.com

Η επιχείρηση votations.com εξειδικεύεται στην παροχή υπηρεσιών δημοψηφισμάτων και δημοσκοπήσεων, που αποτελούν μια ελαφριά μορφή διαδικτυακής ψηφοφορίας. Μπορεί ο σχεδιασμός του συστήματος ηλεκτρονικής ψηφοφορίας να κατευθύνεται σε μια άλλη φιλοσοφία, αλλά στα συστήματα δημοσκόπησης παρουσιάζονται παρεμφερή προβλήματα και λύσεις. Ο σχεδιασμός των υπηρεσιών έχει γίνει με πολύ προσοχή και στο θέμα της ασφαλείας, η votations.com αποτελεί μια από τις καλύτερες επιχείρησης στον τομέα της. Μέσα από τις υπηρεσίες δίνεται η δυνατότητα στους πελάτες να κατασκευάσουν μια δημοσκόπηση μέσα σε λίγα λεπτά και να τη διαχειριστούν απομακρυσμένα χωρίς ιδιαίτερη απαίτηση γνώσεων. Οι επιλογές που μπορεί να υποστηρίξει η υπηρεσία διαδικτυακής δημοσκόπησης είναι:

- Φιλική διεπαφή χρήστη και γρήγορους βοηθούς
- Μέχρι 50 πεδία απαντήσεων στην δωρεάν έκδοση των δημοσκοπήσεων
- Περισσότερες από 11 επιλογές σχεδίασης
- Η ψηφοφορία μπορεί αυτόματα να αντικατασταθεί από τα αποτελέσματα μετά από μια ψήφο
- Δυνατότητα cut & past σε HTML κώδικα
- Επιλογή ημερομηνίας και ώρας της λήξης της δημοσκόπησης
- Έτοιμα πεδία για χώρες, πόλεις κ.λ.π
- Αλφαβητική παρουσίαση των αποτελεσμάτων της ψηφοφορίας
- Cookie, IP και email μεθόδους ασφαλείας για την αποφυγή διπλο-εγγραφών
- IP banning (απαγόρευση εισόδου στο σύστημα) των ανεπιθύμητων IP
- Προστασία και απόκρυψη των αποτελεσμάτων με κωδικό
- Απαγόρευση προβολής των αποτελεσμάτων εως ότου συμπληρωθεί προκαθορισμένος αριθμός ψήφων
- Μετάφραση των μηνυμάτων σε πολλές γλώσσες
- Παραμετροποιήσιμη παρουσίαση αποτελεσμάτων
- Java πίνακες μορφής ράβδου και πίτας
- Απαντήσεις μορφής κειμένου ή γραφικών
- Υποστήριξη Firewall και proxy προώθησης
- Email διαμοιρασμός των αποτελεσμάτων
- Email ειδοποίηση για κάθε νέα ψήφο
- Επιλογές ερωτήσεων τύπου quiz

D. SENSUS

Το SENSUS αποτελεί ένα σύστημα ηλεκτρονικής ψηφοφορίας, προσανατολισμένο στην ασφάλεια. Όπως και τα υπόλοιπα συστήματα ηλεκτρονικής ψηφοφορίας στόχο έχει την ασφαλή και έμπιστη διεξαγωγή των συνεδριών ψηφοφορίας. Ο κύριος στόχος του είναι να καλύψει τα προβλήματα που παρουσιάζονται στην ολοκλήρωση της ασφάλειας σε ένα τέτοιο σύστημα. Εξαιτίας της πολυπλοκότητας των διαδικασιών και της αντίφασης που δημιουργείται στην προσπάθεια κάλυψης των αναγκαίων ιδιοτήτων, είναι κατανοητό ότι δεν μπορεί να δημιουργηθεί ένα σύστημα που να καλύπτει πλήρως όλες τις ιδιότητες στον μέγιστο επιθυμητό βαθμό. Το πρωτόκολλο SENSUS έχει παραλείψει δύο μέρη από τις ιδιότητες αυτές. Στην περίπτωση που οι ψηφοφόροι δεν καλούνται να ψηφίσουν σε μέρος που να είναι απροσπέλαστο από τους λοιπούς ψηφοφόρους τη δεδομένη στιγμή της ψηφοφορίας, δεν είναι εφικτό να διασφαλιστεί ότι δεν θα καταθέτουν την ψήφο τους, έχοντας κάποιον να τους επιβλέπει για τον τρόπο που ψήφισαν. Επίσης όπως τα περισσότερα κατανεμημένα συστήματα ηλεκτρονικής ψηφοφορίας, έτσι και το SENSUS δεν ασχολείται με το θέμα της υποκλοπής ή καθυστέρησης της μετάδοσης της ψήφου μεταξύ του ψηφοφόρου και του συστήματος.

Εξαιρουμένων αυτών των δύο θεμάτων, το SENSUS αποτελεί μια πλήρη λύση στο θέμα της ηλεκτρονικής ψηφοφορίας. Η υλοποίηση του έχει τέσσερα μέρη

Το ***Registrar***. Εδώ γίνεται η εγγραφή των χρηστών προ της συνεδρίας ηλεκτρονικής ψηφοφορίας

Τον ***Pollster***. Ο pollster είναι υπεύθυνος για την συνεδρία της ψηφοφορίας στο σύνολό της. Ο ρόλος του είναι να είναι ο μεσολαβητής των ψηφοφόρων, η παρουσίαση των ψήφων σε μορφή κατανοητή από τον άνθρωπο, η συλλογή των ψήφων από τους ψηφοφόρους, η εκτέλεση των απαραίτητων κρυπτογραφιών στις ψήφους και τέλος η αποστολή των ψήφων στις κάλπες. Το Pollster είναι το μοναδικό στοιχείο που πρέπει οι ψηφοφόροι να εμπιστεύονται ολοκληρωτικά.

Ο ***Validator***. Ο ρόλος του Validator είναι να διασφαλίζει ότι μόνο οι νόμιμοι ψηφοφόροι θα μπορούν να ψηφίσουν και ότι και αυτοί θα μπορούν να ψηφίσουν μόνο μια φορά.

Ο ***Tallier***. Το τελευταίο στοιχείο του SENSUS είναι ο tallier, ο οποίος είναι υπεύθυνος για την καταμέτρηση των αποτελεσμάτων της ηλεκτρονικής ψηφοφορίας ή δημοσκόπησης.

Αυτά τα τέσσερα συστήματα συνεργαζόμενα μεταξύ τους συνθέτουν το σύστημα ηλεκτρονικής ψηφοφορίας SENSUS και δίνουν στον κυβερνοκόσμο ένα πρωτόκολλο για ασφαλή και αξιόπιστη διεξαγωγής συνεδριών ψηφοφορίας. Το SENSUS για να λειτουργήσει απαιτεί μια σειρά από ψηφιακές υπογραφές και πιστοποιητικά. Μια περίληψη των διαδικασιών που εκτελούνται σε μια συνεδρία ψηφοφορίας μπορεί να μας βοηθήσει να κατανοήσουμε τις διαδικασίες που επιτελούνται. Αξίζει να σημειωθεί ότι όλες οι κρυπτογραφικές συναρτήσεις γίνονται από τον pollster, εκ μέρους του ψηφοφόρου.

- Ο pollster κρυπτογραφεί την σημειωμένη ψήφο με ένα ειδικό ιδιωτικό κλειδί που θα χρησιμοποιήσετε μόνο μια φορά.

- Ο pollster "τυφλώνει" την κρυπτογραφημένη ψήφο
- Ο pollster υπογράφει την "τυφλωμένη" ψήφο με το συνηθισμένο ιδιωτικό κλειδί του ψηφοφόρου.
- Ο pollster καταθέτει την τυφλωμένη ψήφο , την υπογραφή και τον ID αριθμό του ψηφοφόρου στον validator.
- Ο validator χρησιμοποιεί τον ID αριθμό του ψηφοφόρου για να βρει το δημόσιο κλειδί, να βεβαιώσει ότι ο ψηφοφόρος δεν έχει ξαναψηφίσει μέχρι τώρα και για να πιστοποιήσει την υπογραφή του. Αξίζει να σημειωθεί ότι η λίστα των εγγεγραμμένων αριθμών των ψηφοφόρων και το αντίστοιχο δημόσιο κλειδί παρέχεται από τον registrar.
- Εάν ο validator επιβεβαιώσει ότι ο ψηφοφόρος είναι νόμιμος, τότε υπογράφει την τυφλωμένη ψήφο και την επιστρέφει στον ψηφοφόρο.
- Ο pollster χρησιμοποιεί μια τεχνική ανάκτησης τυφλών υπογραφών, για να αφαιρεί το επίπεδο κρυπτογράφησης από την ψήφο, αποκαλύπτοντας την υπογραφή του ψηφοφόρου σε μια ψήφο που είναι κρυπτογραφημένη μόνο με το ιδιωτικό κλειδί του ψηφοφόρου.
- Ο pollster στέλνει την υπογεγραμμένη, κρυπτογραφημένη ψήφο στον tallier.
- Ο tallier πιστοποιεί ότι η υπογραφή του validator στην ψήφο είναι αυθεντική και τοποθετεί την κρυπτογραφημένη ψήφο σε μια λίστα μαζί με άλλες ψήφους για καταμέτρηση αργότερα. Ο tallier τότε υπογράφει την ψήφο και την επιστρέφει στον ψηφοφόρο υπό τη μορφή απόδειξης ψηφοφορίας. Εάν στην πορεία της διαδικασίας ψηφοφορίας ο ψηφοφόρος διαπιστώσει ότι η ψήφος του απουσιάζει από τις δημοσιευμένες ψήφους (και κατά συνέπεια δεν καταμετρήθηκε), υπάρχει η δυνατότητα της επανακατάθεσής της, μέσω της απόδειξης
- Ο pollster στέλνει το δημόσιο κλειδί που θα αποκρυπτογραφήσει την ψήφο του ψηφοφόρου στον tallier.
- Ο tallier χρησιμοποιεί αυτό το δημόσιο κλειδί για να αποκρυπτογραφήσει την ψήφο και να την προσθέσει στην καταμέτρηση.
- Μετά από την ψηφοφορία, όλες οι ψήφοι δημοσιεύονται μαζί με τα αντίστοια δημόσια κλειδιά τους.

E. True Ballot

Η επιχείρηση TrueBallot,Inc παρέχει υπηρεσίες ψηφοφορίας σε εργατικά σωματεία και μεμονωμένες επιχειρήσεις και οργανισμούς. Έχει αναπτύξει μια σειρά από συστήματα τα οποία μπορούν να καλύψουν όλων των ειδών τις ανάγκες ψηφοφορίας. Τα συστήματα της TrueBallot παρέχουν ασφάλεια, αξιοπιστία και ταχύτητα στις αντίστοιχες εκλογικές διαδικασίες. Είναι εύκολα στην χρήση και διαχείριση τους από τους ψηφοφόρους και την εκλογική αρχή και μπορούν να συνεργαστούν μεταξύ τους. Τα συστήματα ηλεκτρονικής ψηφοφορίας που παρέχει η TrueBallot είναι τα:

ScanVote, TeleVote, TouchVote και WebVote. Το κάθε ένα από αυτά ειδικεύεται σε διαφορετική μορφή ψηφοφορίας.

Το σύστημα ScanVote έχει αναπτυχθεί για να υποστηρίζει διαδικασίες ψηφοφορίας κλασσικής μορφής. Έχει τη δυνατότητα να παράγει αυτοματοποιημένα την εγγραφή και το σύστημα καταμέτρησης σε ψηφοφορίες που γίνονται μέσω αλληλογραφίας και σε On-site ψηφοφορίες. Με τη χρήση μιας βάσης νόμιμων ψηφοφόρων το ScanVote παράγει μια μεμονωμένη ψήφο για τον κάθε ψηφοφόρο, ανάλογα με την τοποθεσία από όπου ψηφίζει. Κάθε ψηφοφόρος καταληκτικά, μπορεί να δει στο ψηφοδέλτιό του μόνο τις ψηφοφορίες στις οποίες έχει δικαίωμα να συμμετέχει. Επιπλέον, ανάλογα με τις ανάγκες του οργανισμού στην εκάστοτε ψηφοφορία, μπορεί να διαμορφώνεται δυναμικά το κάθε ψηφοδέλτιο, στα θέματα ψηφοφορίας που υπάρχουν για κάθε ψηφοφόρο. Στο σύστημα ScanVote η καταμέτρηση των ψήφων γίνεται με την μέθοδο της οπτικής αναγνώρισης. Κάθε ψήφος αναλύεται από μια συσκευή και ανάλογα με τα σημεία που έχει σημειώσει ο ψηφοφόρος γίνεται η ψηφιοποίησή της και ταυτόχρονα η καταμέτρηση και ενημέρωση των αποτελεσμάτων.

Το TeleVote είναι ένα σύστημα ηλεκτρονικής ψηφοφορίας που χρησιμοποιεί την φωνή για την διεξαγωγή των συνεδριών ψηφοφορίας. Με τη χρήση των τηλεφωνικών συστημάτων μπορεί να εκτελέσει μια ευρεία γκάμα από ψηφοφορίες και δημοσκοπήσεις. Και αυτό χρησιμοποιεί μια βάση δεδομένων με τους νόμιμους ψηφοφόρους και μέσα από αυτή παράγει μεμονωμένα φωνητικά ψηφοδέλτια ανάλογα με τον ψηφοφόρο. Σε αυτό το σύστημα, ο κάθε ψηφοφόρος τηλεφωνεί σε έναν προκαθορισμένο αριθμό, και μετά από την απαραίτητη διαπίστευση των στοιχείων του, ακούει τις ψηφοφορίες στις οποίες έχει δικαίωμα να συμμετέχει. Με τη σειρά του ο ψηφοφόρος χρησιμοποιώντας τα πλήκτρα του τηλεφώνου του, επιλέγει την ψήφο του. Με τη λήξη της συνεδρίας της ψηφοφορίας, το σύστημα δίνει τη δυνατότητα στον ψηφοφόρο να επαληθεύσει την ψήφο του και εάν επιθυμεί και να την αλλάξει προτού κατατεθεί για καταμέτρηση. Από τη στιγμή που έχει κατατεθεί η ψήφος, το σύστημα δεν επιτρέπει στον δεδομένο ψηφοφόρο να ξανασυνδεθεί.

Ακολούθως οι ψήφοι χωρίζονται ανάλογα με τις ψηφοφορίες και καταγράφονται σε μέσα. Το προηγμένο σύστημα ηλεκτρονικής ψηφοφορίας μέσω τηλεφώνου TeleVote συνδυάζει καλύτερα αποτελέσματα με χαμηλότερα κόστη. Επιπλέον η υψηλή ακρίβεια, η ευελιξία και η ισχυρή ασφάλεια που έχει ενσωματωθεί, το καθιστούν μια από τις ισχυρότερες λύσεις στην ψηφοφορία για θέματα που αφορούν έναν οργανισμό.

Το TouchVote είναι ένα σύστημα που έχει σχεδιαστεί για χρήση σε εκλογικές διαδικασίες οργανισμών και υποστηρίζει ψηφοφορίες που γίνονται σε συγκεκριμένη τοποθεσία. Το σύστημα έχει το ίδιο υπόβαθρο με τα υπόλοιπα συστήματα της TrueBallot, αλλά διαφοροποιείται στον τρόπο με τον οποίο γίνεται η ψηφοφορία.

Το TouchVote χρησιμοποιεί ως μέσω επικοινωνίας με τον ψηφοφόρο οθόνες αφής. Με αυτόν τον τρόπο γίνεται η λήψη των στοιχείων του χρήση και η διαπίστευσή του. Αφού ο χρήστης έχει εισάγει τα στοιχεία του και έχει διαπιστευτεί ως νόμιμος ψηφοφόρος, σχηματίζεται από το σύστημα το μεμονωμένο ψηφοδέλτιό του, ανάλογα με τη δυνατότητά του να συμμετάσχει σε διαφορετικές ψηφοφορίες.

Το ψηφοδέλτιο με τους υποψηφίους παρουσιάζεται στην οθόνη του συστήματος και ο ψηφοφόρος με συνοπτικές διαδικασίες μπορεί να καταθέσει την επιλογή του. Τα δυνατά σημεία που μπορεί να επιλέξει ο ψηφοφόρος είναι προκαθορισμένα και η συμπλήρωση της ψήφου γίνεται σύμφωνα με τις υποδείξεις της εκλογικής αρχής. Μέσω της οθόνης αφής ο χρήστης μπορεί να ελέγξει την ψήφο του προτού να την καταθέσει και να την αλλάξει όπου αυτός επιθυμεί. Από τη στιγμή που θα καταθέσει την ψήφο του, το σύστημα τοποθετείται στην αρχική του κατάσταση περιμένοντας τον επόμενο ψηφοφόρο.

Το WebVote αποτελεί την πρόταση της TrueBallot για την ηλεκτρονική ψηφοφορία με χρήση του διαδικτύου. Το σύστημα WebVote επιτρέπει στους ψηφοφόρους να προσπελάσουν μια δεδομένη ηλεκτρονική ψηφοφορία από οποιοδήποτε μέρος στον κόσμο. Η εκλογική διαδικασία και σε αυτό το σύστημα παραμένει ίδια με τα προηγούμενα. Ο πολίτης μετά από την παρουσίαση των διαπιστευτηρίων του στο σύστημα προχωράει στην ψηφοφορία όπου και επιλέγει τους υποψηφίους της αρεσκείας του, ανάλογα με τις ψηφοφορίες στις οποίες μπορεί να συμμετέχει. Ακολούθως, ο πολίτης μπορεί να επανεξετάσει την ψήφο του προτού να την καταθέσει, αφού ψηφίσει δεν μπορεί να ξανασυνδεθεί με το σύστημα.

Το σημαντικό στην πρόταση της TrueBallot, είναι η δυνατότητα εξυπηρέτησης κάθε είδους ψηφοφορίες, από την κλασική μέχρι την ηλεκτρονική. Και στην μορφή της ηλεκτρονικής ψηφοφορίας, όλα τα συστήματα μπορούν να ολοκληρωθούν και να συνεργαστούν μεταξύ τους. Η κοινή αρχιτεκτονική και φιλοσοφία που βρίσκεται στον πυρήνα των συστημάτων της TrueBallot, τα καθιστά ικανά να παρέχουν υψηλού επιπέδου υπηρεσίες ψηφοφορίας, ανεξαρτήτως του κοινού που απευθύνονται την εκάστοτε στιγμή.

F. Votia Empowerment

Η Votia Empowerment είναι μια Σουηδική επιχείρηση που παρέχει συστήματα ηλεκτρονικής διακυβέρνησης κυρίως για τους οργανισμούς του κράτους της Σουηδίας. Ο σχεδιασμός των προϊόντων έχει γίνει κάτω από αυστηρά πρότυπα και πλαίσια, ώστε να μπορεί να διασφαλίσει την ακεραιότητα των αποτελεσμάτων, την ισοτιμία των συμμετεχόντων, την διαύγεια και την αποτελεσματικότητα. Η Votia Empowerment προσφέρει δύο προϊόντα. Το VotiaReferendum και το VotiaConsultation.

VotiaReferendum

Το σύστημα αυτό έχει σχεδιαστεί για να παρέχει υψηλή ασφάλεια και ταχύτητα στις διαδικασίες ψηφοφορίας. Η υλοποίησή του έγινε για την διαχείριση, των θεμάτων που θα αποφασιστούν μέσω ψηφοφορίας, σε δημόσιους οργανισμούς. Η ανάγκη που πρέπει να καλύψει το σύστημα πριν να μπορεί να χρησιμοποιηθεί είναι να διασφαλίζει στους συμμετέχοντες την ακεραιότητα των αποτελεσμάτων, και την διαυγή διεξαγωγή της ψηφοφορίας.

Στην περίπτωση που θα επιλεγεί η Votia Empowerment, ως ο ανεξάρτητος τρίτος ελεγκτής στην εκάστοτε ψηφοφορία, μπορεί να εγγυηθεί την ακεραιότητα των υποψηφίων στην διαδικασία, ως επίσης και την ορθότητα των αποτελεσμάτων. Η επιτυχία του προϊόντος αυτού βασίστηκε στις ακόλουθες επιλογές που προσφέρει.

- Διαχείριση Εφαρμογών
- Σχεδιασμό Θεμάτων ψηφοφορίας
- Πληροφορίες και συμβουλές επικοινωνίας
- Συνεχής υποστήριξη κατά τη διάρκεια της ψηφοφορίας

VotiaConsultation

Το VotiaConsultation είναι ένα σύστημα της Votia Empowerment που έχει σαν στόχο του τη συλλογή πληροφοριών σε θέματα αλλαγής διαδικασιών, για διάφορους οργανισμούς. Η χρήση αυτού του συστήματος οδηγεί στη λήψη καλό-σχηματισμένων αποφάσεων μετά από συνεργασία με τα άτομα που θα το χρησιμοποιήσουν. Δημιουργεί ένα είδος κοινότητας ανάμεσα στα άτομα που συμμετέχουν και καταθέτουν τις απόψεις τους. Οι οργανισμοί που επιλέγουν να το χρησιμοποιούν, παίρνουν σαν αποτέλεσμα μια δομημένη βάση για την λήψη των αποφάσεων.

Η Votia Empowerment με αυτό της το σύστημα παρέχει διασφάλιση της ποιότητας των αποφάσεων μέσω της διαχείρισης του έργου, του σχηματισμού των ερωτήσεων και των πληροφοριών και συμβουλών επικοινωνίας που παρέχει. Τα παγκόσμια και ιδιωτικά δικαιώματα που παρέχει, στο λογισμικό της υποστηρίζουν την ασφάλεια μέσω κωδικών και σε διαδικτυακές ψηφοφορίες και σε κλασσικές ψηφοφορίες.

G. EVOX

Το πρωτόκολλο EVOX αναπτύχθηκε αρχικά από τους Fujioka, Okamoto και Ohta. Αποτελεί ένα σύστημα το οποίο εγγυάται μυστική ηλεκτρονική ψηφοφορία και διασφάλιση της ακεραιότητας των ψήφων. Από την πρώτη του έκδοση μέχρι σήμερα, έχει αλλάξει μορφή και έχουν προστεθεί πλεονάζουσες λειτουργίες που του επιτρέπουν να πληρεί καλύτερα και τις υπόλοιπες ιδιότητες των συστημάτων ηλεκτρονικής ψηφοφορίας. Έτσι έχουν προστεθεί ένας ανώνυμος remailer και ένας εκλογικός αντιπρόσωπος. Ο ανώνυμος remailer είναι ένα πρόγραμμα που επιτρέπει την μεταφορά των ψήφων μέσω ηλεκτρονικού ταχυδρομείου, με την αφαίρεση των μερών στα οποία αναφέρεται η προέλευση του μηνύματος. Έτσι το Anon παρέχει ένα ασφαλές ανώνυμο κανάλι επικοινωνίας. Ο Εκλογικός Αντιπρόσωπος είναι ένα πρόγραμμα στο οποίο στέλνονται όλα τα παράπονα, σε μια εκλογική διαδικασία. Οι υπεύθυνοι αυτού του συστήματος είναι υπεύθυνοι για την εποπτεία των πιθανών σφαλμάτων σε πραγματικό χρόνο. Εάν για παράδειγμα τους αναφερθεί ότι ο ADMIN δεν υπογράφει σωστά τις ψήφους, τότε αυτοί ελέγχουν την ορθότητα του ADMIN. Μιας και αποτελεί την πρώτη προσπάθεια συστήματος ηλεκτρονικής ψηφοφορίας αξίζει να παρατηρήσουμε τις διαδικασίες που ακολουθούνται σε μια εκλογική διαδικασία από αυτό το σύστημα.

1. **Εγκατάσταση**: Εδώ δημιουργείται η Επιτροπή ψηφοφορίας (Election Commission)
2. **Εγγραφή**: Σε αυτή τη φάση έχουμε τρεις διαδικασίες. Αρχικά γίνεται η λίστα των νόμιμων ψηφοφόρων, μετά διαμοιράζονται οι κωδικοί πρόσβασης και ψηφοφορίας και τέλος αρχικοποιούνται τα συστήματα που θα υποστηρίξουν την εκλογική διαδικασία.
3. **Εκλογικό Βήμα 1**: Πιστοποίηση της ψήφου

Ο Ψηφοφόρος

a. Καλεί την ιστοσελίδα (χρήση SSL), η οποία περιλαμβάνει το δημόσιο κλειδί του Admin (APK)
b. Ο ψηφοφόρος εισάγει το όνομα και τον κωδικό του
c. Ο ψηφοφόρος κάνει τις εκλογικές του επιλογές, σχηματίζοντας μια ψήφο από b επιλογές
d. Η ψήφος συνδιάζεται (b)=HMAC-SHAk1,k2(b)
e. Η commited ψήφος γίνεται blinded Bl = re□(b) mod n, όπου n είναι τα στοιχεία του APK και r είναι ένας τυχαίος αριθμός που παράγεται από τον χρήστη
f. Ένα κλειδί συνεδρίας παράγεται από τον ψηφοφόρο και στέλνεται στον Admin, κρυπτογραφημένο με το APK: EA-PK(AS-Key)
g. Ο ψηφοφόρος στέλνει στον Admin το : EAS-Key(Bl, name, password), EAS Key(MAC(EAS-Key(Bl, name, password)))

Ο Διαχειριστής

a. Ο Admin αποκρυπτογραφεί και επαληθεύει το MAC
b. Γίνεται έλεγχος για την πιστοποίηση του ονόματος και του κωδικού
c. Υπογράφεται το Sb= r□(b)d mod n)
d. Αποστέλονται στον ψηφοφόρο τα Sb και MAC(Sb)

Ο Ψηφοφόρος

a. Ο ψηφοφόρος ελέγχει το MAC
b. Εκτελεί το Su= □(b)d mod n)
c. Ελέγχει την υπογραφή του Admin

4. **Εκλογικό Βήμα 2**: Καταχώρηση της ψήφου
   a. Ο ψηφοφόρος στέλνει το Su στον Anon κρυπτογραφημένο με το δημόσιο κλειδί του προγράμματος καταμέτρησης, CPK EC-PK(CS-Key), ECS-Key(□(b)d mod n, b, k1, k2), MAC(ECS-Key((b)d mod n, b, k1, k2))
   b. Ο Anon προωθεί το προηγούμενο μαζί με μια MAC που προκύπτει από το σύνολο του μηνύματος στον καταμετρητή.
5. **Καταμέτρηση**
   a. Ο καταμετρητής αποκρυπτογραφεί τα μηνύματα
   b. Γίνεται έλεγχος των υπογραφών
   c. Δημιουργείται η τελική ψήφος b, ((b)d mod n, k1, k2)
   d. Οι ψήφοι προστίθενται και δημοσιεύονται τα αποτελέσματα
6. **Έλεγχος**
   a. Ο ψηφοφόρος μπορεί τότε να ελέγξει την δική του ψήφο καθώς επίσης και όλες τις υπόλοιπες υπογραφές

Σημειώσεις:

- Το MAC (Message Authentication Code) σε αυτή την περίπτωση αποτελεί ένα hash του μηνύματος.
- Στο σύστημα εάν χρησιμοποιούνται περισσότεροι από 1 Anon, ο καταμετρητής θα ρυθμιστεί να ελέγχει για διπλές ψήφους.
- Εάν χρησιμοποιείται 1 Admin τότε το σύστημα θα δημοσιεύσει μια λίστα από όλα τα ονόματα των ανθρώπων που ψήφισαν
- Εάν χρησιμοποιούνται περισσότεροι από 1 Admin, τότε δημοσιεύουν τον αριθμό των ψήφων που υπέγραψε ο κάθε ένας.
- Τέλος εάν χρησιμοποιούνται πολλαπλοί Admins τότε υποχρεωτικά θα πρέπει να υπογράφεται η κάθε ψήφος τουλάχιστον από το ½ των Admin

Σε κάθε σημείο στις προηγούμενες διαδικασίες υπάρχουν έλεγχοι από τον Εκλογικό Αντιπρόσωπο και στην περίπτωση που παρουσιαστούν τυχόν προβλήματα, τότε ο Εκλογικός Αντιπρόσωπος ενημερώνει τον εκάστοτε διαχειριστή για το μέρος όπου παρουσιάστηκε το πρόβλημα και την μορφή του. Η λειτουργία αυτή, καθιστά εφικτή τη διόρθωση των πιθανών σφαλμάτων σε πραγματικό χρόνο. Αυτό προσδίδει σε όλο το σύστημα ευελιξία στην αντιμετώπιση λαθών και αξιοπιστία στις διαδικασίες.

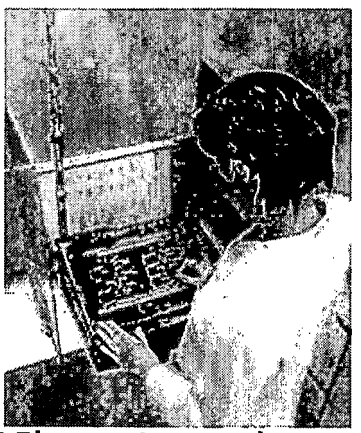
Το EVOX αποτελεί την πιο αξιόπιστη επιλογή σε σύστημα ηλεκτρονικής ψηφοφορίας στον τομέα της μυστικότητας των ψήφων.

H. Association of Electronic Voting Systems (EVS)

Ο Association of Electronic Voting Systems (EVS) είναι μια ομάδα από επιχειρήσεις, που είχαν σχηματιστεί για να αναπτύξουν μια πλήρη λύση για τα συστήματα ηλεκτρονικής ψηφοφορίας στην Ιαπωνία, το 2001. Αρχικά αυτός ο οργανισμός ξεκίνησε σαν ομάδα έρευνας σε συστήματα ηλεκτρονικής ψηφοφορίας το 1989. Το EVS ανέλαβε ένα Ιαπωνικό πρόγραμμα, το οποίο ήταν μέρος του προγράμματος Infrastructure Improvement για το industrial and Social Information Technology και για το 1998 και το 2000. Τα προγράμματα στόχευαν συγκεκριμένα στην ανάπτυξη κατανοητού λογισμικού για συστήματα ηλεκτρονικής ψηφοφορίας. Το πρόγραμμα για το 2000 είχε αναπτυχθεί με σκοπό να δημιουργηθεί μια ολοκλήρωση για την υποστήριξη λειτουργιών για άτομα με ειδικές ανάγκες. Η χρηματοδότηση προερχόταν από τον πλεονάζοντα προϋπολογισμό της κυβέρνησης το 1998 και το 2000.

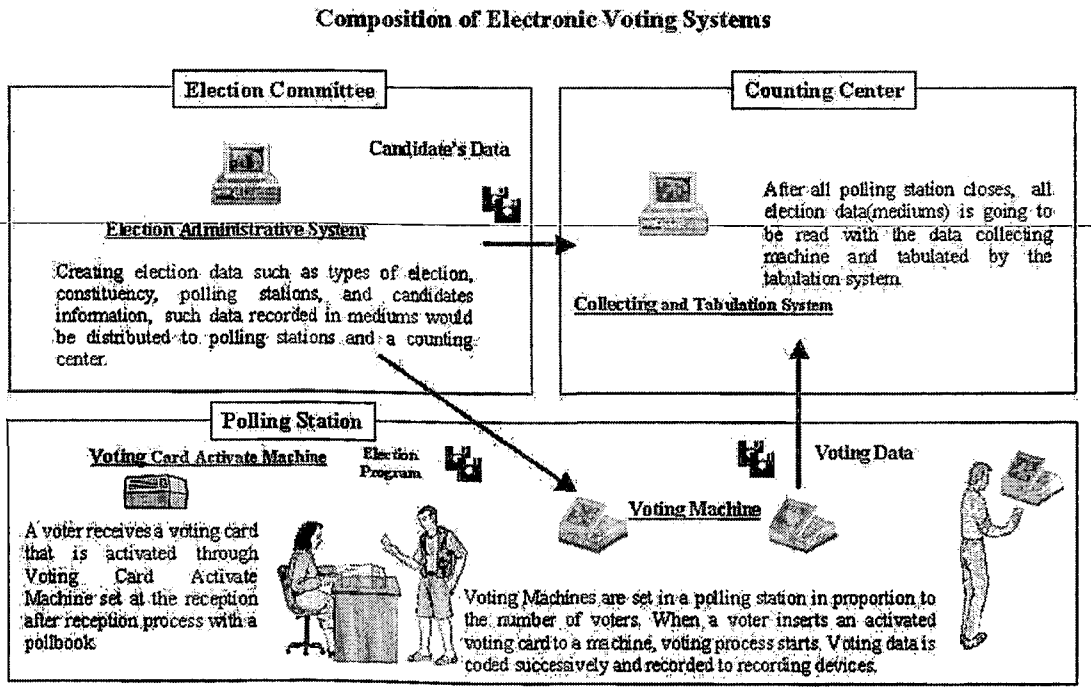
Ο συνδυασμός των πραγματικών εκλογών και των επιδείξεων που υλοποιήθηκαν από το EVS έφτασε τον αριθμό των συμμετεχόντων στο ύψος των 79000 ατόμων. Για την ακρίβεια το EVS χρησιμοποιήθηκε σε εκλογές στην πόλη Kawaguchi, στη Saitama το 1998 και στην εκλογή δημοτικού συμβουλίου το 1999 στην πόλη Kochi. Η εκλογική διαδικασία Community Power είχε υλοποιηθεί στο Norwich της Αγγλίας το 1999. Όταν έγιναν οι εκλογές στις πόλεις Kyushu και Okinawa, το 2000 το EVS σύστημα ηλεκτρονικής ψηφοφορίας παρουσιάστηκε στους ηγέτες των συμμετεχόντων χωρών και σε άλλους συμμετέχοντες. Όλοι αυτοί οι συμμετέχοντες χρησιμοποίησαν αυτό το σύστημα για να εκτελέσουν τις συνεδρίες ψηφοφορίας τους.

Μια ακόμα σημαντική στιγμή στην μέχρι τώρα πορεία του συστήματος έγινε στις 6 Ιουλίου το 2003 όπου και υλοποιήθηκε το σύστημα για τις εκλογές του συμβουλίου της πόλης Fukui. Όλες οι εκλογικές διαδικασίες έγιναν με απόλυτη ορθότητα και τα αποτελέσματα ήταν ορθά και γρήγορα. Αυτή η δεδομένη ψηφοφορίας, ήταν ο σταθμός που χαρακτήρισε το σύστημα EVS ως ένα σύστημα επαρκή και αξιόπιστο. Το EVS συνεχίζει να έχει μια κατανοητή υποστήριξη των εκλογικών διαδικασιών και υπηρεσιών, με τα τελευταίας λέξης συστήματα ηλεκτρονικής ψηφοφορίας που παρέχει.



Σχήμα 8.Η.1:Σύστημα ηλεκτρονικής ψηφοφορίας EVS

Έχοντας λοιπόν όλη αυτήν την εμπειρία, από την πλευρά των προγραμματιστών, κατανοούμε ότι το EVS είναι ένα από τα σημαντικότερα ολοκληρωμένα συστήματα ηλεκτρονικής ψηφοφορίας που υπάρχουν και αναπτύσσονται την σημερινή εποχή. Η τελευταία παρουσίαση του EVS είναι το μοντέλο VT-25. Αυτό αποτελείται από μια οθόνη αφής 15 ιντσών, μέσω της οποίας επιτελούνται όλες οι διαδικασίες σε μια συνεδρία ψηφοφορίας. Ο σχεδιασμός των οθονών που χρησιμοποιεί ο ψηφοφόρος, έχει γίνει με γνώμονα την παρουσίαση του γεγονότος με τον πιο εύκολο τρόπο. Χρησιμοποιείται σύστημα φωνητικής καθοδήγησης και ειδικό πληκτρολόγιο επιλογών για τα άτομα με ειδικές ανάγκες.



I. Infopoll.com

Η επιχείρηση Infopoll.com, ξεκίνησε να λειτουργεί το 1995 και από τότε αποτελεί μια από τις πρωτοπόρες επιχειρήσεις στον τομέα της παροχής εργαλείων για την υποστήριξη της λήψης αποφάσεων. Το κοινό στο οποίο απευθύνεται ποικίλει από μεμονωμένους ιδιώτες μέχρι μεγάλες επιχειρήσεις και φορείς τοπικής αυτοδιοίκησης. Από το 1995 περισσότεροι από 20000 χρήστες σε 75 χώρες έχουν χρησιμοποιήσει τις on-line τεχνολογίες δημοσκόπησης, της infopoll. Το κοινό που έχει κατά καιρούς χρησιμοποιήσει το λογισμικό της εταιρίας, προέρχεται σχεδόν από κάθε μέρος της βιομηχανίας. Η Infopoll έχει πελάτες από τραπεζικούς οργανισμούς, Ασφαλιστικούς οργανισμούς, επιχειρήσεις έρευνας αγοράς, κυβερνητικούς φορείς, διαδικτυακές επιχειρήσεις, αφιλοκερδής οργανισμούς και εκπαιδευτικά ιδρύματα.

Το λογισμικό που παρέχει η επιχείρηση για την υποστήριξη των υπηρεσιών δημοσκόπησης που παρέχει αποτελείται από 2 προγράμματα. Τα "Infopoll Σχεδιαστής" και "Infopoll Εξυπηρετητής". Ο Infopoll Σχεδιαστής δημιουργεί φόρμες δημοσκόπησης επαγγελματικών προδιαγραφών και ο Infopoll Εξυπηρετητής συλλέγει σε πραγματικό χρόνο τα δεδομένα και τα αναλύει.

Infopoll Designer

Ο Infopoll Σχεδιαστής, όπως αναφέραμε είναι υπεύθυνος για τον σχεδιασμό των απαραίτητων φορμών σε κάποια δημοσκόπηση. Με το λογισμικό αυτό κάθε επιχείρηση ή οργανισμός που επιθυμεί να πραγματοποιήσει μια έρευνα – δημοσκόπηση μπορεί σε λίγα λεπτά να κατασκευάσει το ερωτηματολόγιό του και να το διαθέσει στο κοινό για να συλλέξει τις προτιμήσεις του.

Πλεονεκτήματα του Infopoll Σχεδιαστή

- Υποστηρίζει όλες τις Win32
- Εύκολες διεπαφές χρήστη για μέγιστη ταχύτητα και αποδοτικότητα.
- Υποστηρίζει επιλογές copy&paste και drag&drop
- Δεν είναι απαραίτητη η γνώστη HTML ή JavaScript
- Υπάρχει πολύ-γλωσσική υποστήριξη για παγκόσμιες δημοσκοπήσεις
- Ενσωματωμένος ορθογραφικός έλεγχος με πολλαπλά λεξιλόγια
- Ενσωματωμένος browser για σχεδιασμό WYSIWYG
- Ενσωματωμένες ιδιότητες ελέγχου ορθότητας εισαγομένων κειμένων και επιλογών
- Πεδίο εισαγωγής διεύθυνσης E-mail με δυνατότητα ελέγχου ορθότητας
- Δωρεάν χρήση

Infopoll Server

Ο Infopoll Σχεδιαστής είναι η πρώτη Internet- enabled μηχανή δημοσκόπησης και αποτελεί την αρτιότερη υλοποίηση σε συστήματα συλλογής, ανάλυσης και παρουσίασης επιχειρηματικών πληροφοριών στον κόσμο. Έχει σχεδιαστεί για να παρέχει υψηλή ασφάλεια και έλεγχο κατά τις φάσεις συλλογής και επεξεργασίας των

δεδομένων σε πραγματικό χρόνο. Η παρούσα έκδοση ονομάζεται Server 7.0 και είναι διαθέσιμη σε περιβάλλον Windows και Linux, για να μπορεί να ικανοποιήσει όλες τις πιθανές ανάγκες των οργανισμών.

Πλεονεκτήματα του Infopoll Εξυπηρετητή

- Αποδεδειγμένη Τεχνολογία
- Κεντροποιημένη αποθήκευση και διαχείριση των αποτελεσμάτων
- Ευκολία στην εγκατάσταση
- Μεγάλη επεκτασιμότητα δημοσκοπήσεων
- Δεν χρειάζεται διαχείριση
- Προσβάσιμο μέσω του διαδικτύου
- Παράδοση αποτελεσμάτων σε πραγματικό χρόνο

Λειτουργία του Infopoll Εξυπηρετητή

1. Οι χρήστες χρησιμοποιούν τον Infopoll Designer για να δημιουργήσουν τις φόρμες για την δημοσκόπηση που επιθυμούν.
2. Οι χρήστες δημοσιοποιούν την δημοσκόπησή τους online. Ο Infopoll Server τότε ελέγχει τον λογαριασμό χρήστη, δημιουργεί δυναμικά τις βάσεις για τη δημοσκόπηση και ενημερώνει τον χρήστη ότι η δημοσκόπησή του έχει ξεκινήσει.
3. Ο Infopoll Server συλλέγει και αποθηκεύει τα δεδομένα. Ακολούθως, ανανεώνει δυναμικά τη βάση δεδομένων του με τα αποτελέσματα.
4. Αναλύει τα δεδομένα της δημοσκόπησης και εκτελεί μια στατιστική ανάλυση στα δεδομένα που βρίσκονται στην βάση του
5. Δημιουργείται μια αναφορά από τα αποτελέσματα και δημοσιεύεται στο διαδίκτυο.

Το λογισμικό της Infopoll έχει αναπτυχθεί με τέτοιο τρόπο ώστε να υποστηρίζει πολλές βάσεις δεδομένων για την αποθήκευση των δεδομένων της κάθε δημοσκόπησης. Μπορεί να αποθηκεύει τα δεδομένα αυτά σε MS SQL Server, Oracle και Interbase. Εκτός από τις δυνατότητες αποθήκευσης σε αυτές τις βάσεις, μπορεί να τις προσπελαύνει και να τις επεξεργάζεται δυναμικά για να βγάζει τα αποτελέσματα. Το γεγονός ότι αυτή η επιχείρηση έχει κατά καιρούς υποστηρίξει πάρα πολλές από τις σημαντικότερες επιχειρήσεις στην λήψη των αποφάσεών τους, την καθιστά μια από τις πληρέστερες επιχειρήσεις στον τομέα δημοσκοπήσεων.

J. Votenet Solutions Inc.

Η Votenet Solutions Inc. είναι μια εταιρία που προμηθεύεται λογισμικό e-democracy και επικεντρώνεται κυρίως στην ανάπτυξη τεχνολογιών online, εκπαιδευτικών προγραμμάτων και υλικού που σχετίζεται με εκλογές, εγγραφές ψηφοφόρων και καταμετρήσεις ψήφων. Η Votenet Solutions Inc αποτελείται από ένα σύνολο από επιχειρησιακές μονάδες που περιλαμβάνουν:

Online λύσεις ψηφοφορίας και εκλογική συμβούλευση

Η Votenet είναι ο αναγνωρισμένος πρωτοπόρος στην ηλεκτρονική ψηφοφορία και στα συστήματα ηλεκτρονικής ψηφοφορίας που είναι πλήρως διαχειριζόμενα από τους πελάτες. Το τμήμα αυτό έχει εκδώσει τα προϊόντα eduBallot, VOTENET, eBallot Association Edition, eBallot University Edition και eBallot Realtor Association Edition. Επιπλέον έχουν αναπτυχθεί συστήματα που υποστηρίζουν online εκλογικές διαδικασίες για μη-κερδοσκοπικούς οργανισμούς.

Political / Advocacy Λογισμικό

Το Votenet αποτελεί πρωτοπόρο επιχείρηση στο λογισμικό εγγραφής ψηφοφόρων. Το Election Impact είναι η νούμερο 1 λύση στην αγορά για τους συνετερισμούς και τους συμβουλευτικούς οργανισμούς. Το Absentee Impact είναι το πρώτο σύστημα εγγραφής υποψηφίων που υπολογίζει αυτόματα την αποχή από την εκλογική διαδικασία. Αυτή η επιχειρησιακή μονάδα προσφέρει μια πληθώρα προγραμμάτων συμβουλευτικού χαρακτήρα, στο θέμα της καμπάνιας για την επιβολή κάποιου προτεινόμενου νομοσχεδίου.

Πολιτικές Δημοσιεύσεις

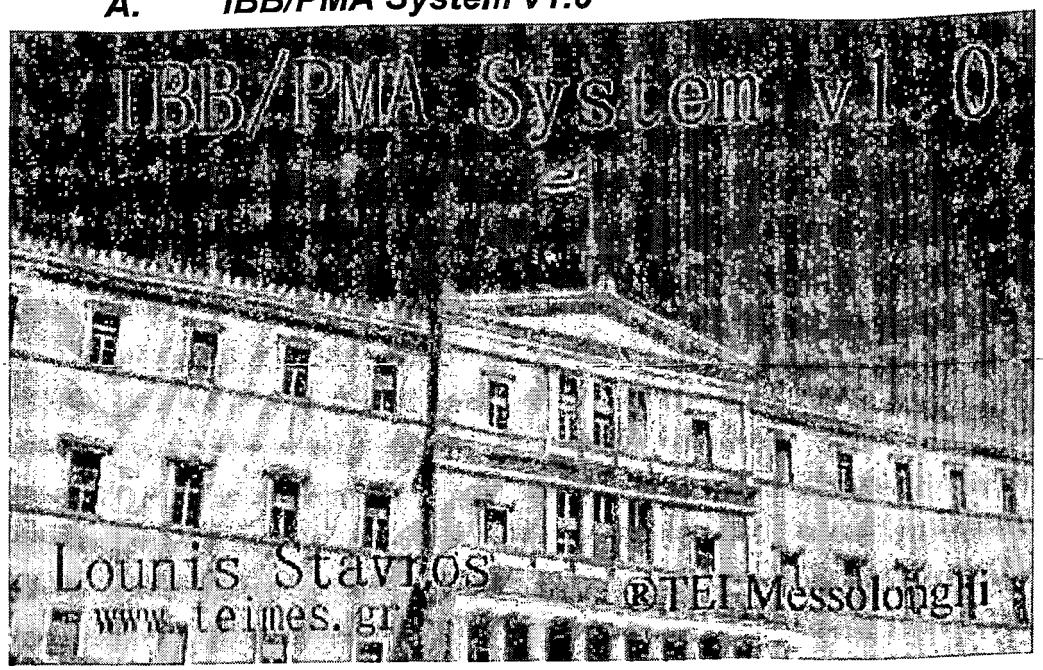
Το Votenet είναι ο εκδότης του περιοδικού "Campaigns & Elections", που είναι το μοναδικό περιοδικό που καλύπτει θέματα που αφορούν τις επιχειρήσεις και τις διαδικασίες μιας πολιτικής εκστρατείας. Επιπλέον το Votenet εκδίδει και τον ετήσιο οδηγό του Κογκρέσου και το U.S Congress Handbook.

Τα δύο σημαντικότερα προιόντα της επιχείρησης Votenet είναι το "VOTENET: On Demand Web Elections" και το " VOTENET: eBallot ". Το onDemand έχει στόχο του, τις μικρού μεγέθους επιχειρήσεις που επιθυμούν να διεξάγουν μια ηλεκτρονική ψηφοφορία. Τα πλεονεκτήματα που παρέχει περιλαμβάνουν την εκκίνηση της ψηφοφορίας μέσα σε λίγα λεπτά, το φθηνό κόστος, την εκτέλεση μεγάλου αριθμού ταυτόχρονων ψηφοφοριών και την εποπτεία από κάποιον τρίτο έμπιστο οργανισμό.

Το eBallot απευθύνεται σε μεγαλύτερες επιχειρήσεις, οργανισμούς και πανεπιστήμια, τα οποία επιθυμούν πιο σύνθετες και πιο ασφαλείς εκλογικές διαδικασίες. Σε αυτήν την περίπτωση παρέχονται δικαιώματα ανάλογα με την έκταση της ηλεκτρονικής ψηφοφορίας και το εύρος των δυνατοτήτων είναι σημαντικά αυξημένο. Παρέχεται η δυνατότητα ενσωμάτωσης βιογραφιών των υποψηφίων, η αυτόματη καταμέτρηση αποτελεσμάτων και ακόμα μεγαλύτερη ασφάλεια, της τάξεων των 128bit. Η διαφορά της Votenet Solutions Inc. από τις υπόλοιπες επιχειρήσεις έγκειται στο γεγονός ότι, η VOTENET παρέχει μεγαλύτερη γκάμα προϊόντων τα οποία μπορεί να χρησιμοποιήσει οποιοσδήποτε οργανισμός, ανάλογα με τις απαιτήσεις του.

8. IBB/PMA Electronic Voting System

A. *IBB/PMA System v1.0*



Το IBB/PMA Electronic Voting System αποτελεί μια πρόταση καινοτομία στα συστήματα ηλεκτρονικής ψηφοφορίας. Είναι ένα σύστημα που μπορεί να διεξάγει εκλογικές διαδικασίες μικρής (π.χ Εκλογές ανάδειξης προέδρου σε οργανισμούς) και μεσαίας κλίμακας (π.χ Εκλογές ανάδειξης νομάρχη σε κάποιο νομό). Με κάποιες συμβάσεις και τροποποιήσεις μπορεί να αναλάβει και την διεξαγωγή εκλογικών διαδικασιών μεγάλης κλίμακας (π.χ Εκλογές ανάδειξης πολιτικής ηγεσίας σε κάποιο κράτος).

Ο σχεδιασμός του το καθιστά ασφαλές και διαυγές, κατάλληλο για εκλογικές διαδικασίες. Το σύστημα αποτελεί μια ψηφιακή μεταφορά του κλασσικού συστήματος ψηφοφορίας, με τις ανάλογες τροποποιήσεις για να μπορεί να υλοποιηθεί η μετάβαση σε ψηφιακή μορφή. Το σύστημα υποστηρίζει όλες τις ιδιότητες που έχουν οριστεί ότι πρέπει να υποστηρίζει ένα σύστημα ηλεκτρονικής ψηφοφορίας.

Έχει ενσωματώσει πολλές ιδιότητες από τα προϋπάρχοντά του συστήματα και έχει επεκτείνει αρκετές από τις ιδιότητές τους. Κάποιες από τις ιδιότητες αυτές έχουν τροποποιηθεί ή απορριφθεί, γιατί δεν ταιριάζουν στο σκοπό του συστήματος.

Επιπλέον εισάγει την τεχνική Physical Multiple Administrators (PMA), την τεχνική Identical Ballot Boxes (IBB) και την διαδικασία Επαλήθευσης Ψήφου. Ο συνδυασμός αυτός οδηγεί σε μεγαλύτερα σημεία ασφάλειας, αναλογικά με τις ήδη υπάρχουσες λύσεις ηλεκτρονικής ψηφοφορίας.

9. Ιδιότητες του συστήματος ηλεκτρονικής ψηφοφορίας IBB/PMA

Α. Συμβατότητα του προτεινόμενου συστήματος με τις ιδιότητες

Μελετώντας τις ιδιότητες που πρέπει να έχει ένα σύστημα ηλεκτρονικής ψηφοφορίας για να θεωρείται εφαρμόσιμο καταλήγουμε στο συμπέρασμα, ότι και το υπό ανάπτυξη σύστημα θα πρέπει να συμβαδίζει με αυτές τις ιδιότητες. Το σύστημα ηλεκτρονικής ψηφοφορίας IBB/MPA System, ολοκληρώνει όλες τις προαναφερθείσες ιδιότητες στο βαθμό που κρίνεται απαραίτητος, για την ορθή του λειτουργία και τους στόχους κατασκευής του. Θα εξετάσουμε την ολοκλήρωση των ιδιοτήτων ιεραρχικά, όπως παρουσιάστηκαν μέχρι στιγμής.

Ακρίβεια: Το IBB/MPA System ολοκληρώνει διεργασίες και μηχανισμούς που διασφαλίζουν την ύπαρξη ακρίβειας στο σύστημα. Με μια σειρά από κωδικοποιήσεις και ελέγχους στην επικοινωνία του συστήματος με τον ψηφοφόρο και στην μεταφορά της ψήφου από το μηχάνημα του ψηφοφόρου στις κάλπες του συστήματος, διασφαλίζεται ότι το σύστημα είναι αδιάβλητο στην τροποποίηση της ψήφου κατά τη μεταφορά της. Από τη στιγμή που η ψήφος εισέρχεται στο σύστημα και αποθηκεύεται στις βάσεις - κάλπες, ισχύει μια άλλη σειρά από μηχανισμούς για την προστασία της ψήφου. Ένα σύνολο από προγράμματα εποπτείας συστημάτων εκτελούνται στο παρασκήνιο καθ' όλη τη διάρκεια της ψηφοφορίας και ελέγχουν το αδιάβλητο του συστήματος, καταγράφοντας και επισημαίνοντας οποιαδήποτε ανωμαλία. Το IBB/PMA System έχει τη δυνατότητα να καταγράφει όλες τις αλλαγές που μπορεί να προκύψουν από τους διαχειριστές του.

Άμεσο αποτέλεσμα αυτού είναι η ικανότητα του συστήματος να μπορεί να αναγνωρίσει κάποια πιθανή προσπάθειας αλλοίωσης ψήφου και να μπορεί να την επαναφέρει στο αρχικό της. Επιπλέον, μια από τις δύο καινοτομίες που εισάγει το σύστημα είναι οι Identical Ballot Boxes (IBB). Σύμφωνα με αυτόν τον μηχανισμό οι κάλπη που κρατάει την ψήφο αποτελείται από ένα σύνολο από διαφορετικές κάλπες, που περιέχουν ακριβώς τα ίδια δεδομένα. Αυτές οι κάλπες ψηφοφορίας περιέχουν τις ίδιες ψήφους και με την ίδια σειρά, είναι δηλαδή πανομοιότυπες. Επιπλέον δεν βρίσκονται φυσικά στο ίδιο μηχάνημα, αλλά σε απομακρυσμένα μηχανήματα, με διαχειριστές που δεν έχουν επικοινωνία μεταξύ τους και πρόσβαση ο ένας, στο μηχάνημα του άλλου. Έτσι, ακόμα και εάν υπάρξει κάποια προσπάθεια αλλοίωσης της ψήφου, θα φανεί από τις υπόλοιπες "αδελφές" κάλπες. Ο συνδυασμός λοιπόν των IBBs με τα εργαλεία εποπτείας, καθιστούν το σύστημα ασφαλές από πιθανές τροποποιήσεις ψήφων, από οποιαδήποτε προέλευση.

Το IBB/MPA System δεν μπορεί να διαγράψει κάποια ψήφο από τη στιγμή που έχει εισαχθεί στις κάλπες. Είναι δεδομένο ότι από τη στιγμή που θα γίνει επιτυχώς η καταχώρηση της ψήφου στις κάλπες, θα περάσει και στην φάση της καταμέτρησης. Η διατήρηση της ψήφου ανέπαφής κατά τη διάρκεια της ψηφοφορίας και της μετάβασής της στην φάση της καταμέτρησης, διασφαλίζεται και αυτή από το σύστημα των IBBs.

Τέλος, όσον αφορά το θέμα της ακρίβειας, η ιδιότητα συμπληρώνεται μέσω μιας ακόμα λειτουργίας που ενσωματώθηκε στο σχεδιασμό του συστήματος. Το τελευταίο μέρος θέλει το σύστημα ηλεκτρονικής ψηφοφορίας να μην επιτρέπει σε μια

άκυρη ψήφο να καταμετρηθεί στο τελικό αποτέλεσμα. Αυτό επιτυγχάνεται στο σύστημά μας μέσω της λειτουργίας της μη-αποδοχής άκυρης ψήφου. Στο σύστημα ηλεκτρονικής ψηφοφορίας IBB/PMA System δεν δίνεται η δυνατότητα για είσοδο άκυρης ψήφου. Η διεπαφή που χρησιμοποιείται για την ψηφοφορία παρέχει το δικαίωμα ψήφου ανάμεσα σε συγκεκριμένες και προκαθορισμένες επιλογές. Και μετά από εκτενή μελέτη σε θέματα ψηφοφοριών έχει οριστεί ότι η επιλογή άκυρης ψήφου δεν θα είναι δεκτή. Η χρήση των ανωτέρων μηχανισμών καθιστά το IBB/PMA System ένα σύστημα ηλεκτρονικής ψηφοφορίας που υποστηρίζει την ιδιότητα της Ακρίβειας.

Έχοντας διασφαλίσει την, ίσως σημαντικότερη από όλες, ιδιότητα της Ακρίβειας, προχωράμε στην ιδιότητα της Δημοκρατικότητας. Το σύστημά μας κατά την προετοιμασία του δέχεται σαν είσοδο μια λίστα από νόμιμους ψηφοφόρους. Η δημιουργία της λίστας και η παράδοσή της γίνεται από τα άτομα που είναι υπεύθυνα για την ψηφοφορία στην κατά τόπους περιοχή όπου εφαρμόζεται το σύστημα. Αυτή η λίστα περιέχει όλα τα απαιτούμενα στοιχεία των νόμιμων ψηφοφόρων και μέσα από αυτή γίνεται η αναγνώριση των ψηφοφόρων και τους επιτρέπεται η ψηφοφορία. Εάν κάποιος ζητήσει από το σύστημα να ψηφίσει, τότε αυτό ψάχνει την λίστα μέχρι να βρει εάν ο χρήστης αυτός είναι καταχωρημένος. Στην περίπτωση που είναι καταχωρημένος σημαίνει ότι είναι νόμιμος ψηφοφόρος και κατά συνέπεια μπορεί να ψηφίσει. Εάν δεν βρεθεί κάποια εγγραφή που να αντιπροσωπεύει τον χρήστη, τότε χαρακτηρίζεται μη – έγκυρος και δεν του επιτρέπεται η πρόσβαση στο σύστημα ψηφοφορίας.

Με τη χρήση λοιπόν της λίστα νόμιμων ψηφοφόρων διασφαλίζεται ότι πρόσβαση στο σύστημα και στην ψηφοφορία θα έχουν μόνο όσοι έχουν δικαίωμα ψήφου. Το δεύτερο σκέλος της ιδιότητας της Δημοκρατικότητας που πρέπει να διασφαλίζεται από ένα σύστημα είναι, το να μην μπορεί κάποιος να ψηφίσει περισσότερες από μια φορές. Η λίστα των νόμιμων ψηφοφόρων έχει σχεδιαστεί να περιέχει ένα ακόμα πεδίο, το οποίο θα ενημερώνεται αυτόματα κατά την ψηφοφορία ενός πολίτη. Από την στιγμή που θα ζητήσει πρόσβαση στο σύστημα ψηφοφορίας και θα του δοθεί, η εγγραφή στο πεδίο εκείνο θα τροποποιείται και θα παίρνει τιμή που θα δείχνει ότι ο συγκεκριμένος ψηφοφόρος έχει ήδη ασκήσει το εκλογικό του δικαίωμα. Έτσι στην προσπάθεια επανασύνδεσης του δεδομένου χρήστη, σε μια απόπειρα να ψηφίσει περισσότερο από μια φορά, η βάση, ελέγχοντας το συγκεκριμένο πεδίο, θα του απορρίπτει την αίτηση και θα τον αποσυνδέει από το σύστημα. Με αυτούς τους μηχανισμούς διασφαλίζεται και η ιδιότητα της Δημοκρατικότητας στο IBB/PMA System.

Εφόσον και η ιδιότητα της δημοκρατικότητας έχει ολοκληρωθεί με τους αντίστοιχους μηχανισμούς στο σύστημά μας, προχωράμε στην ιδιότητα της μυστικότητας. Στο IBB/MPA System, έχει ενσωματωθεί ένας μηχανισμός επικοινωνίας και κρυπτογράφησης που δεν επιτρέπει σε κανέναν στο σύστημα να ταιριάζει την ψήφο με τον ψηφοφόρο. Η επικοινωνία του ψηφοφόρου με το σύστημα γίνεται μέσω ενός τρίτου έμπιστου εξυπηρετητή, διασφαλίζονται ότι σε κανένα σημείο στο σύστημα δεν υπάρχει ταυτόχρονα η ψήφος και ο ψηφοφόρος.

Με αυτόν τον μηχανισμό ασφαλείας, το σύστημα καθίσταται έμπιστο στο θέμα της ταυτοποίησης ψηφοφόρου με ψήφο. Επιπλέον η ιδιότητα της μυστικότητας θέλει να μην μπορεί ο ψηφοφόρος να αποδείξει ότι ψήφισε με κάποιον συγκεκριμένο τρόπο. Στο σύστημά μας δεν υπάρχει η δυνατότητα να λάβει κάποιο αποδεικτικό της ψήφου του, ο ψηφοφόρος σε μορφή τέτοια που να μπορεί να το απομακρύνει από το σύστημα και να το χρησιμοποιήσει όπως αυτός επιθυμεί. Σε όλα τα στάδια της ψηφοφορίας, ο ψηφοφόρος έχει τα αντίστοιχα δικαιώματα με αυτά του παραδοσιακού

τρόπου ψηφοφορίας. Τέλος η ιδιότητα της μυστικότητας, εκτός από τους δύο προηγούμενους παράγοντες θέλει τις ψήφους μυστικές μέχρι το τέλος της διαδικασίας ψηφοφορίας. Στο σύστημα αυτό διασφαλίζεται με την τεχνική της κρυπτογράφησης. Όλες οι ψήφοι εισάγονται στο σύστημα και αποθηκεύονται με τέτοια μορφή ώστε να μην είναι αναγνώσιμες από κάποιον άνθρωπο, ούτε και από το ίδιο το σύστημα κατά τη διάρκεια της ψηφοφορίας. Οι ψήφοι έχουν υπόσταση κώδικα και η αποκρυπτογράφησή τους δεν δείχνει κάτι σωστό, παρά μόνο όταν χρησιμοποιηθεί το αντίστοιχο κλειδί, το οποίο δημιουργείται μόνο κατά την έναρξη και λήξη της διαδικασίας της ψηφοφορίας.

Έχοντας τεκμηριώσει και την ολοκλήρωση της ιδιότητας της μυστικότητας στο σύστημά μας, ακολουθεί η ιδιότητα της επαληθευσιμότητας. Αυτή η ιδιότητα θέλει τον κάθε πολίτη- ψηφοφόρο να μπορεί να επαληθεύσει την ψήφο του και ότι αυτή καταμετρήθηκε σωστά. Αυτή η τεχνική μπορεί να υλοποιηθεί σε συστήματα που χρησιμοποιούν blind – signatures από τους ψηφοφόρους στις ψήφους.

Το IBB/PMA System δεν κάνει χρήση αυτού του μηχανισμού, αλλά ολοκληρώνει αυτήν την ιδιότητα με άλλον τρόπο. Έχει ενσωματωθεί ένας μηχανισμός, ο οποίος ταιριάζει την ψήφο με τον ψηφοφόρο μόνο όταν αυτός το επιθυμεί και μόνο μετά το πέρας της διαδικασίας της ψηφοφορίας. Το ταίριασμα αυτό δεν αναιρεί τις προηγούμενες ιδιότητες γιατί γίνεται μέσω μηχανισμού είναι συμβατός με τις ιδιότητες αυτές. Η ολοκλήρωση της επαληθευσιμότητας γίνεται μέσω μιας εσωτερικής διαδικασίας, σύμφωνα με την οποία, μετά το πέρας της ψηφοφορίας το σύστημα επιλέγει από τους ψηφοφόρους έναν αριθμό από αυτούς που έχουν ψηφίσει. Η εκλεκτική αρχή τότε καλεί τους ψηφοφόρους που επιλέχθηκαν και τους ζητάει να επαληθεύσουν την ψήφο τους. Αυτοί με τη σειρά τους εισέρχονται σε ειδικά τερματικά και εισάγοντας τον προσωπικό τους κωδικό ξεκλειδώνουν την ψήφο τους και τους παρουσιάζεται. Βέβαια είναι κατανοητό ότι και σε αυτή τη φάση της επαλήθευσης οι δεδομένοι ψηφοφόροι δεν μπορούν να πάρουν κάποιο αποδεικτικό της ψήφου τους.

Ακολούθως έρχεται η ιδιότητα της Ευκολίας. Ο σχεδιασμός του IBB/PMA System έχει γίνει με τέτοιο τρόπο που να μην προκαλεί κάποιο πρόβλημα στους χρήστες του. Οι επιλογές είναι δεδομένες και ο χρήστης δεν μπορεί να κάνει κάποιο εσκεμμένο ή αθέμιτο λάθος που να οδηγήσει σε ανεπιτυχή ψηφοφορία. Κατά τη διάρκεια της ψηφοφορίας, χρησιμοποιούνται γραμματοσειρές και πλαίσια που είναι ευδιάκριτα από τον μέσο άνθρωπο. Το σύστημα στη χρήση του δεν αποτελεί σύστημα που απαιτεί ειδική γνώση από τους χρήστες του, καθώς είναι όλα θέμα μερικών μόνο επιλογών.

Επιπλέον στον σχεδιασμό έχουν ενσωματωθεί και ιδιότητες που απευθύνονται σε άτομα ηλικιωμένα και σε άτομα που παρουσιάζουν ειδική κατηγορίας (π.χ άνθρωποι με αχρωματοψία). Τέλος εκτός από την ευκολία στην χρήση, το σύστημα ολοκληρώνει και την δεύτερη απαίτηση της ιδιότητας της Ευκολίας. Το σύστημα δεν απαιτεί κάποιο εξειδικευμένο ή ισχυρό υπολογιστικό σύστημα για να εκτελεστεί από την πλευρά του ψηφοφόρου. Το μόνο που χρειάζεται είναι ένας συμβατικός υπολογιστής με πρόσβαση στο διαδίκτυο για να εισέλθει και χρησιμοποιήσει το σύστημα για να ασκήσει τα εκλογικά του δικαιώματα στη δεδομένη ψηφοφορία.

Το τελευταίο από τις σημαντικές ιδιότητες για τα συστήματα ηλεκτρονικής ψηφοφορίας είναι η Ευελιξία, η οποία θέλει ένα τέτοιο σύστημα να μπορεί να διαχειρίζεται ταυτόχρονα έναν αριθμό από διαφορετικές διαδικασίες ψηφοφορίας. Αυτή η ιδιότητα δεν ενσωματώνεται στην παρούσα φάση ανάπτυξης στο IBB/PMA System. Ο λόγος που επιλέχθηκε να μην χρησιμοποιείται είναι ότι το σύστημα έχει σκοπό να εξυπηρετεί μια διαδικασία ψηφοφορίας την φορά και να στήνεται ειδικά για

αυτή. Θεωρείται ότι αυτή η ιδιότητα, απευθύνεται κατά κύριο λόγο στα συστήματα που διαχειρίζονται δημοσκοπήσεις και πρέπει να μπορούν να θέσουν πολλά ερωτήματα σε κάθε συνεδρία ψηφοφορίας. Το IBB/PMA System έχει κατασκευαστεί για χρήση σε εκλογικές διαδικασίες, στις οποίες θα διεξάγεται μια κάθε φορά.

Με την ανάλυση και της Ευελιξίας κλείνει ο κύκλος των σημαντικών ιδιοτήτων των συστημάτων ηλεκτρονικής ψηφοφορίας και της ολοκλήρωσής τους στο IBB/PMA System. Βέβαια υπάρχουν ακόμα και οι ιδιότητες που τελούν υπό συζήτηση και όμως είναι και αυτές βασικές. Για την κατασκευή του IBB/PMA System έχουν ληφθεί και αυτές υπ' όψη. Αρχικά θα αναφερθούμε στην ιδιότητα του παραλληλισμού με το παραδοσιακό σύστημα ψηφοφορίας.

Το σύστημα είναι συμβατό με τον τρόπο που χρησιμοποιείται μέχρι τώρα στην κλασσική μορφή ψηφοφορίας. Έχουν υιοθετηθεί οι κλασσικές φάσεις στη διαδικασία ψηφοφορίας, η πιστοποίηση του ψηφοφόρου, η ψήφος, η καταμέτρηση και η δημοσίευση των αποτελεσμάτων ως επίσης και η μορφή οπτικής εμφάνισης των ψήφων έχει παραμείνει η ίδια. Ο λόγος που χρησιμοποιούνται οι ίδιες φάσεις στην ψηφοφορία, καθώς επίσης και η ίδια μορφή οπτικής εμφάνισης της ψήφου είναι:

Πρώτον ότι υπάρχει συμβατότητα με την ολοκλήρωση και των υπολοίπων ιδιοτήτων των συστημάτων ηλεκτρονικής ψηφοφορίας και

Δεύτερον ότι εφόσον αυτές οι φάσεις και η μορφή ψήφου χρησιμοποιείται μέχρι τώρα, θα υπάρχει ένα εύκολο στάδιο μετάβασης και προσαρμογής στο καινούργιο σύστημα, γιατί φαινομενικά στους χρήστες δεν θα υπάρχει κάποια αλλαγή ή παρέκκλιση από αυτά που έχουν συνηθίσει να χρησιμοποιούν.

Ακολούθως της ιδιότητας του παραλληλισμού με τον παραδοσιακό τρόπο ψηφοφορίας έρχεται η ιδιότητα της ισοτιμίας των υποψηφίων. Στο σύστημα, δεν επιτρέπονται διακρίσεις μεταξύ των υποψήφιων. Όλοι εισάγονται και παρουσιάζονται με τον ίδιο τρόπο. Η αναφορά και εύρεση του κατάλληλου ψηφοδελτίου γίνεται αλφαβητικά και η εμφάνισή του δεν θα διαφέρει από τον ένα υποψήφιο στον άλλο. Στο θέμα της ομοιόμορφης εμφάνισης των υποψηφίων εισάγεται και η χρήση των προτύπων ψήφου, σύμφωνα με τα οποία, η εικονική ψήφος θα δημιουργείται αυτόματα και θα παρέχει όμοιο αποτέλεσμα για όλους τους υποψηφίους. Στο σύστημα η ψήφος έχει την μορφή μιας σελίδας που φέρει το σήμα της υποψήφιας παράταξης και ακολούθως τα μέλη που δηλώνουν υποψηφιότητα σε αλφαβητική σειρά. Η μορφή αυτή έχει επιλεχθεί να είναι κοινή για όλους έτσι ώστε να μην δημιουργείται πρόβλημα διαχωρισμού και κατηγοριοποίησης των υποψηφίων και να διατηρείται το επίπεδο εμπιστοσύνης μεταξύ των ψηφοφόρων και του συστήματος.

Μετά υπάρχει μια παρεμφερής ιδιότητα με την προηγούμενη η οποία θέλει να μην υπάρχει καμιάς μορφής προπαγάνδα από τους υποψήφιους ή από το ίδιο το σύστημα. Η διασφάλιση αυτής της ιδιότητας γίνεται μέσω των προτύπων που ακολουθούνται στην δημιουργία των ψήφων. Κανένας υποψήφιος δεν έχει το δικαίωμα να εισάγει στο σύστημα τίποτα παραπάνω από αυτά που πρέπει. Επιπλέον δεν είναι δυνατή η εισαγωγή κάποιου προεκλογικού μηνύματος «τελευταίας στιγμής» προς τους μελλοντικούς ψηφοφόρους, γιατί επιθυμούμε ο ψηφοφόρος κατά την διάρκεια της ψηφοφορίας να μην δέχεται εξωτερικές πηγές ερεθισμάτων που να μπορούν να του μεταβάλλουν υποσυνείδητα την εκλογική του απόφαση. Ο ψηφοφόρος θα πρέπει να χρησιμοποιεί το σύστημα μόνο σαν μέσω άσκησης του εκλογικού του δικαιώματος και να μην κατευθύνεται από το σύστημα.

Ο κύκλος των επιθυμητών ιδιοτήτων κλείνει με την ιδιότητα της μη – ταυτόχρονης καταγραφής- καταμέτρησης των ψήφων. Το IBB/PMA System διαχωρίζει τις φάσεις ρητά. Κατά τη διάρκεια που καταγράφεται η ψήφος στο σύστημα, δεν είναι δυνατή η καταμέτρηση της με κανένα μηχανισμό γιατί δεν είναι

σε μορφή που αναγνωρίζεται από το σύστημα. Από τη στιγμή που θα λήξει η διαδικασία της ψηφοφορίας και οι ψήφοι μετασχηματιστούν στην αρχική τους μορφή, τότε το σύστημα θα μπορεί να περάσει στην φάση της καταμέτρησης.

Σε αυτό το σημείο έχουμε τεκμηριώσει την ολοκλήρωση των ιδιοτήτων στο IBB/PMA System και αναγνωρίζουμε ότι αποτελεί ένα σύστημα το οποίο μπορεί να υποστηρίξει εκλογικές διαδικασίες. Η πλήρης ανάλυση των μηχανισμών που παρουσιάστηκαν γίνεται σε ακόλουθο τμήμα της εργασίας.

10. Ανάλυση Συστήματος

Το σύστημα IBB/PMA αποτελείται από πέντε διακριτά τμήματα, τα οποία είτε συνδέονται άμεσα το ένα με το άλλο, είτε έμμεσα, ανάλογα με τον σχεδιασμό του συστήματος και τις λειτουργίες που επιτελούν. Τα πέντε αυτά συστήματα είναι:

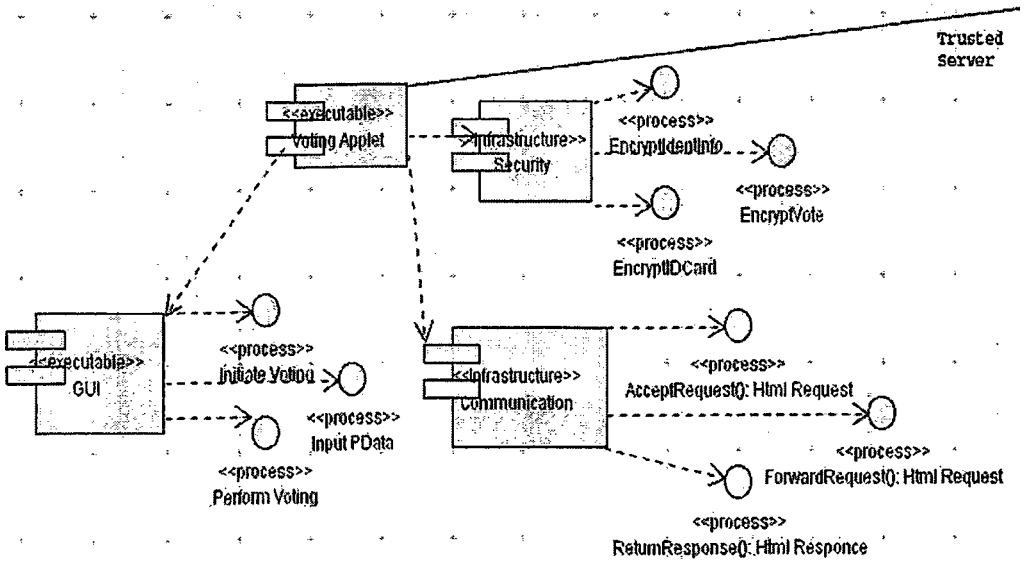
- Το πρόγραμμα πελάτη – IBB/PMA Voting Client
- Το πρόγραμμα Εξυπηρετητή πιστοποίησης – Authentication Server
- Το πρόγραμμα Έμπιστου Εξυπηρετητή – Trusted Server
- Το πρόγραμμα Καταμέτρησης ψήφων - Tallier
- Το σύνολο των πανομοιότυπων καλπών - IBBs

A. Πρόγραμμα πελάτη – IBB/PMA Voting Client

Το πρόγραμμα πελάτη είναι υπεύθυνο για την οποιαδήποτε επικοινωνία του ψηφοφόρου με το σύστημα ψηφοφορίας. Μέσω αυτού γίνεται η εισαγωγή των προσωπικών του στοιχείων για την εξακρίβωση της ταυτότητάς του, η ενημέρωση για την δυνατότητα ψήφου, η ψηφοφορία και η επιβεβαίωση για την λήψη της ψήφου από το σύστημα. Το πρόγραμμα έχει ενσωματωμένες δυνατότητες ασφαλείας, γραφικών και επικοινωνιών. Η ασφάλεια που υποστηρίζει είναι για την κρυπτογράφηση των προσωπικών στοιχείων και της ψήφου, καθώς επίσης και για την κρυπτογράφηση του πακέτου που θα αποσταλεί στο σύστημα.

Η δυνατότητα παραγωγής γραφικών επιτρέπει στο πρόγραμμα τη δημιουργία των απαραίτητων διεπαφών για την ολοκλήρωση της εκλογικής διαδικασίας από τον χρήστη. Τέλος οι δυνατότητες επικοινωνίας που έχουν ενσωματωθεί στο πρόγραμμα πελάτη είναι υπεύθυνες για την αποστολή και λήψη όλων των δεδομένων κατά τη διάρκεια της εκλογικής διαδικασίας.

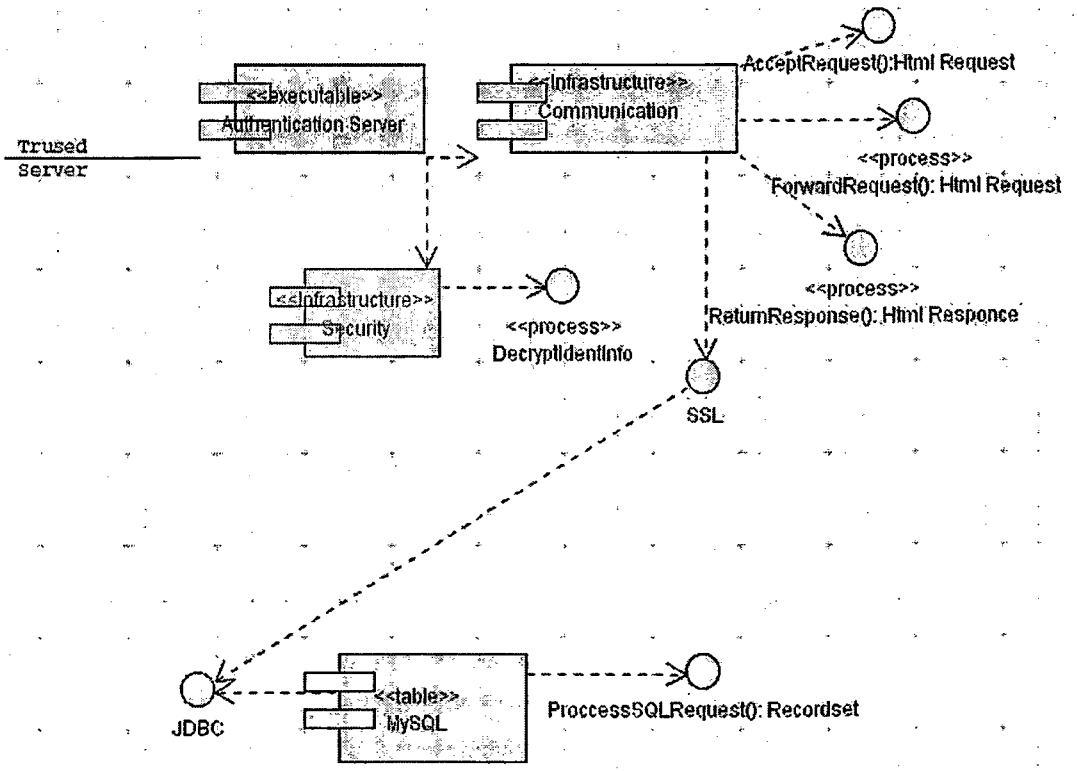
Στο σύστημα ηλεκτρονικής ψηφοφορίας IBB/PMA το πρόγραμμα πελάτη επικοινωνεί άμεσα μόνο με τον Έμπιστο Εξυπηρετητή – Trusted Server, για λόγους ασφαλείας.



Σχήμα 11.A.1: Πελάτης IBB/PMA

Β. Πρόγραμμα Εξυπηρετητή Πιστοποίησης – Authentication Server

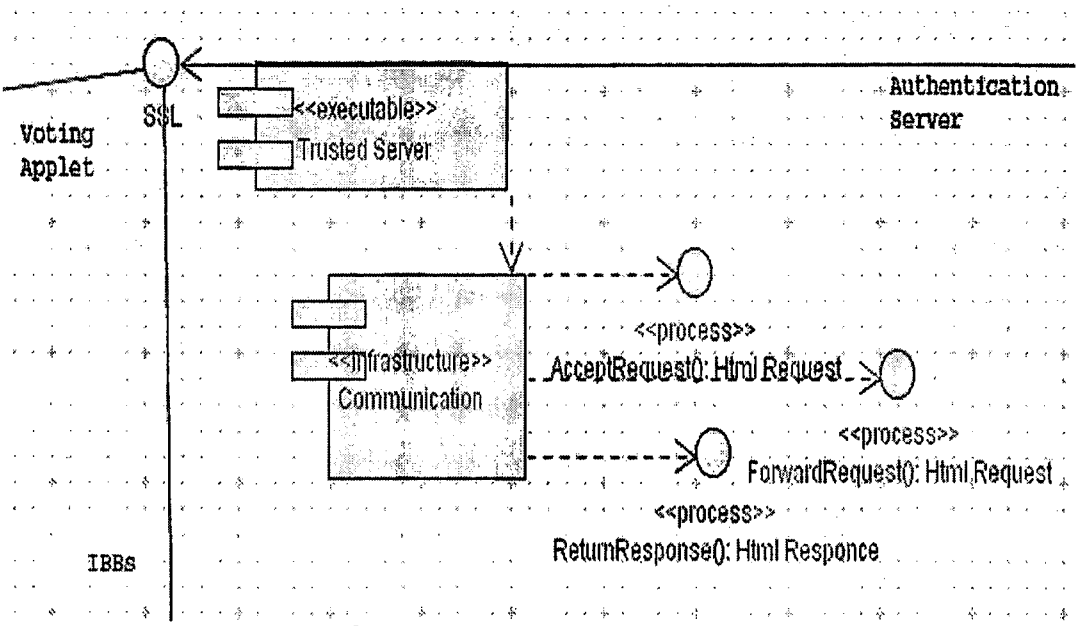
Ο Εξυπηρετητής πιστοποίησης είναι υπεύθυνος για την πιστοποίηση των χρηστών και την εξουσιοδότησή τους να ψηφίσουν. Συνδέεται άμεσα με τον Έμπιστο Εξυπηρετητή και το σύστημα που διαχειρίζεται τη βάση των νόμιμων ψηφοφόρων από την οποία και γίνεται η πιστοποίηση. Περιέχει και αυτός δυνατότητες ασφάλειας, επικοινωνιών και γραφικών. Επιπλέον περιέχει και τους απαραίτητους οδηγούς για την σύνδεση με την εκάστοτε βάση δεδομένων που διατηρεί τα στοιχεία των νόμιμων ψηφοφόρων. Το τμήμα της ασφάλειας είναι επιφορτισμένο με την αποκρυπτογράφηση των στοιχείων που έχει στείλει ο κάθε ψηφοφόρος, ώστε να τα μετατρέψει σε μορφή τέτοια που να του επιτρέπει την ανεύρεση του χρήστη στην Βάση νόμιμων ψηφοφόρων. Το τμήμα της επικοινωνίας είναι υπεύθυνο για την ολοκλήρωση των επικοινωνιών με τον Έμπιστο Εξυπηρετητή και την Βάση Δεδομένων που περιέχει τους νόμιμους ψηφοφόρους. Τέλος το σύστημα γραφικών του είναι περιορισμένο στις απαραίτητες λειτουργίες που πρέπει να πραγματοποιηθούν για την εγκατάσταση του συστήματος ηλεκτρονικής ψηφοφορίας και την εκτέλεση απλών εντολών προς την βάση των νόμιμων ψηφοφόρων.



Σχήμα 11.Β.1: Εξυπηρετητής πιστοποίησης

C. Το πρόγραμμα Έμπιστου Εξυπηρετητή – Trusted Server

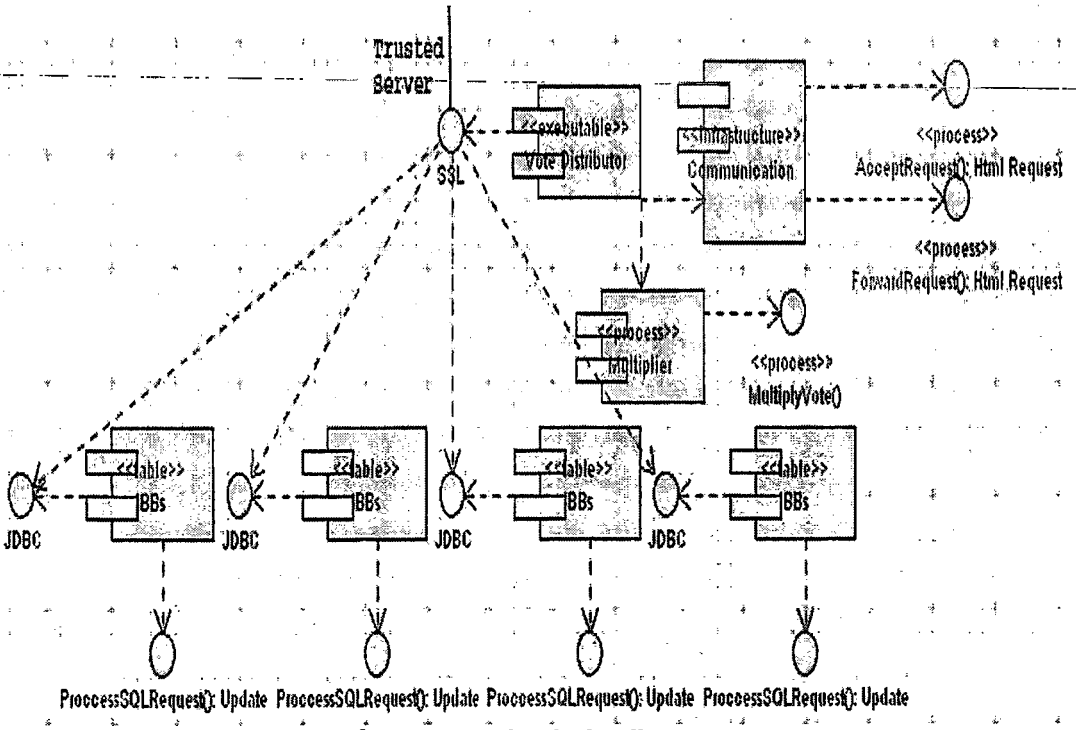
Ο Έμπιστος Εξυπηρετητής βρίσκεται ανάμεσα στον Εξυπηρετητή Πιστοποίησης και στο Πρόγραμμα Πελάτη, στο IBB/PMA σύστημα. Το σύστημα αυτό είναι ο ενδιάμεσος κρίκος που συνδέει τα δύο αυτά προγράμματα μεταξύ τους και με τις κάλπες ψηφοφορίας IBB. Με την ύπαρξη του Έμπιστου Εξυπηρετητή στο σχεδιασμό μας, επιτυγχάνεται η μη ταυτόχρονη ύπαρξη της ψήφου και των στοιχείων του ψηφοφόρου στο ίδιο σημείο μέσα στο σύστημα. Με αυτόν τον τρόπο ισχυροποιείται η ασφάλεια και η διαύγεια στο σύστημα, καθώς είναι από σχεδιασμού του αδύνατον να μπορέσει να συνδεθεί η ψήφος με τον ψηφοφόρο με τρόπο που θα μπορέσει να χρησιμοποιηθεί από κάποιον που θα προσπαθήσει να αποσπάσει δεδομένα. Το μόνο σημείο που βρίσκεται η ψήφος και τα στοιχεία του ψηφοφόρου ταυτόχρονα είναι στο πρόγραμμα πελάτη, κατά τη διάρκεια της ψηφοφορίας. Αυτό δεν αποτελεί πρόβλημα γιατί από την στιγμή που ο πελάτης θα τερματίσει την σύνδεσή του, όλα τα δεδομένα από το πρόγραμμα θα σβηστούν. Ο Έμπιστος Εξυπηρετητής συνδέεται άμεσα με το Πρόγραμμα Πελάτη, τον Εξυπηρετητή Πιστοποίησης και τις κάλπες IBBs. Στον Έμπιστο Εξυπηρετητή έχει ενσωματωθεί μόνο η δυνατότητα της επικοινωνίας, μιας και ο μοναδικός του ρόλος είναι η δρομολόγηση των εκάστοτε δεδομένων στην σωστή τοποθεσία.



Σχήμα 11.C.1:Έμπιστος Εξυπηρετητής

D. Οι Πανομοιότυπες Κάλπες – Identical Ballot Boxes

Οι Identical Ballot Boxes είναι υπεύθυνες για την αποθήκευση των ψήφων κατά τη διάρκεια της εκλογικής διαδικασίας. Οι κάλπες είναι τέσσερις, όσοι και οι υποψήφιοι που υποστηρίζονται από το σύστημα στην παρούσα έκδοσή. Η ιδιαιτερότητα του σχεδιασμού των IBBs έγκειται στο ότι κατά την εκλογική διαδικασία και οι τέσσερις ενημερώνονται με τα ίδια δεδομένα ταυτόχρονα. Το αποτέλεσμα αυτής της λειτουργίας είναι ότι στο τέλος της ψηφοφορίας, στο σύστημα θα βρίσκονται τέσσερις φορές όλες οι ψήφοι και θα μπορούν να συγκριθούν μεταξύ τους για την ανεύρεση πιθανής παρατυπίας από πλευράς των διαχειριστών τους. Οι IBBs συνδέονται με τον Έμπιστο Εξυπηρετητή, από όπου και παίρνουν τα δεδομένα.

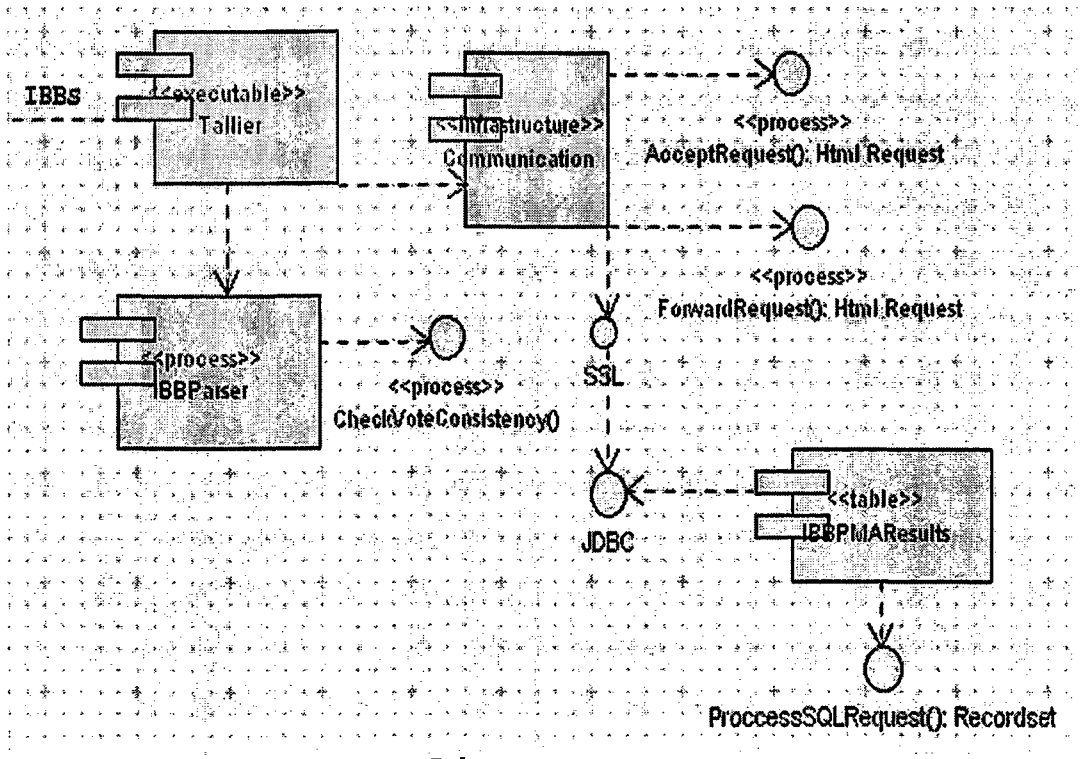


Σχήμα 11.D.1:Identical Ballot Boxes

Το πρόγραμμα Καταμέτρησης Ψήφων - Tallier

Ο Tallier είναι υπεύθυνος για την καταμέτρηση των ψήφων μετά από την λήξη της διαδικασίας ψηφοφορίας. Αφού έχει ολοκληρωθεί η ψηφοφορία και έχουν συγκριθεί οι Identical Ballot Boxes, ενεργοποιείται το πρόγραμμα καταμέτρησης. Αυτό αρχικά δημιουργεί την τελική IBB από την οποία θα εξαχθούν τα δεδομένα της ψηφοφορίας. Για να δημιουργηθεί αυτός ο πίνακας, το πρόγραμμα προσπελαύνει όλες σειριακά τις εγγραφές από τις IBBs και δημιουργεί την τελική IBB. Αφού έχει ολοκληρωθεί η διαδικασία, ο Tallier προσπελαύνει την IBB, και ανάλογα με τα περιεχόμενά της, ενημερώνει την IBBPMAResults Βάση Δεδομένων. Η IBB/PMAResults περιέχει τα ονόματα των υποψηφίων και αθροιστικά τις ψήφους τις οποίες πήραν. Με την ολοκλήρωση της εργασίας του Tallier γίνεται η δημοσίευση των αποτελεσμάτων.

Ο Tallier άμεσα συνδέεται με τις IBBs και την IBBPMAResults Βάση Δεδομένων.



Σχήμα 11.Ε.1:Tallier

11. Φάσεις ψηφοφορίας IBB/PMA System

Όπως αναφέρθηκε στο τμήμα των ιδιοτήτων του IBB/PMA System, το σύστημα ηλεκτρονικής ψηφοφορίας διαχωρίζει ρητά τις φάσεις της ψηφοφορίας. Οι φάσεις κατά κύριο λόγο ακολουθούν το πρότυπο φάσεων της κλασσικής ψηφοφορίας με τις απαραίτητες προσαρμογές για την ηλεκτρονική τους απόδοση. Οι φάσεις σε χρονική ακολουθία είναι: Αρχικοποίηση διαδικασίας ψηφοφορίας, Ψηφοφορία, Καταμέτρηση ψήφων, Δημοσίευση Αποτελεσμάτων και Επαλήθευση διαδικασίας ψηφοφορίας.

A. Αρχικοποίηση διαδικασίας ψηφοφορίας

Αυτή η φάση είναι χρονικά η πρώτη που διενεργείται όταν χρησιμοποιείται το σύστημα ηλεκτρονικής ψηφοφορίας και είναι ίσως η βασικότερη. Ο λόγος που καθιστά τόσο σημαντική την φάση αυτή είναι ότι, αποτελεί την φάση όπου γίνονται οι αρχικές ρυθμίσεις στο σύστημα και σύμφωνα με αυτές θα συνεχιστεί η ψηφοφορία. Η επιτυχής ολοκλήρωση αυτής της φάσης έγκειται στην ορθή εισαγωγή των δεδομένων και στις επιτυχείς εγκαταστάσεις των επιμέρους συστημάτων του IBB/PMA System. Αυτή η φάση έχει υποκατηγορίες διαδικασιών που μπορούν να γίνουν είτε ταυτόχρονα είτε με χρονική ακολουθία, ανάλογα με τη μορφή τους και τη θέση τους στο σύστημα.

Οι διαδικασίες αυτές είναι:

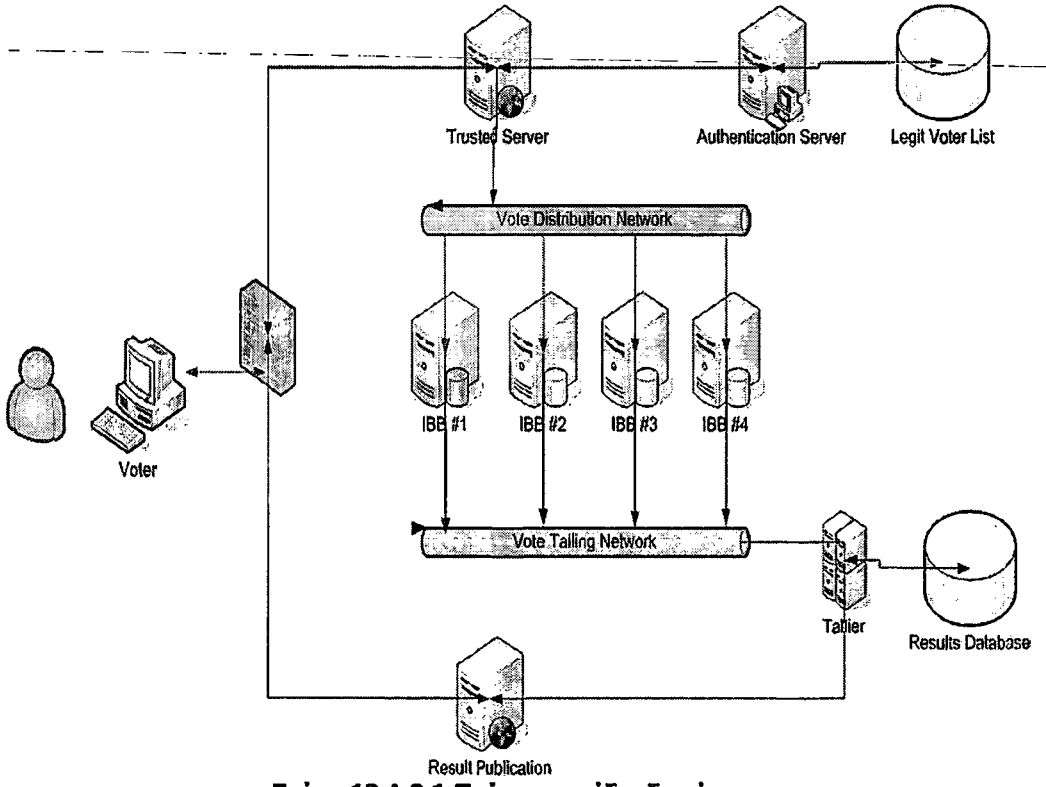
1. Διορισμός της εκλογικής αρχής (Voting Authority)
2. Εφαρμογή του προτύπου δικτύωσης των υπολογιστικών συστημάτων που θα φιλοξενήσουν τα λειτουργικά τμήματα του IBB/PMA System.
3. Αρχικοποίηση και απαραίτητες ρυθμίσεις στα υπολογιστικά συστήματα που θα φιλοξενήσουν τα λειτουργικά τμήματα του IBB/PMA System.
4. Εγκατάσταση των λειτουργικών τμημάτων του IBB/PMA System στα υπολογιστικά συστήματα που θα χρησιμοποιηθούν.
5. Δημιουργία της λίστας νόμιμων ψηφοφόρων – legit voter (Authorization_DB)
6. Δημιουργία της λίστας υποψηφίων (Candidate_DB)
7. Εισαγωγή της λίστας νόμιμων ψηφοφόρων και της λίστας υποψηφίων στο IBB/MPA System

1. Διορισμός της εκλογικής αρχής

Σε αυτήν την διαδικασία έχουμε το σχηματισμό της εκλογικής αρχής, η οποία είναι υπεύθυνη για το σύνολο της διαδικασίας ψηφοφορίας και έχει ορισμένες αρμοδιότητες και ευθύνες επί του συστήματος. Σε γενικές γραμμές είναι υπεύθυνη για την δημιουργία του κλειδιού κρυπτογράφησης που θα χρησιμοποιηθεί κατά τη διάρκεια της ψηφοφορίας, για την έναρξη και λήξη της ψηφοφορίας ως επίσης είναι και η επιτροπή που καλεί τους ψηφοφόρους που επιλέχθηκαν για την επαλήθευση του συστήματος.

2. Εφαρμογή του προτύπου δικτύωσης των υπολογιστικών συστημάτων που θα φιλοξενήσουν τα λειτουργικά τμήματα του IBB/PMA System

Το σύστημα για να μπορεί να λειτουργήσει θα πρέπει να είναι συνδεδεμένα τα επιμέρους συστήματά του σε ειδική τοπολογία με συγκεκριμένη μορφή. Εκτελείται λοιπόν η αρχικά η μελέτη του χώρου που θα φιλοξενήσει το σύστημα και ακολούθως πραγματοποιείται η χωροταξική κατανομή των μηχανημάτων ανάλογα με το πρότυπο σχέδιο δικτύωσης. Το πρότυπο σχέδιο δικτύωσης δημιουργήθηκε για να παρουσιάσει τον τρόπο που θα πρέπει να συνδέονται οι υπολογιστές που υποστηρίζουν την εκλογική διαδικασία στο δίκτυο. Ανάμεσα στο IBB/PMA σύστημα και στον ψηφοφόρο θα πρέπει να υπάρχει firewall το οποίο να μην επιτρέπει την είσοδο σε καμιά άλλη θύρα εκτός από τις ορισμένες για το σύστημα.



Σχήμα 12.A.2.1:Πρότυπο σχέδιο δικτύωσης

3. Αρχικοποίηση και απαραίτητες ρυθμίσεις στα υπολογιστικά συστήματα που θα φιλοξενήσουν τα λειτουργικά τμήματα του IBB/PMA System

Η διαδικασία αυτή περιλαμβάνει τις υπολειτουργίες που είναι απαραίτητο να πραγματοποιηθούν στα μηχανήματα που θα φιλοξενήσουν το σύστημα, προτού της εγκατάστασής του. Η επιτυχία της ψηφοφορίας εξαρτάται άμεσα από την ορθή διεκπεραίωση της συγκεκριμένης λειτουργίας. Είναι κατανοητό ότι εάν δεν πραγματοποιηθεί σωστά αυτή η λειτουργία, το σύστημα θα είναι ευάλωτο σε παράγοντες εκτός του συστήματος ηλεκτρονικής ψηφοφορίας και δεν θα είναι έμπιστο. Οι διαδικασίες που υπάρχουν σε αυτό το στάδιο είναι:

I. Έλεγχος του υλικού και της καλής λειτουργίας του, ως επίσης και της συμβατότητάς του με τις εκάστοτε υπολογιστικές συνθήκες. Το υλικό και η ορθή λειτουργία του είναι ένας βασικός παράγοντας που διασφαλίζει την επιτυχημένη εγκατάσταση του λογισμικού. Σε ένα ελαττωματικό υλικό συχνά παρουσιάζονται προβλήματα σε φαινομενικά απλές και συνηθισμένες λειτουργίες. Η ύπαρξη σωστού υλικού είναι καίρια για την μετάβαση στην ακόλουθη διεργασία.

II. Διαμόρφωση των υπολογιστικών συστημάτων που φιλοξενούν το IBB/PMA System. Από τη στιγμή που έχουν συναρμολογηθεί τα υπολογιστικά συστήματα που θα χρησιμοποιηθούν για το IBB/PMA System, περνάμε στην διαμόρφωσή χαμηλού επιπέδου (low-level format). Ο λόγος που προβαίνουμε σε αυτή τη διαδικασία είναι ότι επιθυμούμαι το σύστημα να μην έχει κανένα κατάλοιπο από πιθανή προηγούμενη χρήση του.

III. Εγκατάσταση των απαραίτητων προγραμμάτων τρίτων προμηθευτών. Με την ολοκλήρωση της προηγούμενης διαδικασίας, περνάμε στην εγκατάσταση των απαραίτητων προγραμμάτων τρίτων προμηθευτών (third party software). Το λογισμικό που εγκαθίσταται είναι:

i. Λειτουργικό Σύστημα [Microsoft Windows 2000 ή Microsoft Windows 2003 Cooperate Edition] για τα συστήματα που θα φιλοξενήσουν τα "Έμπιστος εξυπηρετητής – Trusted Server", "Εξυπηρετητής Πιστοποίησης – Authentication Server" και για τους "Εξυπηρετητές Κάλπης – IBB Servers"

ii. Antivirus Software για όλα τα συστήματα. Προτεινόμενο λογισμικό Antivirus για όλα τα μηχανήματα είναι το F-Prot Antivirus version 3.15

iii. Τα προγράμματα Microsoft Office στον Εξυπηρετητή Δημοσίευσης Αποτελεσμάτων- Publication Server

iv. Λογισμικό εποπτείας – Monitoring Software, τύπου key logger. Προτεινόμενο λογισμικό monitoring είναι το Perfect Keylogger version 1.5.3. Και εκτός αυτού το σύστημα ηλεκτρονικής ψηφοφορίας έχει ενσωματωμένο το logging σύστημα log4j.

IV. Ενημέρωση των προγραμμάτων τρίτων προμηθευτών. Τα προγράμματα που έχουν εγκατασταθεί στις προηγούμενες διεργασίες πρέπει να ελεγχθούν και να ενημερωθούν με τις τελευταίες ενημερώσεις λογισμικού που προσφέρει η αντίστοιχη εταιρία, για το κάθε ένα από αυτά. Τα λειτουργικά συστήματα θα πρέπει να έχουν τις πιο πρόσφατες ενημέρωσης ασφαλείας και τα αντιβιοτικά θα πρέπει να έχουν τις πιο πρόσφατες ενημερώσεις στις λίστες προστασίας τους.

V. Τέλος στην περίπτωση που επιθυμείται η χρήση κάποιου DBMS διαφορετικού από αυτό που ενσωματώνεται στο σύστημα ηλεκτρονικής ψηφοφορίας IBB/PMA θα πρέπει να γίνει και η εγκατάστασή του στον αντίστοιχο εξυπηρετητή.

4. Εγκατάσταση των λειτουργικών τμημάτων του IBB/PMA System στα υπολογιστικά συστήματα που θα χρησιμοποιηθούν

Έχοντας ολοκληρώσει τις εγκαταστάσεις των εξωτερικών, του συστήματος ηλεκτρονικής ψηφοφορίας, προγραμμάτων θα προβούμε στην εγκατάσταση πλέον του ίδιου του συστήματος. Τα υποσυστήματα του IBB/PMA System θα εγκατασταθούν στα αντίστοιχα υπολογιστικά συστήματα που θα τα φιλοξενήσουν. Το πρόγραμμα TrustedSrv θα εγκατασταθεί στον εξυπηρετητή που θα βρίσκεται στην θέση του Έμπιστου Εξυπηρετητή. Η λίστα με τους νόμιμους ψηφοφόρους τοποθετείται στον Εξυπηρετητή Πιστοποίησης – Authentication Server. Τέλος η λίστα που θα είναι η κάλπη IBB τοποθετείται σε κάθε έναν από τους εξυπηρετητές κάλπης – IBB Servers. Έχοντας ολοκληρώσει και αυτήν την διαδικασία, το σύστημα είναι έτοιμο να λειτουργήσει σαν σύστημα ηλεκτρονικής ψηφοφορίας. Οι υπολογιστές στους οποίους αντιστοιχούν οι λειτουργικές μονάδες του IBB/PMA System έχουν προ-αποφασιστεί κατά τη διαδικασία της εφαρμογής του προτύπου δικτύωσης.

5. Δημιουργία της λίστας νόμιμων ψηφοφόρων – legit voter (Authorization_DB)

Σε αυτή τη διαδικασία συντάσσεται η λίστα των νόμιμων ψηφοφόρων. Αυτή η λίστα θα χρησιμοποιηθεί στο σύστημα ηλεκτρονικής ψηφοφορίας για να παρέχει πιστοποίηση στους χρήστες που επιθυμούν να αποκτήσουν πρόσβαση στην ψηφοφορία που θα υποστηρίζει το IBB/PMA System. Η σύνταξή της είναι αρμοδιότητα της εκλογικής αρχής.

Η διαδικασία αυτή υποχρεωτικά έπεται της διαδικασίας του διορισμού της εκλογικής αρχής και προηγείται της διαδικασίας της εγκατάστασης των λειτουργικών τμημάτων στα υπολογιστικά συστήματα. Η λίστα αυτή εκτός από τις εγγραφές που περιέχουν τα προσωπικά στοιχεία του κάθε νόμιμου ψηφοφόρου περιέχει και ένα πεδίο το οποίο χρησιμοποιείται από το σύστημα για τον έλεγχο της ψηφοφορίας.

Κατά τη στιγμή της ψηφοφορίας, το πεδίο αυτό συμπληρώνεται δείχνοντας ότι ο χρήστης έχει ψηφίσει και απαγορεύοντας του την είσοδο κατά την υπόλοιπη ώρα στη διαδικασία της ψηφοφορίας.

Δομή της λίστας νόμιμων ψηφοφόρων – Legit Voter List

Field Name	Field Type	Description
VName	Varchar	Όνομα Ψηφοφόρου
VSurname	Varchar	Επώνυμο Ψηφοφόρου
VIDCode	Int	Αριθμός Δελτίου Ταυτότητας
VAddress	Varchar	Διεύθυνση Ψηφοφόρου
VEmployment	Varchar	Επάγγελμα Ψηφοφόρου
VAge	Float	Ηλικία Ψηφοφόρου
VLogin	Varchar	Όνομα χρήστη Ψηφοφόρου
VPwd	Varchar	Κωδικός χρήστη Ψηφοφόρου
Voted	Boolean	Κατάσταση Ψήφου False = Legit Voter True= Ex-legit Voter

6. Δημιουργία της λίστας υποψηφίων (Candidate_DB)

Ομοίως και με την λίστα των νόμιμων ψηφοφόρων, η σύνταξη της λίστας υποψηφίων αποτελεί ευθύνη της εκλεκτικής αρχής. Η λίστα αυτή θα χρησιμοποιηθεί από το σύστημα κατά την διαδικασία της καταμέτρησης των ψήφων, παρέχοντας τα στοιχεία των υποψηφίων προς εκλογή ανθρώπων. Τα στοιχεία από την λίστα αυτή θα αποτελέσουν είσοδο για την δημιουργία της λίστας αποτελεσμάτων της εκλογικής διαδικασίας.

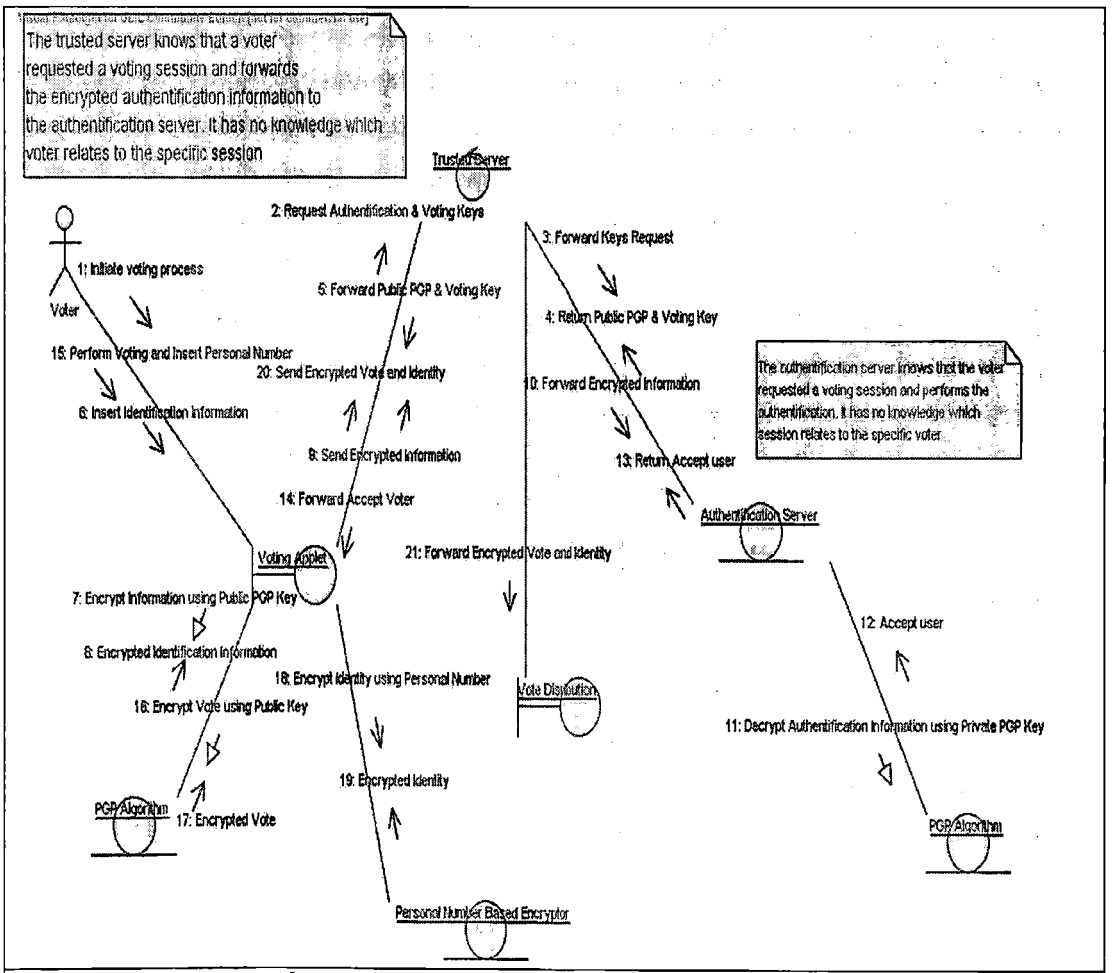
Δομή της λίστας υποψηφίων Candidate_DB

Field Name	Field Type	Description
CName	Varchar	Όνομα Υποψηφίου
CSurname	Varchar	Επώνυμο Υποψηφίου
CIDCode	Int	Αριθμός Δελτίου Ταυτότητας
CCity	Varchar	Πόλη Υποψηφίου
CAddress	Varchar	Διεύθυνση Υποψηφίου
CEmployment	Varchar	Επάγγελμα Υποψηφίου
CAge	Float	Ηλικία Υποψηφίου

Β. Ψηφοφορία

Ακολούθως της διαδικασίας της Αρχικοποίησης Ψηφοφορίας έχουμε την διαδικασία της Ψηφοφορίας. Το σημαντικότερο μέρος σε ένα σύστημα ηλεκτρονικής ψηφοφορίας είναι το μέρος της ψηφοφορίας. Ο σχεδιασμός του αποτελεί διαδικασία που πρέπει να γίνει πολύ προσεκτικά, ανεξαρτήτου της προσέγγισης που ακολουθείται στην υλοποίηση. Ο λόγος που απαιτείται αυτή η προσοχή είναι ότι σε ένα σύστημα με τέτοια πολυπλοκότητα και παράγοντες και η παραμικρή παράληψη μπορεί κάλλιστα να οδηγήσει σε αστοχία όλου του συστήματος.

Οι υλοποιήσεις που έχουν πραγματοποιηθεί μέχρι σήμερα, έχουν η κάθε μια, τη δική τους διαφορετική προσέγγιση στο θέμα του σχεδιασμού της φάσης της ψηφοφορίας. Στο IBB/PMA System στην φάση της ψηφοφορίας έχουμε την εμφάνιση της καινοτομίας που προσφέρει αυτή η προσέγγιση στα συστήματα ηλεκτρονικής ψηφοφορίας. Οι Identical Ballot Boxes, έχουν αρχικοποιηθεί στην προηγούμενη φάση και είναι έτοιμες να υποστηρίξουν το σύστημα, παρέχοντας αξιοπιστία στην αποθήκευση και διαφύλαξη των ψήφων, σε όλη τη διάρκεια της φάσης.



Σχήμα 12.Β.1 : Διαδικασία Ψηφοφορίας IBB/PMA System

Η φάση της ψηφοφορίας μπορεί να χωριστεί και να παρουσιαστεί σύμφωνα με τη χρονική αλληλουχία των διαδικασιών που επιτελούνται. Ο λόγος που πραγματοποιείται αυτό είναι ότι το θέμα της ηλεκτρονικής ψηφοφορίας είναι πολυδιάστατο και δανείζεται στοιχεία και εφαρμογές από όλους τους τομείς της υπολογιστικής γνώσεις. Κάθε άλλη προσέγγιση μπορεί να οδηγήσει σε σύγχυση και παρερμηνεύσεις που πιθανότατα θα καταλήξουν σε κάποιο ημιτελές ή ελαττωματικό έργο.

Η διαδικασία της ψηφοφορίας αρχίζει όταν ο χρήστης στέλνει ένα "Σήμα Έναρξης Ψηφοφορίας [1] " στην εφαρμογή ψηφοφορίας – Voting Applet που βρίσκεται πλέον τοπικά στο μηχάνημά του.

Η εφαρμογή αυτή με την σειρά της επικοινωνεί με τον ενδιάμεσο "Έμπιστο Εξυπηρετητή – Trusted Server" στέλνοντας του δύο αιτήσεις [2]. Η πρώτη είναι η αίτηση για Πιστοποίηση και η δεύτερη είναι μια αίτηση για την αποστολή των κλειδιών κρυπτογράφησης που θα χρησιμοποιηθούν στην σύνοδο ψηφοφορίας.

Ο Έμπιστος Εξυπηρετητής [3] με τη σειρά του προωθεί μόνο την μια από τις δύο αιτήσεις. Η αίτηση που προωθείται είναι η αίτηση για τα κλειδιά κρυπτογράφησης. Ο εξυπηρετητής ο οποίος λαμβάνει αυτήν την αίτηση είναι ο "Εξυπηρετητής Πιστοποίησης", ο οποίος του απαντάει στέλνοντας δύο κλειδιά [4]. Πρώτα στέλνει το Δημόσιό του κλειδί κρυπτογράφησης (PubPGP) και ακολούθως στέλνει το κλειδί που θα χρησιμοποιηθεί κατά τη σύνοδο της ψηφοφορίας.

Ο Έμπιστος Εξυπηρετητής προωθεί τα δύο κλειδιά στην Εφαρμογή Ψηφοφορίας του πολίτη - Voting Applet [5]. Σε αυτό το σημείο έχει ολοκληρωθεί η ανταλλαγή των απαραίτητων κλειδιών για την συνεδρία και ο ψηφοφόρος δέχεται ένα μήνυμα [6]. Το μήνυμα αυτό τον προτρέπει να εισάγει τα στοιχεία του για την εξακρίβωση της ταυτότητάς του και την πιστοποίησή του ως έγκυρο ψηφοφόρο. Η εισαγωγή των στοιχείων γίνεται μέσω μιας φόρμας εισαγωγής στοιχείων που δημιουργείται αυτόματα από το Voting Applet.

Έτσι ολοκληρώνεται η λήψη των στοιχείων του ψηφοφόρου και μεταβαίνουμε στην επικύρωσή του. Η Voting Applet, επικοινωνεί με τον αλγόριθμο κρυπτογράφησης [7], που έχει ενσωματωμένο και κρυπτογραφεί τα στοιχεία του υποψηφίου με το δημόσιο κλειδί του Εξυπηρετητή Πιστοποίησης.

Αφού έχουν κρυπτογραφηθεί τα στοιχεία και έχουν επιστραφεί την εφαρμογή [8], η Voting Applet τα στέλνει στον Trusted Server [9], ο οποίος και τα προωθεί στον Authentication Server, για επεξεργασία και πιστοποίηση. Επειδή ο Trusted Server λαμβάνει τα στοιχεία σε κωδικοποιημένη μορφή γνωρίζει μόνο ότι κάποιος ψηφοφόρος ζήτησε να ψηφίσει, αλλά δεν γνωρίζει ποιος ψηφοφόρος συνδέεται με τη συγκεκριμένη συνεδρία. Η προώθηση των κρυπτογραφημένων πληροφοριών πιστοποίησης γίνεται στο στάδιο 10.

Μόλις λάβει ο Authentication Server τα κρυπτογραφημένα δεδομένα από τον Trusted Server, τα προωθεί στον αποκωδικοποιητή του [11]. Ο αποκωδικοποιητής χρησιμοποιεί το ιδιωτικό του κλειδί κρυπτογράφησης και αποκρυπτογραφεί τα δεδομένα. Η αποκρυπτογράφηση αυτή θα δώσει τα στοιχεία του χρήστη που επιθυμεί να ψηφίσει. Τότε ο Authentication Server, ελέγχει την λίστα νόμιμων ψηφοφόρων που έχει και αντίστοιχα επιστρέφει κάποιο αποτέλεσμα. Θα μελετηθεί η διαδικασία, για επιστρεφόμενο αποτέλεσμα "Accept User", γιατί σε διαφορετική περίπτωση , απλώς επιστρέφει σήμα τερματισμού της συνεδρίας, οπότε και τερματίζεται η σύνδεση από την Voting Applet. Αφού έχει λάβει ο Authentication Server σήμα "Accept User" [12], το στέλνει με τη σειρά του στον Trusted Server [13].

Με την ολοκλήρωση αυτής της διαδικασίας, έχομε διασφαλίσει ότι ο Authentication Server γνωρίζει ότι ο ψηφοφόρος ζήτησε μια συνεδρία ψηφοφορίας, και πραγματοποίησε την πιστοποίηση της ταυτότητάς του. Πέρα από αυτό όμως δεν γνωρίζει ποια συνεδρία σχετίζεται με τον συγκεκριμένο ψηφοφόρο.

Λαμβάνοντας ο Trusted Server το σήμα "Accept User", το στέλνει την Voting Applet [14]. Πλέον ο χρήστης είναι νόμιμος ψηφοφόρος και του εμφανίζεται ένα μήνυμα που τον προτρέπει να συνεχίσει την διαδικασία ψηφίζοντας. Η Voting Applet τότε δημιουργεί την φόρμα, που θα χρησιμοποιηθεί από τον ψηφοφόρο για να ολοκληρώσει την ψηφοφορία του.

Η φόρμα αυτή αποτελείται από δύο ξεχωριστά πεδία. Στο πρώτο πεδίο ο ψηφοφόρος καλείται να εισάγει την ψήφο προτίμησής του. Το δεύτερο πεδίο δέχεται έναν αριθμό από τον υποψήφιο. Αυτός ο αριθμός θα χρησιμοποιηθεί μετέπειτα στην διαδικασία κρυπτογράφησης της ψήφου. Έχοντας ολοκληρώσει την διαδικασία της κατάθεσης ψήφου, η Voting Applet έχει δεχθεί δύο εισόδους, την Ψήφο και τον Προσωπικό Αριθμό Ψηφοφόρου - VtrPNumber [15].

Η Voting Applet τότε εκτελεί δύο λειτουργίες. Αρχικά επικοινωνεί με τον ενσωματωμένο αλγόριθμο κρυπτογράφησης και κρυπτογραφεί την Ψήφο με το δημόσιο κλειδί του Authentication Server και μετά κρυπτογραφεί το παραγόμενο αποτέλεσμα με το κλειδί της εκλογικής αρχής [16]. Αφού λάβει την κρυπτογραφημένη ψήφο [17], επικοινωνεί με τον αλγόριθμο κρυπτογράφησης προσωπικού αριθμού και κρυπτογραφεί τον αριθμό ταυτότητας του ψηφοφόρου με κλειδί το κλειδί που εισάγει αυτός κατά την ψηφοφορία του – VtrPNumber [18]. Αφού λάβει τον κρυπτογραφημένο αριθμό ταυτότητας [19], τον ενώνει με την κρυπτογραφημένη ψήφο.

Το πακέτο που δημιουργείται τότε αποστέλλεται στον Trusted Server [20]. Αυτός μόλις λάβει το πακέτο το προωθεί στον Διανομέα Ψήφων –Vote Distributor [21]. Ο Vote Distributor, τότε πολλαπλασιάζει το πακέτο, δημιουργώντας τρία ακόμα πανομοιότυπα κομμάτια κώδικα με τον αρχικό [22]. Μετά την δημιουργία των τριών νέων πακέτων, ο Vote Distributor, επικοινωνεί με τους εξυπηρετητές που έχουν τις κάλπες IBBs και τοποθετεί το πακέτο μέσα τους [23]. Σε αυτή τη φάση έχει ολοκληρωθεί η διαδικασία της ψηφοφορίας.

C. Καταμέτρηση Ψήφων

Με τον επίσημο τερματισμό της διαδικασίας ψηφοφορίας ξεκινάει η διαδικασία της καταμέτρησης. Πλέον οι πολίτες δεν έχουν το δικαίωμα να ψηφίσουν, καθώς το σύστημα πλέον δεν δέχεται εισερχόμενες αιτήσεις σύνδεσης. Η διαδικασία που ακολουθείται στην φάση της καταμέτρησης στο σύστημά μας είναι σειριακή. Από την έναρξη μέχρι την λήξη της διαδικασίας της καταμέτρησης εκτελείται μια υπο-διεργασία την φορά. Πλέον οι IBBs έχουν λάβει την τελευταία τους μορφή σαν κάλπες και δεν θα έχουν τροποποιήσεις, εφόσον δεν θα καταθέτονται άλλες ψήφοι.

Οι κάλπες αυτές θα πρέπει να είναι λοιπόν πανομοιότυπες. Αρχικά γίνεται μια σύγκριση στις κάλπες. Κάθε πεδίο σειριακά ελέγχεται με το αντίστοιχο πεδίο από τις αδερφές κάλπες του. Εφόσον ο έλεγχος επιστρέψει ότι τα δεδομένα στα πεδία είναι ίδια, τότε προχωράει στη σύγκριση του επόμενου πεδίου. Στην περίπτωση που παρουσιαστεί κάποια παρατυπία, γίνονται δύο διαδικασίες για την επαναφορά της αρχικής ψήφου. Πρώτα ελέγχονται τα εργαλεία εποπτείας συστήματος (Keyloggers), που υπήρχαν στο μηχάνημα που παρουσιάστηκε η ανωμαλία. Αυτή η διαδικασία θα επιστρέψει την οποιαδήποτε επέμβαση του διαχειριστή της αντίστοιχης κάλπης, στην κάλπη αυτή, καθιστώντας εφικτό τον προσδιορισμό του σημείου όπου παρουσιάστηκε η ανωμαλία. Η δεύτερη ενέργεια που πραγματοποιείται είναι η σύγκριση των υπόλοιπων εγγραφών από τις κάλπες στο ίδιο πεδίο. Στην περίπτωση που η δεδομένη κάλπη έχει αλλοιωθεί, οι υπόλοιπες από τις IBB θα έχουν εγγραφές που θα συμφωνούν μεταξύ τους. Αυτές οι εγγραφές συγκρίνονται με τις αλλαγές που παρατηρήθηκαν στην αρχική κάλπη και η ψήφος επανέρχεται στο αρχικό της. Με αυτόν τον τρόπο ελέγχονται όλες οι ψήφοι για την ορθότητά τους. Αφού ολοκληρωθεί η διαδικασία της σύγκρισης των IBB και ελεγχθεί η εγκυρότητα των ψήφων μέχρι σε αυτό το στάδιο της καταμέτρησης, οι IBB ενώνονται και δημιουργείται η βάση καταμέτρησης. Αυτή η βάση θα χρησιμοποιηθεί για την καταμέτρηση των ψήφων και την έκδοση του αποτελέσματος. Σε αυτό το στάδιο έχουμε μια, πλέον, βάση με τις ψήφους και τους Αριθμούς Αστυνομικής Ταυτότητας των ψηφοφόρων σε κρυπτογραφημένη μορφή. Η βάση λοιπόν πρέπει να έρθει σε μια μορφή που θα είναι κατανοητή από το σύστημα προτού να μπορεί να δώσει ποσοτικά αποτελέσματα.

Για την μετατροπή της βάσης αυτής απαιτείται αρχικά το κλειδί της Εκλογικής Αρχής και εν συνεχεία το Ιδιωτικό Κλειδί του Εξυπηρετητή Πιστοποίησης.

Έτσι λοιπόν με την εισαγωγή του κλειδιού κρυπτογράφησης της Εκλογικής Αρχής, ο αλγόριθμος κρυπτογράφησης που χρησιμοποιήθηκε στις δεδομένες εκλογές, επαναφέρει τις ψήφους ένα επίπεδο κρυπτογράφησης πάνω. Τώρα πια οι ψήφοι στα πεδία μπορούν να αποκρυπτογραφηθούν, με αποτέλεσμα κάποια μορφή κατανοητή από το σύστημα , με χρήση του αλγορίθμου κρυπτογράφησης και του Ιδιωτικού κλειδιού του Εξυπηρετητή Πιστοποίησης.

Ακολούθως έρχεται η προαναφερθείσα διαδικασία. Οι ψήφοι αποκρυπτογραφούνται και πλέον η βάση στο πεδίο της ψήφου, περιέχει εγγραφές σε μορφή που μπορούν να γίνουν κατανοητές από το σύστημα. Αυτή θα είναι και η τελική μορφή της κάλπης την οποία θα χρησιμοποιήσει το σύστημα για να καταμετρήσει τις ψήφους των εκλογών και να μπορεί να αποφανθεί για το αποτέλεσμα. Σε αυτό το σημείο έχουμε λοιπόν μια βάση – κάλπη η οποία έχει ένα πεδίο με εγγραφές τις ψήφους των πολιτών και ένα δεύτερο πεδίο με εγγραφές τους αριθμούς των αστυνομικών τους δελτίων. Οι ψήφοι είναι σε μορφή κατανοητή από το σύστημα, αλλά οι αριθμοί του δελτίου ταυτότητας είναι ακόμα κρυπτογραφημένοι.

Έτσι και σε αυτή τη φάση, το απόρρητο της ψηφοφορίας του κάθε πολίτη είναι ακόμα διασφαλισμένο. Όπως έχουμε πει ακόμα και τώρα ο πολίτης, και μόνο αυτός μπορεί να ξεκλειδώσει την ένωση της ψήφου με την ταυτότητά του.

Έχοντας λοιπόν την τελική βάση, από την οποία θα εξαχθούν τα αποτελέσματα της εκλογικής διαδικασία, περνάμε στην κυριολεκτική καταμέτρηση των ψήφων. Ένα πρόγραμμα περνάει σειριακά τις εγγραφές και αντίστοιχα ενημερώνει μια διαφορετική βάση με τα αποτελέσματα (Result_DB). Στο τέλος της διεργασίας αυτής έχουμε μια βάση η οποία περιέχει τον αριθμό από τις ψήφους του κάθε υποψηφίου, όπως προκύπτουν από την κάλπη ψηφοφορίας. Η διαδικασία έχει ολοκληρωθεί και μπορούμε να προχωρήσουμε στη δημοσίευση των αποτελεσμάτων.

D. Δημοσίευση Αποτελεσμάτων

Στο σύστημα ηλεκτρονικής ψηφοφορίας IBB/PMA System με την ολοκλήρωση της διαδικασίας της καταμέτρησης των ψήφων, ξεκινάει η διαδικασία της δημοσίευσης των αποτελεσμάτων. Σε αυτή τη διαδικασία έχουμε την πρώτη αναγνώσιμη έξοδο του συστήματος, προς τους πολίτες. Το σύστημα παράγει μια σελίδα, μέσα από την οποία, οι ψηφοφόροι και υποψήφιοι ενημερώνονται για τα αποτελέσματα της ηλεκτρονικής ψηφοφορίας. Η δημιουργία της ιστοσελίδας αυτής γίνεται με χρήση της βάσης που φιλοξενεί τα αποτελέσματα της ψηφοφορίας και παράγεται κατά την προηγούμενη διαδικασία στο σύστημα.

Αρχικά στην φάση της δημοσίευσης των αποτελεσμάτων, η Βάση που φιλοξενεί τα αποτελέσματα της ψηφοφορίας, ελέγχεται από το σύστημα ως προς την μορφή των δεδομένων που περιέχει. Ακολούθως, το πρόγραμμα περνάει τις εγγραφές και παίρνει τα αποτελέσματα και τα στοιχεία του κάθε υποψηφίου. Τα δεδομένα αυτά αποτελούν είσοδο για τη δημιουργία της αντίστοιχης ιστοσελίδας. Η παρουσίαση των στοιχείων – αποτελεσμάτων γίνεται αρχικά με αριθμητικό και ακολούθως με γραφικό τρόπο. Η σελίδα αυτή είναι ελεύθερα προσβάσιμη από το κοινό και φιλοξενεί τα αποτελέσματα της εκλογικής διαδικασίας.

Ε. Επαλήθευση διαδικασίας ψηφοφορίας

Έχοντας περάσει στο σύνολό της σχεδόν την διαδικασία της ψηφοφορίας στο IBB/PMA System, θα εξετάσουμε την τελευταία φάση και τον τρόπο εκτέλεσής της μέσα στο σύστημα. Η τελευταία φάση είναι η φάσης της επαλήθευσης των ψήφων. Κάποια συστήματα ηλεκτρονικής ψηφοφορίας, έχουν σαν μέτρο ασφάλειας, τη διαδικασία του σφραγίσματος της ψήφου τους με μια blind signature. Αυτά τα συστήματα παρέχουν τη δυνατότητα στον κάθε ψηφοφόρο να επαληθεύει την ψήφο του μετά την ψηφοφορία. Το IBB/PMA System υλοποιεί κάποια παρόμοια διαδικασία, οριοθετώντας την με κάποιους περιορισμούς. Οι περιορισμοί επικεντρώνονται στον τρόπο, τη διαδικασία και το πλήθος των ατόμων που μπορούν να επαληθεύσουν τη ψήφο τους. Οι χρήση αυτών των περιορισμών κρίθηκε απαραίτητη, επειδή σε ένα τόσο σημαντικό θέμα όπως την ηλεκτρονική ψηφοφορία, όπου υπάρχει πάντα κάποιος κερδισμένος και κάποιος χαμένος, είναι πιθανό οι οπαδοί του υποψηφίου που έχασε να επιθυμούν όλοι να ελέγξουν την ψήφο τους, κατηγορώντας την εκλογική διαδικασία για νοθεία. Κάτι τέτοιο είναι κατανοητό ότι δεν μπορεί να είναι επιτρεπτό. Έτσι στο σύστημα IBB/PMA System η διαδικασία της επαλήθευσης της ψηφοφορίας, ενώ παρέχει το αντίστοιχο επίπεδο εμπιστοσύνης με τα προηγούμενα συστήματα, δεν επιτρέπει την ανεξέλεγκτη επαλήθευση των ψήφων. Το σύστημα στην φάση της δημοσίευσης των ψήφων, παραδίδει και τα στοιχεία τριών ψηφοφόρων, από την βάση νόμιμων ψηφοφόρων του. Οι ψηφοφόροι επιλέγονται στην τύχη, ανάμεσα σε όλους τους νόμιμους ψηφοφόρους, που όμως έχουν ασκήσει το εκλογικό τους δικαίωμα.

Η Εκλογική Αρχή με την σειρά της παίρνει αυτά τα ονόματα και καλεί τους ψηφοφόρους για την διαδικασία της επαλήθευσης. Οι ψηφοφόροι Παρουσιάζονται στην Εκλογική Αρχή και οδηγούνται σε κάποιο ειδικό τερματικό, το οποίο χρησιμοποιείται μόνο για την επαλήθευση των ψήφων. Σε αυτό το τερματικό , ο χρήστης καλείται να εισάγει τον προσωπικό του τετραψήφιο αριθμό. Αυτός είναι ο ίδιος αριθμός που εισάγει ο ψηφοφόρος μαζί με την ψήφο του. Όπως αναφέρθηκε στην διαδικασία της ψηφοφορίας αυτός ο αριθμός χρησιμοποιήθηκε από την εφαρμογή ψηφοφορίας ως κλειδί κρυπτογράφησης του δελτίου αστυνομικής ταυτότητάς του. Επιπλέον μαζί με τον Προσωπικό Τετραψήφιο αριθμό εισάγει και τον αριθμό του δελτίου της αστυνομικής του ταυτότητας. Το σύστημα περνάει όλες τις εγγραφές της κάλπης, στο πεδίο της Κρυπτογραφημένης αστυνομικής ταυτότητας και αρχικά τις αποκρυπτογραφεί και μετά τις συγκρίνει με τον αριθμό που έδωσε ο χρήστης. Όταν αποκρυπτογραφηθεί η σωστή ταυτότητα, τότε ανασύρεται και η ψήφος με την οποία ήταν δεμένη. Τελικά παρουσιάζεται η ψήφος στον ψηφοφόρο και ο χρήστης την επαληθεύει. Οι χρήστες έρχονται στο ειδικό τερματικό με αλφαβητική σειρά βάση των επωνύμων τους. Είναι σημαντικό επίσης να τονίσουμε ότι για λόγους ασφαλείας, δεν επιτρέπεται η λήψη της ψήφου αυτών των ψηφοφόρων με κανένα τρόπο, έτσι ώστε να μην μπορούν να αποδείξουν ότι ψήφισαν με κάποιον συγκεκριμένο τρόπο, σε άτομο εκτός του συστήματος.

12. Απαιτούμενο HW για το IBB/PMA Voting System

Το σύστημα ηλεκτρονικής ψηφοφορίας IBB/PMA θα πρέπει να εγκατάσταθεί σε μηχανήματα σύμφωνα με το πρότυπο δικτύωσης. Τα μηχανήματα στα οποία θα εγκατασταθούν ο Εξυπηρετητής Πιστοποίησης, ο Έμπιστος Εξυπηρετητής και οι Πανομοιότυπες Κάλπες θα πρέπει να είναι πάντοτε στην αιχμή της υπολογιστικής ισχύος. Τελευταίας τεχνολογίας συστήματα απαιτούνται για να μπορούν να υποστηρίξουν τον φόρτο που θα δημιουργείται από τις πολλές αιτήσεις σύνδεσης και από τη διαχείριση των ψηνδέσεων ψηφοφορίας. Από την πλευρά του πελάτη, το μηχάνημα που χρησιμοποιείται μπορεί να είναι υποδεέστερης υπολογιστικής ισχύος. Τα προτυνόμενα υπολογιστικά συστήματα είναι:

Για τους Εξυπηρετητές:

Processors Up to two Intel® Xeon™ processors with Intel Extended Memory 64 Technology up to 3.6GHz, Front side bus 800MHz, Cache 1MB L2, Chipset Intel E7520
Memory 256MB/12GB DDR-2 400 SDRAM; 16GB with availability of dual rank 4GB DIMMs1
Drive bays Six 1" Ultra320 hot-plug SCSI drives or five drive bays and one tape drive bay
Maximum internal storage Up to 1.8TB with 300GB HDD, Internal storage performance 10K/15K RPM SCSI drives
Network interface card Dual embedded Intel Gigabit4 NICs; single and dual port Intel PRO/1000
Power supply 700W, optional hot-plug redundant power
Availability ECC memory, Single Device Data Correction (SDDC), Spare Bank, Memory Mirroring; hot-plug SCSI hard drives; optional hot-plug redundant power; hot-plug redundant cooling; tool-less chassis; high availability fibre channel and SCSI cluster support; optional ROMB with battery-backed cache; optional Split Backplane; optional PERC RAID controller
Video Embedded ATI Radeon 7000-M with 16MB SDRAM

Για τους Πελάτες:

Τύπος HW	Μοντέλο
CPU	<300 MHz
RAM	<64 MB
Modem	Yes

13. Deploying του IBB/PMA Voting System

Α. Ρυθμίσεις Προ Εγκατάστασης

i. Αρχείο Επιλογών XML

Προτού να είναι δυνατή η εκτέλεση του IBB/PMA Voting System χρειάζεται να δημιουργηθεί ή να αλλαχτεί το αρχείο επιλογών. Οι επιλογές είναι γραμμένες σε XML και σκοπός τους είναι να χρησιμοποιούνται από τον installer κατά την παραμετροποίηση του IBB/PMA για κάθε ηλεκτρονική ψηφοφορία. Στην δοκιμαστική εγκατάσταση του IBB/PMA χρησιμοποιείται το IBB_PMA_sample.xml .

Τα σημαντικότερα πεδία που χρησιμοποιούνται στην παρούσα έκδοση είναι:

- **Ibbpma:vote_system**

Με έγκυρες επιλογές :

fptp	First Past the Post (Πλειοψηφικά)
borda	Σύστημα που βγάζει τον νικητή ανάλογα με τον βαθμό της κάθε ψήφου.

- **ibbpma:choice**

code	Αυτή η επιλογή είναι η μορφή που στέλνονται τα μηνύματα στον Authentication Server, αντίστοιχα για κάθε υποψήφιο. Πρέπει να είναι διαφορετικό για κάθε υποψήφιο.

Όταν θα δημιουργηθεί το αρχείο επιλογών για την εκλογική διαδικασία, το σύστημα είναι έτοιμο για να εγκατασταθεί και να ξεκινήσει η διαδικασία. Η εγκατάσταση γίνεται ανάλογα με τα μηχανήματα που θα χρησιμοποιηθούν για την εκάστοτε ψηφοφορία.
Στην περίπτωση επιθυμητής εγκατάστασης σε συστήματα με λειτουργικό Win32 η εγκατάσταση πραγματοποιείται μέσω της γραμμής εντολών με την εντολή:

java -cp .;FreeInstall.jar;log4j.jar;freeawt.jar;cryptix32.jar;xerces.jar
FreeInstall.Install

ii. Build του IBB/PMA λογισμικού

Με την χρήση του προγράμματος εγκατάστασης IBBPMAInstall, ένα υψηλής ασφαλείας αρχείο στον κατάλογο *UTIL* θα αλλάξει. Αυτό το αρχείο κρατάει σημαντικές πληροφορίες ασφαλείας, οι οποίες είναι μοναδικές σε κάθε υλοποίηση. Αποθηκεύονται μαζί με τα άλλα κοινά αρχεία του IBB/PMA στο αρχείο ibbpmautil.jar. Για ασφαλή διεξαγωγή των εκλογικών διαδικασιών θα πρέπει να δημιουργείται κάθε φορά καινούργιο αντίγραφο του IBBPMAutil.jar. Ο τρόπος για την δημιουργία είναι:

1) Μετονομασία του υπάρχοντος IBBPMAutil.jar σε IBBPMAutil.old.jar

2) Μετακίνηση από τον κατάλογο IBBPMAv1 στον κατάλογο IBBPMAv1/IBBPMA/util

3) Compile των αρχείων σε ένα Win32 σύστημα γίνεται με την εντολή

*javac -classpath .;cryptix32.jar *.java*

4) Δημιουργία ενός jar αρχείου στο Win32 σύστημα γίνεται με την εντολή

jar -cf IBBPMAutil.jar IBBPMA\util.class*

5) Τέλος αντιγράφετε το καινούργιο IBBPMAutil.jar που βρίσκεται στο IBBPMAv1/IBBPMA/util στον κατάλογο IBB/PMAv1

Αφού δημιουργηθεί το καινούργιο αντίγραφο του IBBPMAutil.jar πρέπει να γίνουν compile όλα τα κύρια προγράμματα του IBB/PMA συστήματος, τα οποία βασίζονται στο IBBPMAutil.jar που κατασκευάσαμε. Για να γίνει το compile των προγραμμάτων πρέπει να διασφαλίσουμε ότι έχουμε ένα classpath που να περιλαμβάνει το IBBPMAv1 κατάλογο, καθώς και τα αρχεία που περιλαμβάνονται το IBB/PMA σύστημα.

IBBPMAutil.jar	Το νέο αντίγραφο που δημιουργήσαμε
IBBPMAawt.jar	Υποστήριξη διεπαφών του IBB/PMA συστήματος
IBBPMAdbpool.jar	Αρχεία για το cache και την βελτιστοποίηση των Βάσεων Δεδομένων
Log4j.jar	Το Log4j σύστημα monitoring
Cryptix32.jar	Αλγόριθμοι κρυπτογράφησης και ασφάλειας
Xerces.jar	XML parser
Hsql.jar	Hypersonic SQL

Για να γίνει το compile θα πρέπει να βρίσκεστε στο parent κατάλογο του IBB/PMA συστήματος. Για συστήματα win32 η εντολή για το compile θα είναι :

javac –
classpathIBBPMAutil.jar;IBBPMAawt.jar;IBBPMAdbpool.jar;hsql.jar;log4j.jar;cryptix32.jar;.IBBPMA/.java*

Στην περίπτωση που θα παρατηρηθεί κάποιο πρόβλημα σχετικά με τα αρχεία IBBPMA.DBPool τότε μετακινηθείτε στο DBPool κατάλογο και δοκιμάστε ξανά αλλά με την εντολή

javac -classpath
IBBPMAutil.jar;IBBPMAawt.jar;IBBPMAdbpool.jar;hsql.jar;log4j.jar;cryptix32.jar;.
IBBPMAServer/*.java

Ξανά για τον PollManager:

javac -classpath IBBPMAutil.jar;IBBPMAawt.jar;crytix32.jar;.
PollManager/*.java

Και τέλος μια ακόμα φορά για τον IBBPMAClient:

javac -classpath IBBPMAutil.jar;IBBPMAawt.jar;cryptix32.jar;.
IBBPMAClient/*.java

Πλέον το IBB/PMA σύστημα ηλεκτρονικής ψηφοφορίας είναι έτοιμο για εγκατάσταση.

Β. Σημειώσεις κατά την εγκατάσταση

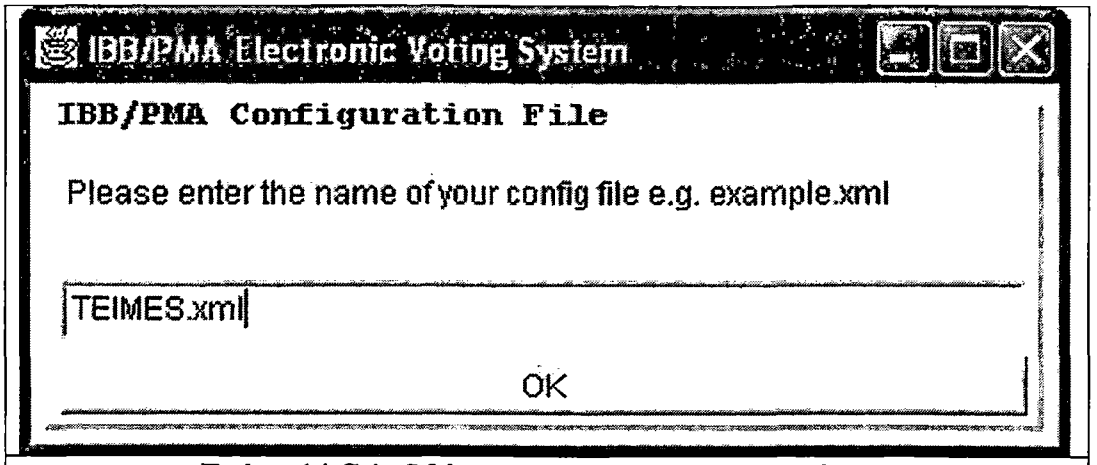
Κατά την εισαγωγή των οποιοδήποτε κωδικών συνίσταται η αποφυγή χρήσης σε απλά Αγγλικά. Οι κωδικοί είναι επιθυμητό να είναι συνδιασμός γραμμάτων, αριθμών και λοιπών χαρακτήρων. Στο IBB/PMA System μπορούν να χρησιμοποιηθούν και οι ακόλουθοι χαρακτήρες στους κωδικούς :

*; ( ) * = - _ + @ , . < > ?*

C. Εγκατάσταση

Κατά την εγκατάσταση θα ζητηθεί το αρχείο ρυθμίσεων της ψηφοφορίας. Για να επιλεγεί θα πρέπει να δοθεί η πλήρης διαδρομή του αρχείου και το όνομά του με την κατάληξη. Μια σωστή μορφή της εγκατάστασης αποτελεί :

IBB_PMA_sample.xml



Σχήμα 14.C.1: Οθόνη εισαγωγής αρχείου ρυθμίσεων

Ακολούθως από την επιλογή και τον έλεγχο του αρχείου ξεκινάει η εγκατάσταση. Η εγκατάσταση και παραμετροποίηση γίνεται με γραφικό τρόπο μέσα από οθόνες επιλογών.

Οθόνη Πρώτη:

Στην πρώτη οθόνη εγκατάστασης, ο χρήστης καλείται να ορίσει ποιοι εξυπηρετητές θα χρησιμοποιηθούν κατά την ηλεκτρονική ψηφοφορία. Τα ονόματα των εξυπηρετητών μπορούν να τοποθετηθουν είται με DNS μορφή (π.χ. litw.teimes.gr), είτε με IP διεύθυνση (π.χ. 194.177.216.16).

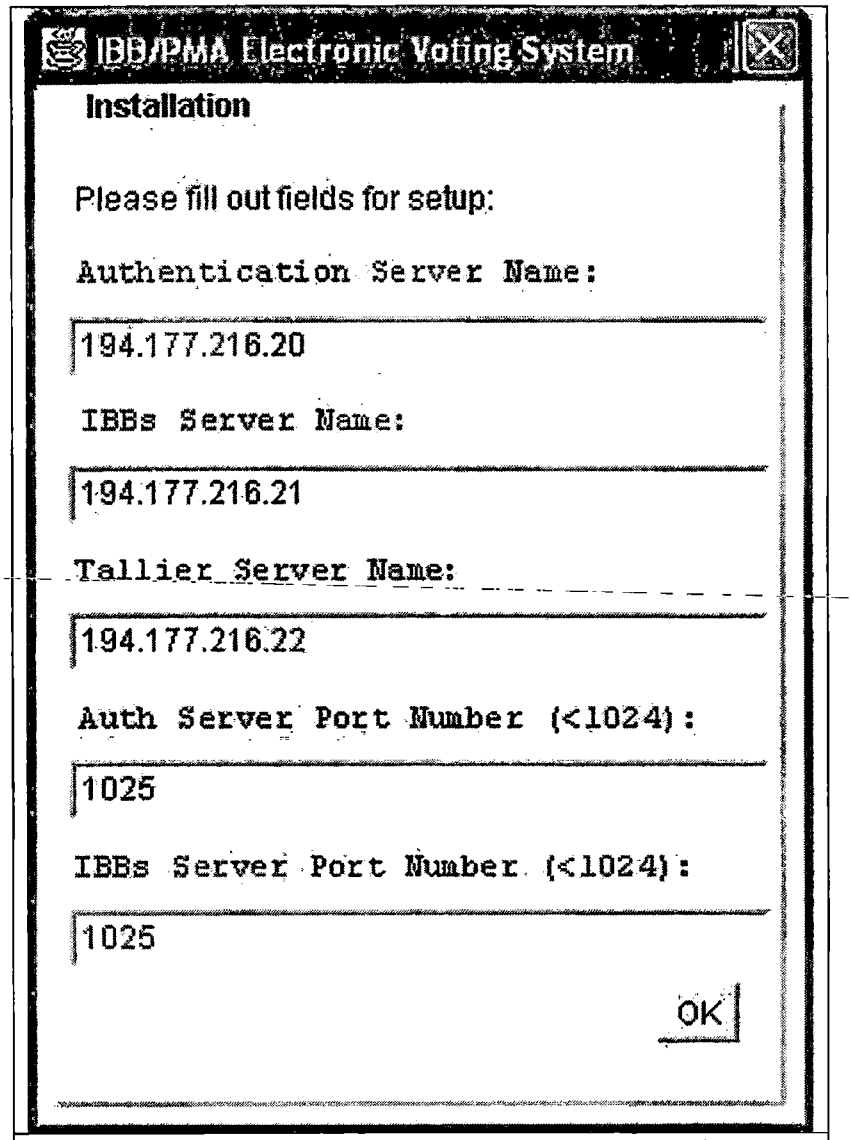
Authentication Server name: Ο εξυπηρετητής, στον οποίο θα τρέχει το πρόγραμμα IBBPMAServer.

IBBs Server name: Ο εξυπηρετητής, στον οποίο θα τρέχουν οι Πανομοιότυπες Κάλπες.

Tallier Server name: Ο εξυπηρετητής, στον οποίο θα στηθεί το πρόγραμμα της καταμέτρησης.

AuthServer port number: Η θύρα στην οποία θα επικοινωνούν οι υπηρεσίες του IBB/PMA συστήματος.

IBBServer port number: Η θύρα στην οποία θα στέλνονται τα πακέτα ψήφων για καταχώρηση



Σχήμα 14.C.2: Οθόνη Ορισμού Εξυπηρετητών

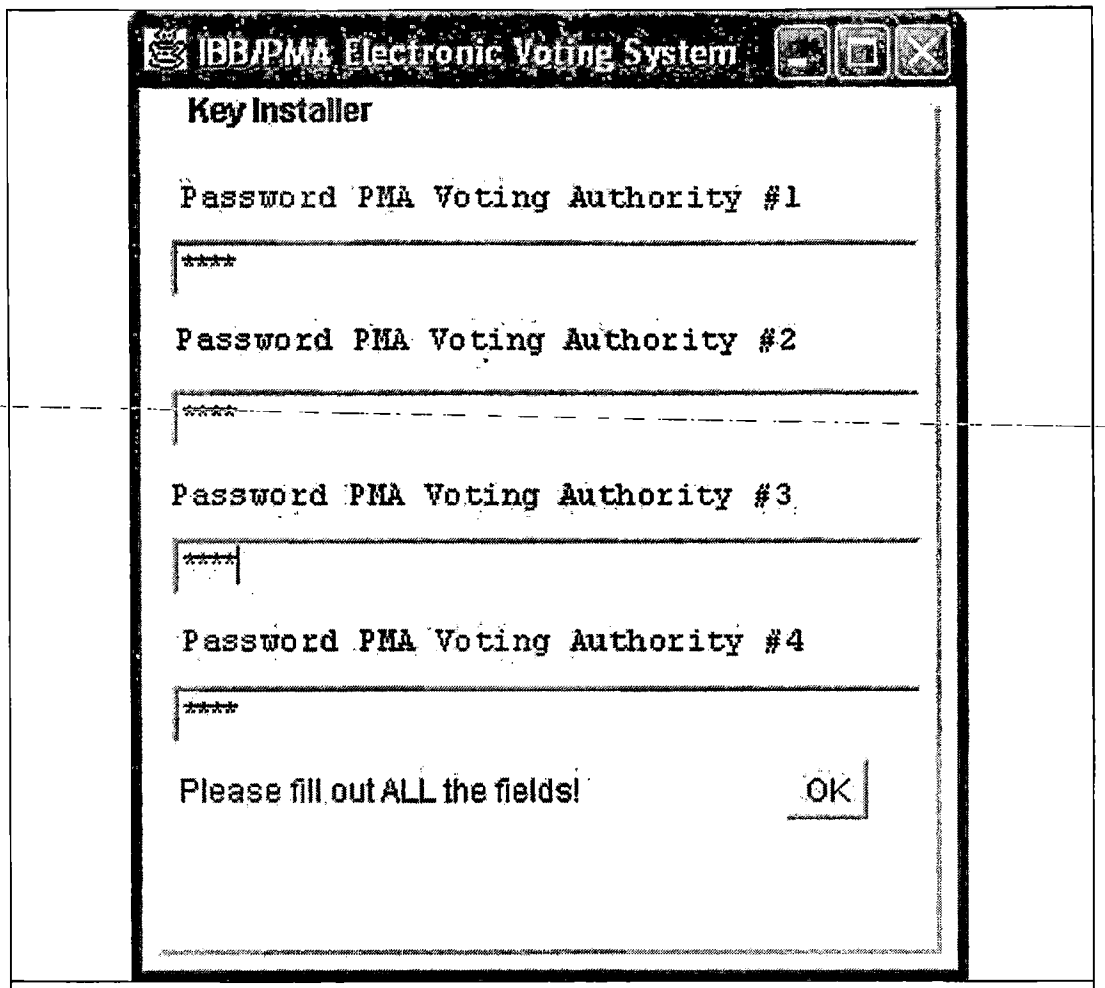
Οθόνη Δεύτερη:

Στην δεύτερη οθόνη εγκατάστασης, γίνεται η εισαγωγή των πληροφοριών που θα χρησιμοποιηθούν στην δεδομένη εκλογική διαδικασία. Την δεδομένη στιγμή είναι εφικτή η ολοκλήρωση έως και τεσσάρων υποψηφίων προς εκλογή. Είναι η ίδια με την διαδικασία εισαγωγής των ονομάτων στο XML αρχείο.

Οθόνη Τρίτη - Τέταρτη:

Στην Τρίτη – Τέταρτη οθόνη εγκατάστασης, γίνεται η εισαγωγή των απαραίτητων κωδικών από το σώμα της PMA Voting Authority. Σειριακά εισάγονται οι κωδικοί από τους τέσσερις αντιπροσώπους των υποψηφίων, από την IBB/PMA Voting Authority. Ο κωδικός που θα εισάγει ο πρώτος θα χρησιμοποιηθεί για την κατασκευή

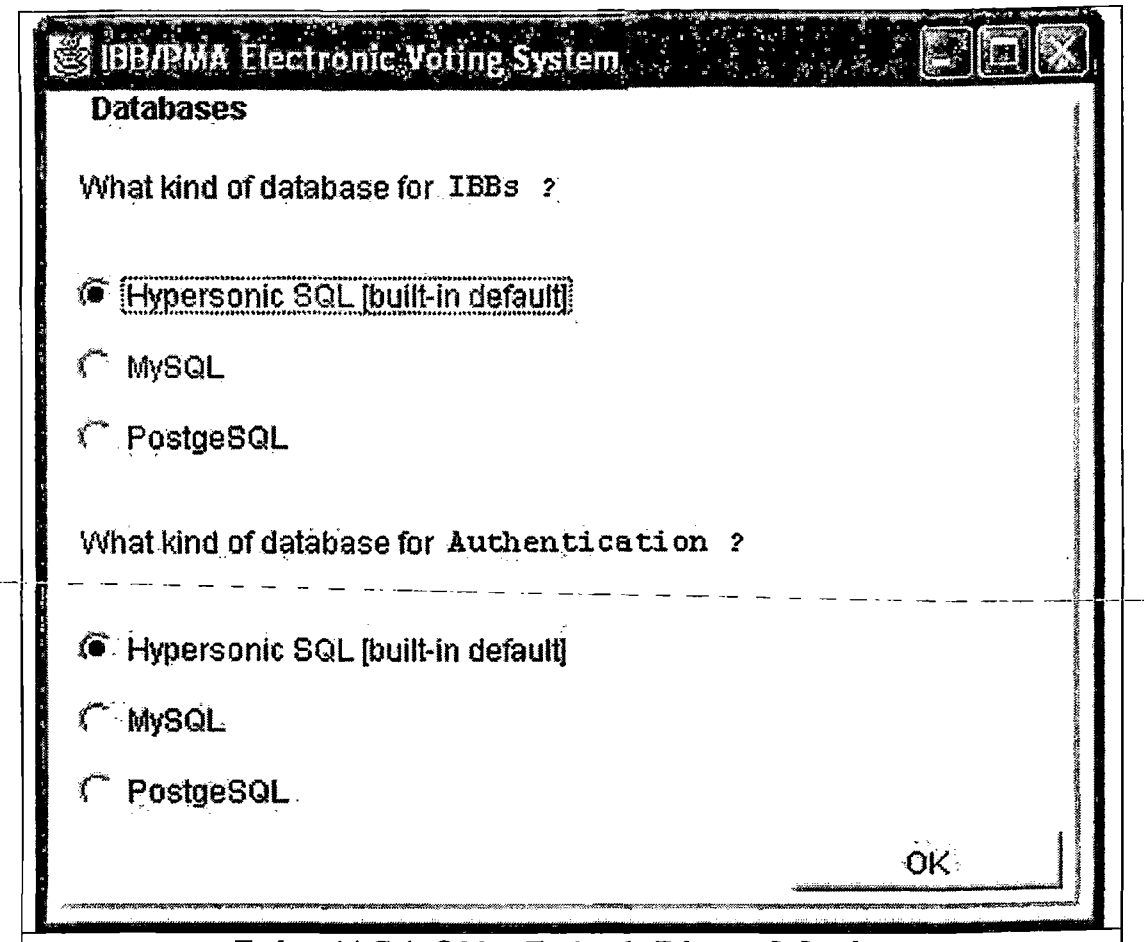
των Message Authorization Codes (MACs), ο κωδικός του δεύτερου θα χρησιμοποιηθεί για να δημιουργήσει το κλειδί συνεδρίας 1, ο κωδικός του τρίτου για το κλειδί συνεδρίας 2 και τέλος ο κωδικός του τελευταίου θα παράγει τα MAC που θα χρησιμοποιούνται για την κρυπτογράφηση των IBBs.



Σχήμα 14.C.3: Εισαγωγή κωδικών από την PMA Voting Authority

Οθόνη Πέμπτη:

Στην Πέμπτη οθόνη γίνεται η επιλογή της βάσης δεδομένων που θα χρησιμοποιηθεί για την εκλογική συνεδρία. Προτείνεται η χρήση της Hypersonic-SQL που είναι ενσωματωμένη, γιατί σε αυτήν έχει ολοκληρωθεί η τεχνολογία των πανομοιότυπων καλπών IBB. Στην περίπτωση που θα χρησιμοποιηθεί κάποιο άλλο DBMS θα πρέπει να έχει προγραμματιστεί με τέτοιο τρόπο που να μπορούν να εφαρμοστούν οι IBBs.



Σχήμα 14.C.4: Οθόνη Επιλογής Βάσεων δεδομένων

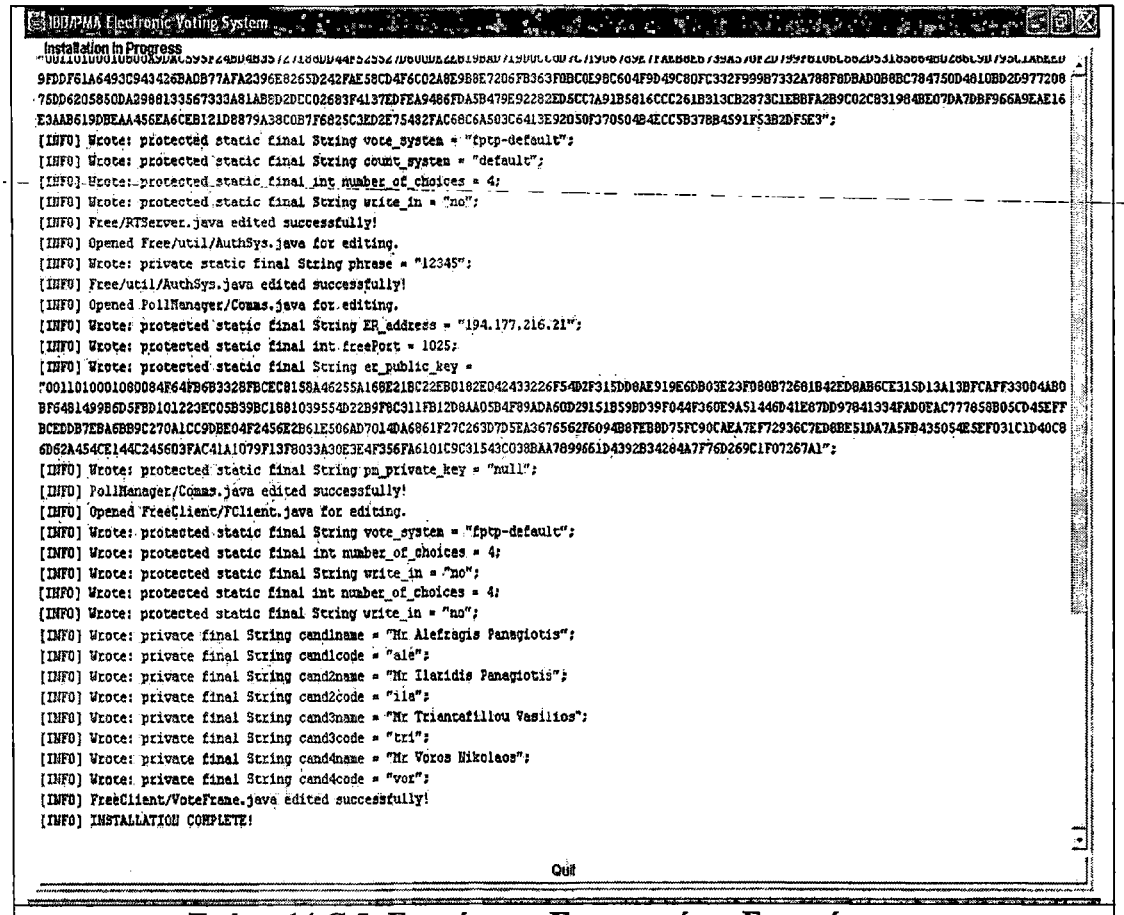
Οθόνη Πέμπτη (Β):

Αυτή η συμπληρωματική οθόνη χρησιμοποιείται για να οριστούν οι Βάσεις δεδομένων που θα χρησιμοποιηθούν εάν δεν χρησιμοποιηθεί η Hypersonic-SQL. Εμφανίζεται μόνο όταν επιλεγεί κάποια άλλη στην προηγούμενη οθόνη.

Οθόνη Έκτη:

Στην έκτη οθόνη γίνεται η διαδικασία της εγκατάστασης του IBB/PMA συστήματος. Ανάλογα με την έκδοση της Java που υποστηρίζεται, το λειτουργικό σύστημα και το υλικό (HW) που χρησιμοποιείται, η εγκατάσταση θα χρειαστεί και τον αντίστοιχο χρόνο. Για ένα μηχάνημα 2.6 Gh , με 1024 MB Ram και λειτουργικό Windows 2003 Server απαιτούνται περίπου 2 λεπτά για την ολοκλήρωση της εγκατάστασης.

Με την ολοκλήρωση της εγκατάστασης το IBB/PMA Electronic Voting System είναι έτοιμο για χρήση.



Σχήμα 14.C.5: Ενημέρωση Επιτυχημένης Εγκατάστασης

D. Εκτέλεση

Πλέον το IBB/PMA Electronic Voting System είναι έτοιμο για λειτουργία. Για να ξεκινήσει η εκτέλεσή του χρησιμοποιούμε τις ακόλουθες εντολές.

java -cp
IBBPMAutil.jar;IBBPMAawt.jar;IBBPMAdbpool.jar;hsql.jar;log4j.jar;cryptix32.jar;.
;%CLASSPATH% IBBPMA.RTServer

java -cp
IBBPMAutil.jar;IBBPMAawt.jar;IBBPMAdbpool.jar;hsql.jar;log4j.jar;cryptix32.jar;.
;%CLASSPATH% IBBPMAServer.IBBPMAServer

java -cp IBBPMAutil.jar;IBBPMAawt.jar;cryptix32.jar;.;%CLASSPATH%
PollManager.PollManager

java -cp IBBPMAutil.jar;IBBPMAawt.jar;cryptix32.jar;.;%CLASSPATH%
IBBPMAClient.IbPClient

14. Τεκμηρίωση του IBB/PMA συστήματος ηλεκτρονικής ψηφοφορίας

Α. Πακέτο Εγκατάστασης IBB/PMA

i. AuthInstallFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Μια οθόνη που επιτρέπει στους χρήστες να ορίσουν τις μεταβλητές του IBB/PMA
 * με ευκολία.
 *
 */

public class AuthInstallFrame extends Frame
{

        Label jLabel1 = new Label();
        Label jLabel4 = new Label();
        TextField jTextField1 = new TextField();
        Button jButton1 = new Button();
        Label jLabel5 = new Label();

        protected AuthInstallFrame() {

                jLabel4.setText("Please fill out fields for setup:");
                jLabel1.setText("Authorisation password:");
                jLabel5.setText("Please fill out ALL the fields!");
                jButton1.setLabel("OK");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("Authorisation Password");

                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, jLabel4);
                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, jTextField1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton1);
                add( scrollPanel);

                setTitle("IBB/PMA: Setup (3)");
                setResizable(true);
                setSize(new java.awt.Dimension(350, 180));
        }
```

```
        protected void initComponents() throws Exception
        {

                jButton1.addActionListener(new java.awt.event.ActionListener() {
                        public void actionPerformed(java.awt.event.ActionEvent e) {
                                jButton1ActionPerformed(e);
                        }
                });
                addWindowListener(new java.awt.event.WindowAdapter() {
                        public void windowClosing(java.awt.event.WindowEvent e) {
                                thisWindowClosing(e);
                        }
                });
        }

        // Κλείστε το παράθυρο όταν το close box επιλεγεί.
        void thisWindowClosing(java.awt.event.WindowEvent e)
        {
        }

        protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

                if (jTextField1.getText().equals("")) {
                        jLabel5.setVisible(true);
                } else {
                        if (Install.isSafe(jTextField1.getText())) {
                                setVisible(false);
                                jLabel5.setVisible(false);

                                Install.passphrase = jTextField1.getText();

                                Install.frame5.setVisible(true);
                                dispose();
                        } else {
                                jLabel5.setVisible(true);
                        }

                }

        }

} //EOF class
```

ii. DBChoiceFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Μια οθόνη που επιτρέπει στους χρήστες να ορίσουν τις μεταβλητές του IBB/PMA
 * με ευκολία.
 *
 */

public class DBChoiceFrame extends Frame {

        private final CheckboxGroup group = new CheckboxGroup();
        private final CheckboxGroup group2 = new CheckboxGroup();

        Checkbox jRadioButton1;
        Checkbox jRadioButton2;
        Checkbox jRadioButton3;
        Checkbox jRadioButton21;
        Checkbox jRadioButton22;
        Checkbox jRadioButton23;
        Label jLabel1 = new Label();
        Label jLabel2 = new Label();
        Button jButton1 = new Button();

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}

        protected DBChoiceFrame() {

                setTitle("IBB/PMA: Setup (5)");
                jRadioButton1 = new Checkbox("Hypersonic SQL [built-in default]", group, true);
                jRadioButton2 = new Checkbox("MySQL", group, false);
                jRadioButton3 = new Checkbox("PostgeSQL", group, false);
                jRadioButton21 = new Checkbox("Hypersonic SQL [built-in default]", group2, true);
                jRadioButton22 = new Checkbox("MySQL", group2, false);
                jRadioButton23 = new Checkbox("PostgeSQL", group2, false);
                jButton1.setLabel("OK");
                jLabel1.setText("What kind of database for RTServer?");
                jLabel2.setText("What kind of database for IBBPMAServer?");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("Databases");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 1, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jRadioButton1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jRadioButton2);
                scrollPanel.addComponent( 4, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jRadioButton3);
                scrollPanel.addComponent( 5, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, jLabel2);
                scrollPanel.addComponent( 6, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
```

```
                1.0f, 0.0f, jRadioButton21);
        scrollPanel.addComponent( 7, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jRadioButton22);
        scrollPanel.addComponent( 8, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jRadioButton23);
        scrollPanel.addComponent( 9, 1, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jButton1);

        add(scrollPanel, BorderLayout.CENTER);
        setSize(new java.awt.Dimension(425, 375));

}

class itemListener implements ItemListener{
        public void itemStateChanged(ItemEvent e){
                if(e.getSource() == jRadioButton1){
                        jRadioButton1StateChanged(e);
                }
                else if(e.getSource() == jRadioButton2){
                        jRadioButton2StateChanged(e);
                }
                else if(e.getSource() == jRadioButton3){

                        jRadioButton3StateChanged(e);
                }
                if(e.getSource() == jRadioButton21){
                        jRadioButton21StateChanged(e);
                }
                else if(e.getSource() == jRadioButton22){

                        jRadioButton22StateChanged(e);
                }
                else if(e.getSource() == jRadioButton23){

                        jRadioButton23StateChanged(e);
                }
        }

}

protected void initComponents() throws Exception
{

        jRadioButton1.addItemListener(new itemListener());
        jRadioButton2.addItemListener(new itemListener());
        jRadioButton3.addItemListener(new itemListener());
        jRadioButton21.addItemListener(new itemListener());
        jRadioButton22.addItemListener(new itemListener());
        jRadioButton23.addItemListener(new itemListener());

        jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton1ActionPerformed(e);
                }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
                public void windowClosing(java.awt.event.WindowEvent e) {
                        thisWindowClosing(e);
                }
```

```
            });
        }

        void thisWindowClosing(java.awt.event.WindowEvent e)
        {
        }

        public void jRadioButton1StateChanged(ItemEvent e) {
            Install.rt_dbtype = 3; // HSQL
        }

        public void jRadioButton2StateChanged(ItemEvent e) {
            Install.rt_dbtype = 1; // MySQL
        }

        public void jRadioButton3StateChanged(ItemEvent e) {
            Install.rt_dbtype = 2; // PostgreSQL
        }

        public void jRadioButton21StateChanged(ItemEvent e) {
            Install.er_dbtype = 3; // HSQL
        }

        public void jRadioButton22StateChanged(ItemEvent e) {
            Install.er_dbtype = 1; // MySQL
        }

        public void jRadioButton23StateChanged(ItemEvent e) {
            Install.er_dbtype = 2; // PostgreSQL
        }

        protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {
            if (Install.er_dbtype!=3) {
                setVisible(false);
                Install.frame7.setVisible(true);
                dispose();
            } else if (Install.rt_dbtype!=3) {
                setVisible(false);
                Install.frame8.setVisible(true);
                dispose();
            } else {
                // Δεν χρειάζεται η εγκατάσταση των DBs
                setVisible(false);
                Install.frame.setVisible(true);
                Install.doInstallation();
                dispose();
            }
        }

} //EOF class
```

iii. ElectionDefinition.java

```
package IBBPMAInstall;

import org.apache.xerces.parsers.DOMParser;
import org.xml.sax.ErrorHandler;
import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import org.w3c.dom.NamedNodeMap;
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;
import org.xml.sax.SAXNotRecognizedException;
import org.xml.sax.SAXNotSupportedException;
import java.io.IOException;

/**
 * Ο <code>ElectionDefinition</code> παρέχει τις απαραίτητες μεθόδους για την
 * ανάγνωση και χρήση του Election Definition
 * από τα αρχεία στα XML αρχεία που περιέχουν τις απαραίτητες πληροφορίες
 * Ο κώδικας είναι βασισμένος στο tutorial at http://www.ecerami.com and from
 * O'Reilly "Java & XML"
 */

public class ElectionDefinition implements ErrorHandler {

  // Constructor
  public ElectionDefinition (String xmlFile) {
    // Δημιουργεί το Xerces DOM Parser
    DOMParser parser = new DOMParser();

    // Turn Validation on
    try {
      parser.setFeature("http://xml.org/sax/features/validation", true);
    } catch (SAXNotRecognizedException e) {
            Install.NORM.error("Election Definition error (SAXNotRecognized): " + e);
    } catch (SAXNotSupportedException e) {
            Install.NORM.error("Election Definition error (SAXNotSupported): " + e);
    }

    // Εγραφή του Error Handler
    parser.setErrorHandler(this);

    // Πέρασμα του Αρχείου
    try {
      parser.parse(xmlFile);
      Document document = parser.getDocument();
      getInstallVars(document);
    } catch (SAXException e) {
            Install.NORM.error("Election Definition error: " + e);
    } catch (IOException e) {
            Install.NORM.error("Election Definition error: " + e);
    }
  } // eof constructor

  /**
   * Εύρεση κάθε ιδιότητες που χρειάζεται για την εγκατάσταση, συλλογή όλων
   * των ιδιοτήτων και
```

```
     * των δεδομένων για αποθήκευση και περεταίρω χρήση.
     *
     * @param  doc Ένα αρχείο στο οποίο θα γίνει η εύρεση για tags.
     */

    private void getInstallVars(Document doc) {

        NodeList result = doc.getElementsByTagName("IBBPMA:name");
      if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_name = childnodes.item(0).getNodeValue(); // αποθήκευση τιμής
      } else {
        Install.NORM.error("Election Definition error: No name specified!");
      }

        result = doc.getElementsByTagName("IBBPMA:jurisdiction");
      if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_jurisdiction = childnodes.item(0).getNodeValue(); // αποθήκευση τιμής
      } else {
        Install.NORM.error("Election Definition error: No jurisdiction specified!");
      }

        result = doc.getElementsByTagName("IBBPMA:contact_email");
      if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_contact_email = childnodes.item(0).getNodeValue(); // αποθήκευση τιμής
      } else {
        Install.NORM.error("Election Definition error: No contact email specified!");
      }

        result = doc.getElementsByTagName("IBBPMA:website");
      if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_website = childnodes.item(0).getNodeValue(); // αποθήκευση τιμής
      } else {
        Install.NORM.error("Election Definition error: No website specified!");
      }

        result = doc.getElementsByTagName("IBBPMA:jnlp");
                NamedNodeMap attributes = result.item(0).getAttributes();
                for (int x=0; x< attributes.getLength(); x++) {
                        Node current = attributes.item(x);
                        Install.edl_jnlp_attr[x] = new
String(current.getNodeValue()).toLowerCase(); // αποθήκευση ιδιότητας
                }

        result = doc.getElementsByTagName("IBBPMA:auth_system");
      if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_auth_system = childnodes.item(0).getNodeValue(); // αποθήκευση τιμής

                        attributes = result.item(0).getAttributes();
                        for (int x=0; x< attributes.getLength(); x++) {
                                Node current = attributes.item(x);
                                Install.edl_auth_system_attr[x] = new
String(current.getNodeValue()).toLowerCase(); // αποθήκευση ιδιότητας
                        }
      } else {
        Install.NORM.error("Election Definition error: No auth system specified!");
```

```
    }

        result = doc.getElementsByTagName("IBBPMA:vote_system");
    if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_vote_system = childnodes.item(0).getNodeValue(); // αποθήκευση τιμής

                        attributes = result.item(0).getAttributes();
                        for (int x=0; x< attributes.getLength(); x++) {
                                Node current = attributes.item(x);
                                Install.edl_vote_system_attr[x] = new
String(current.getNodeValue()).toLowerCase(); // αποθήκευση ιδιότητας
                        }
    } else {
        Install.NORM.error("Election Definition error: No vote system specified!");
    }

        result = doc.getElementsByTagName("IBBPMA:count_system");
    if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_count_system = childnodes.item(0).getNodeValue(); // αποθήκευση τιμής

                        attributes = result.item(0).getAttributes();
                        for (int x=0; x< attributes.getLength(); x++) {
                                Node current = attributes.item(x);
                                Install.edl_count_system_attr[x] = new
String(current.getNodeValue()).toLowerCase(); // αποθήκευση ιδιότητας
                        }
    } else {
        Install.NORM.error("Election Definition error: No count system specified!");
    }

        result = doc.getElementsByTagName("IBBPMA:write_in");
    if (result != null) {
        NodeList childnodes = result.item(0).getChildNodes();
                Install.edl_write_in = new
String(childnodes.item(0).getNodeValue()).toLowerCase(); // αποθήκευση τιμής
    } else {
        Install.NORM.error("Election Definition error: No write in option specified!");
    }

        result = doc.getElementsByTagName("IBBPMA:choice");
    if (result != null) {
        Install.edl_number_of_choices = result.getLength();
       for (int i=0; i< result.getLength(); i++) {
        NodeList childnodes = result.item(i).getChildNodes();
        for (int z=0; z< childnodes.getLength(); z++) {
                        Install.edl_choice[i] = childnodes.item(z).getNodeValue(); // αποθήκευση
τιμής
                        }
                attributes = result.item(i).getAttributes();
                for (int x=0; x< attributes.getLength(); x++) {
                    Node current = attributes.item(x);
                    Install.edl_choice_attr[i][x] = new
String(current.getNodeValue()).toLowerCase(); // αποθήκευση ιδιότητας
                }
            }
    } else {
        Install.NORM.error("Election Definition error: No choice specified!");
    }
```

```
        } // EOF getInstallVars

    // Warning Event Handler
    public void warning (SAXParseException e)
      throws SAXException {
      Install.NORM.warn("Election Definition warning: " + e);
    }

    // Διαχειριστής Λαθών
    public void error (SAXParseException e)
      throws SAXException {
      Install.NORM.error("Election Definition error: " + e);
    }

    // Διαχειριστής Fatal Λαθών
    public void fatalError (SAXParseException e)
      throws SAXException {
      Install.NORM.fatal("Election Definition fatal error: " + e);
    }
-------------------------------------------------------------------------------
} //EOF class
```

iv. ERDBFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Μια οθόνη που επιτρέπει στους χρήστες να ορίσουν τις μεταβλητές του IBB/PMA
 * με ευκολία.
 */

public class ERDBFrame extends Frame {

        Label jLabel1 = new Label();
        Label jLabel2 = new Label();
        Label jLabel3 = new Label();
        Label jLabel4 = new Label();
        Label jLabel6 = new Label();
        Label jLabel7 = new Label();
        TextField jTextField1 = new TextField();
        TextField jTextField2 = new TextField();
        TextField jTextField3 = new TextField();
        TextField jTextField4 = new TextField();
        Button jButton1 = new Button();
        Label jLabel5 = new Label();

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}

        protected ERDBFrame() {

                jLabel4.setText("Please fill out fields for setup:");
                jLabel1.setText("IBB/PMA Database username:");
                jLabel2.setText("IBB/PMA Database password:");
                jLabel3.setText("IBB/PMA Database name:");
                jLabel7.setText("IBB/PMA Database server name:");
                jLabel5.setText("Please fill out ALL the fields!");
                jButton1.setLabel("OK");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("IBB/PMA Database Setup");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel4);
                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel2);
                scrollPanel.addComponent( 4, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField2);
                scrollPanel.addComponent( 5, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel3);
                scrollPanel.addComponent( 6, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
```

```
                1.0f, 0.0f, jTextField3);
        scrollPanel.addComponent( 7, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jLabel7);
        scrollPanel.addComponent( 8, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jTextField4);
        scrollPanel.addComponent( 9, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jLabel5);
        scrollPanel.addComponent( 9, 1, 1, 1, 0, 0, GridBagConstraints.NONE,
                1.0f, 0.0f, jButton1);

        add(scrollPanel);
        setTitle("IBB/PMA: Setup (5a)");
        setResizable(true);
        setSize(new java.awt.Dimension(320, 380));

    }

    protected void initComponents() throws Exception
    {
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                jButton1ActionPerformed(e);
            }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                thisWindowClosing(e);
            }
        });
    }

    void thisWindowClosing(java.awt.event.WindowEvent e)
    {
    }

    protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

        if
(jTextField1.getText().equals("")|jTextField2.getText().equals("")|jTextField3.getText().equals("")|jTe
xtField4.getText().equals("")) {
            jLabel5.setVisible(true);
        } else {
            if
(Install.isSafe(jTextField1.getText())|Install.isSafe(jTextField2.getText())|Install.isSafe(jTextField3.get
Text())|Install.isSafe(jTextField4.getText())) {
                setVisible(false);
                jLabel5.setVisible(false);

                Install.er_dbuser = jTextField1.getText();
                Install.er_dbpass = jTextField2.getText();
                Install.er_dbname = jTextField3.getText();
                Install.er_dbhostname = jTextField4.getText();

                if (Install.rt_dbtype!=3) {
                    setVisible(false);
                    Install.frame8.setVisible(true);
                    dispose();
                } else {
                    setVisible(false);
```

```
                                        Install.frame.setVisible(true);
                                        Install.doInstallation();
                                        dispose();
                                }
                        } else {
                                jLabel5.setVisible(true);
                        }
                }

}

} //EOF class
```

v. FileFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Κλάση που προτρέπει τον χρήστη να δώσει το όνομα του XML config αρχείου
 *
 */

public class FileFrame extends Frame {

	Label jLabel1 = new Label();
	TextField jTextField1 = new TextField();
	Button jButton1 = new Button();

	public Insets getInsets() {

		return new Insets( 30, 10, 10, 10);
	}

	protected FileFrame() {

		setTitle("IBB/PMA: Install");
		jButton1.setLabel("OK");
		jLabel1.setText("Please enter the name of your config file e.g. example.xml");

		IBBPMAPanel scrollPanel = new IBBPMAPanel("IBB/PMA: Config file");
		scrollPanel.setInsets(3, 3, 3, 3);

		scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
			1.0f, 1.0f, jLabel1);
		scrollPanel.addComponent( 2, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
			1.0f, 0.0f, jTextField1);
		scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
			1.0f, 0.0f, jButton1);

		add(scrollPanel, BorderLayout.CENTER);
		setSize(new java.awt.Dimension(400, 175));

	}

	public void initComponents() throws Exception {

		jButton1.addActionListener(new java.awt.event.ActionListener() {
			public void actionPerformed(java.awt.event.ActionEvent e) {
				jButton1ActionPerformed(e);
			}
		});
		addWindowListener(new java.awt.event.WindowAdapter() {
			public void windowClosing(java.awt.event.WindowEvent e) {
				thisWindowClosing(e);
			}
		});
	}
```

```
        void thisWindowClosing(java.awt.event.WindowEvent e)
        {
        }

        public void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

                String input = new String(jTextField1.getText());

                if (!input.equals("")) {
                        try {
                                Install.config_filename=input;
                                setVisible(false);
                                Install.frame2.setVisible(true);

                                dispose();
                        } catch (Exception ex) {
                                setVisible(false);
                                dispose();
                                System.exit(1);
                        }
                }

        }

} //EOF Class
```

vi. Install.java

```
package IBBPMAInstall;

import java.security.*;
import java.awt.*;
import java.io.*;
import java.util.*;
import IBBPMA.awt.*;

import cryptix.provider.rsa.RawRSAPrivateKey;
import cryptix.provider.rsa.RawRSAPublicKey;
import cryptix.util.core.Hex;

import org.apache.log4j.*;

/**
 * Η κλάση <code>Install</code> παρέχει έναν γρήγορο γραφικό τρόπο της
 * προσαρμογής
 * του IBB/PMA για μια εκλογική διαδικασία.
 *
 * Υπάρχουν τα απαραίτητα AWT frames που καθοδηγούν τον χρήστη με τις
 * λεπτομέρειες για κάθε μέρος της εγκατάστασης προτού την τελική επιλογή.
 * Ο πηγαίος κώδικας αυτόματα παίρνεται back up σε ένα .bak αρχείο, που
 * διαγράφεται μετά από επιτυχημένες αλλαγές.
 * Επιπρόσθετα όλες οι αλλαγές καταγράφονται στο <code>IBBPMAInstall.log</code>
 */

public class Install {

        private static String[] installStrings;
        private static int installCounter;

        private static char quoteMark = '"';

        // logging Κατηγορίες
        protected static Category NORM;
        protected static Category DEV;

        // AWT frames
        protected static RTDBFrame frame8;
        protected static ERDBFrame frame7;
        protected static DBChoiceFrame frame6;
        protected static KeyInstallFrame frame5;
        protected static AuthInstallFrame frame4;
        protected static VoteInstallFrame frame3;
        protected static InstallFrame frame2;
        protected static StatusFrame frame;
        protected static FileFrame frame9;

        // Install variables
        protected static String regional_name;
        protected static String totaller_name;
        protected static String er_name;
        protected static String freePort;
        protected static String freeRTPort;
        protected static String party1;
        protected static String party2;
        protected static String party3;
```

```
protected static String party4;
protected static String passphrase;
protected static String secret_key1;
protected static String secret_key2;
protected static String MAC_key;
protected static String er_dbuser;
protected static String er_dbpass;
protected static int er_dbtype;
protected static String er_dbname;
protected static String er_dbhostname;
protected static String rt_dbuser;
protected static String rt_dbpass;
protected static int rt_dbtype;
protected static String rt_dbname;
protected static String rt_dbhostname;
protected static String rt_private_key;
protected static String er_private_key;
protected static String fc_private_key;
protected static String pm_private_key;
protected static String rt_public_key;
protected static String er_public_key;
protected static String fc_public_key;
protected static String pm_public_key;

// EDL variables
protected static String config_filename;
protected static String edl_name;
protected static String edl_jurisdiction;
protected static String edl_contact_email;
protected static String edl_website;
protected static String[] edl_jnlp_attr;
protected static String edl_auth_system;
protected static String[] edl_auth_system_attr;
protected static String edl_vote_system;
protected static String[] edl_vote_system_attr;
protected static String edl_count_system;
protected static String[] edl_count_system_attr;
protected static String edl_write_in;
protected static int edl_number_of_choices;
protected static String[] edl_choice;
protected static String[][] edl_choice_attr;

public Install() {

}

static public void main(String[] args) {
        IBBPMAPanel.showSplash(4000);

        // Αρχικοποίηση του logging system
        ScreenAppender A1 = new ScreenAppender();
        PropertyConfigurator.configure("log4j.install.properties");
        NORM = Category.getInstance("NORM");
        NORM.addAppender(A1);
        DEV = Category.getInstance("DEV");
        DEV.addAppender(A1);

        // default vals
        er_dbtype=3;
        rt_dbtype=3;
```

```
            installCounter = 0;

            edl_jnlp_attr = new String[5];
            edl_auth_system_attr = new String[5];
            edl_vote_system_attr = new String[5];
            edl_count_system_attr = new String[5];
            edl_choice = new String[50];
            edl_choice_attr = new String[50][5];

            installStrings = new String[100];

            try {
                    frame = new StatusFrame();
                    frame.initComponents();
                    frame2 = new InstallFrame();
                    frame2.initComponents();
                    frame3 = new VoteInstallFrame();
                    frame3.initComponents();
                    frame4 = new AuthInstallFrame();
                    frame4.initComponents();
                    frame5 = new KeyInstallFrame();
                    frame5.initComponents();
                    frame6 = new DBChoiceFrame();
                    frame6.initComponents();
                    frame7 = new ERDBFrame();
                    frame7.initComponents();
                    frame8 = new RTDBFrame();
                    frame8.initComponents();
                    frame9 = new FileFrame();
                    frame9.initComponents();
                    frame9.setVisible(true);

            } catch (Exception e) {
                    NORM.error("Startup Error: " + e.getMessage());
            }

    } //eof main

    /**
     * Δημιουργούνται τα Public and Private set κλειδιών για την RSA
* κρυπτογράφηση.
     *
     * @returns Έναν πίνακα αλφαριθμητικών με 0 για ιδιωτικό και 1 για
* δημόσιο, hex κωδικοποιημένα.
     */

    private static String[] buildRSAKeyPair() {

            String[] reskeys = new String[] {"", ""};

try {

  KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA","Cryptix");

  // Αριθμός των bit στο κλειδί.
  kpg.initialize(2048);

  NORM.info("Creating keypair. This may take a while.");
  KeyPair pair = kpg.generateKeyPair();
```

```
    RawRSAPrivateKey seckey = (RawRSAPrivateKey)pair.getPrivate();
    RawRSAPublicKey  pubkey = (RawRSAPublicKey) pair.getPublic ();

                        /* μετατροπή σε hex */
    reskeys[0] = Hex.toString(seckey.getEncoded());
    reskeys[1] = Hex.toString(pubkey.getEncoded());

} catch (Exception e) {
                        NORM.error("buildRSAKeyPair error: " + e.getMessage());
}

            return reskeys;

    } //EOF builRSAKeyPair


    /**
     * Δημιουργεί ένα κλειδί από μια αλφαριθμητική είσοδο με χρήση message
* digest, ημερομηνία και ένα secure random.
     *
     */

    private static String buildKey(String pass) {

            String result="";

            try {
                        String data = new Date().toString();

                        SecureRandom sr = new SecureRandom();
                        byte[] pseudoRandom = new byte[100];
                        sr.nextBytes(pseudoRandom);

                        MessageDigest md = MessageDigest.getInstance("SHA-1");

                        byte dataBytes[] = data.getBytes();
                        byte passBytes[] = pass.getBytes();
                        md.update(passBytes);
                        md.update(dataBytes);
                        md.update(pseudoRandom);
                        byte digest1[] = md.digest();

                        md.update(passBytes);
                        md.update(digest1);
                        byte mac[] = md.digest();

                        /* μετατροπή σε hex */
                        result = Hex.toString(mac);

             } catch (Exception e) {
                        NORM.error("buildKey error: " + e.getMessage());
            }

            return result;

    } //eof buildKey


    /**
     * @param input  Το αλφαριθμητικό για έλεγχο
```

```
* @returns  True εάν το αλφαριθμητικό είναι OK
*/
protected static boolean isSafe(String input) {

        boolean ok = true;

        for(int i = 0; (i<input.length())&&(ok==true); i++) {

                switch (input.charAt(i)) {
                        case 'a': ok = true; break;
                        case 'b': ok = true; break;
                        case 'c': ok = true; break;
                        case 'd': ok = true; break;
                        case 'e': ok = true; break;
                        case 'f': ok = true; break;
                        case 'g': ok = true; break;
                        case 'h': ok = true; break;
                        case 'i': ok = true; break;
                        case 'j': ok = true; break;
                        case 'k': ok = true; break;
                        case 'l': ok = true; break;
                        case 'm': ok = true; break;
                        case 'n': ok = true; break;
                        case 'o': ok = true; break;
                        case 'p': ok = true; break;
                        case 'q': ok = true; break;
                        case 'r': ok = true; break;
                        case 's': ok = true; break;
                        case 't': ok = true; break;
                        case 'u': ok = true; break;
                        case 'v': ok = true; break;
                        case 'w': ok = true; break;
                        case 'x': ok = true; break;
                        case 'y': ok = true; break;
                        case 'z': ok = true; break;
                        case 'A': ok = true; break;
                        case 'B': ok = true; break;
                        case 'C': ok = true; break;
                        case 'D': ok = true; break;
                        case 'E': ok = true; break;
                        case 'F': ok = true; break;
                        case 'G': ok = true; break;
                        case 'H': ok = true; break;
                        case 'I': ok = true; break;
                        case 'J': ok = true; break;
                        case 'K': ok = true; break;
                        case 'L': ok = true; break;
                        case 'M': ok = true; break;
                        case 'N': ok = true; break;
                        case 'O': ok = true; break;
                        case 'P': ok = true; break;
                        case 'Q': ok = true; break;
                        case 'R': ok = true; break;
                        case 'S': ok = true; break;
                        case 'T': ok = true; break;
                        case 'U': ok = true; break;
                        case 'V': ok = true; break;
                        case 'W': ok = true; break;
                        case 'X': ok = true; break;
                        case 'Y': ok = true; break;
```

```
                        case 'Z': ok = true; break;
                        case '1': ok = true; break;
                        case '2': ok = true; break;
                        case '3': ok = true; break;
                        case '4': ok = true; break;
                        case '5': ok = true; break;
                        case '6': ok = true; break;
                        case '7': ok = true; break;
                        case '8': ok = true; break;
                        case '9': ok = true; break;
                        case '0': ok = true; break;
                        case ';': ok = true; break;
                        case ')': ok = true; break;
                        case '(': ok = true; break;
                        case '*': ok = true; break;
                        case ' ': ok = true; break;
                        case '=': ok = true; break;
                        case '-': ok = true; break;
                        case '_': ok = true; break;
                        case '+': ok = true; break;
                        case '@': ok = true; break;
                        case ',': ok = true; break;
                        case '.': ok = true; break;
                        case '<': ok = true; break;
                        case '>': ok = true; break;
                        case '?': ok = true; break;
                        default:  ok = false; break;
                    } //eof case

                } //eof for

                return ok;

        } //eof isSafe


        /**
         * Ορίζει τα αλφαριθμητικά εγκατάστασης και επεξεργάζεται τα αρχεία.
         *
         */

        protected static void doInstallation() {

                NORM.info("IBB/PMA v1.0");

                /* Διαβάζει από το XML config */
                NORM.info("Reading XML configuration file...");
                ElectionDefinition install_ed = new ElectionDefinition(config_filename);
                NORM.info("File read complete!");

                try {
                        java.security.Security.addProvider(new cryptix.provider.Cryptix());
                } catch(Exception e) {
                        NORM.error("Cryptix initialistion error: " + e.toString());
                }

                /* Κατασκευή των κλειδιών */
                NORM.info("Please wait... building Authorisation session keys.");
                secret_key1 = buildKey(secret_key1);
                secret_key2 = buildKey(secret_key2);
```

```
                NORM.info("Keys built.");

                NORM.info("Please wait... building RSA keys.");
                String temp[] = new String[] {"","",""};
                temp = buildRSAKeyPair();
                rt_private_key = temp[0];
                rt_public_key = temp[1];
                temp = buildRSAKeyPair();
                er_private_key = temp[0];
                er_public_key = temp[1];
                NORM.info("Keys built.");

                /* IBBPMAClient.Comms strings Εγκατάστασης */
                installStrings[0] = "protected static final String R_address = " + quoteMark +
regional_name +  quoteMark + ";";
                installStrings[1] = "protected static final String ER_address = " +  quoteMark +
er_name + quoteMark +  ";";
                installStrings[2] = "protected static final int freePort = " + freePort + ";";
                installStrings[3] = "protected static final int freeRTPort = " + freeRTPort + ";";
                installStrings[4] = "protected static final String rt_public_key = " + quoteMark +
rt_public_key + quoteMark + ";";
                installStrings[5] = "protected static final String er_public_key = " + quoteMark +
er_public_key + quoteMark + ";";
                installStrings[6] = "protected static final String fc_private_key = " + quoteMark +
fc_private_key + quoteMark + ";";
                /* IBBPMAServer.IBBPMAServer strings Εγκατάστασης*/
                installStrings[7] = "private static final int freePort = " + freePort + ";";
                installStrings[8] = "protected static final String rt_public_key = " + quoteMark +
rt_public_key + quoteMark + ";";
                installStrings[9] = "protected static final String fc_public_key = " + quoteMark +
fc_public_key + quoteMark + ";";
                installStrings[10] = "protected static final String pm_public_key = " + quoteMark +
pm_public_key + quoteMark + ";";
                installStrings[11] = "protected static final String er_private_key = " + quoteMark +
er_private_key + quoteMark + ";";
                /* IBBPMAServer.AuthKey install strings */
                installStrings[12] = "private static final String skey1 = " + quoteMark + secret_key1
+ quoteMark + ";";
                installStrings[13] = "private static final String skey2 = " + quoteMark + secret_key2
+ quoteMark + ";";
                installStrings[14] = "private static final String MACpass = " + quoteMark +
MAC_key + quoteMark + ";";
                /* IBBPMAServer.DBase strings Εγκατάστασης*/
                if (er_dbtype==3) {
                        installStrings[15] = "import org.hsql.*;";
                } else {
                        installStrings[15] = "";
                }
                installStrings[16] = "private static final String username = " + quoteMark +
er_dbuser + quoteMark + ";";
                installStrings[17] = "private static final String password = " + quoteMark +
er_dbpass + quoteMark + ";";
                if (er_dbtype==1) {
                        installStrings[18] = "Class.forName(" + quoteMark +
"org.gjt.mm.mysql.Driver" + quoteMark + ").newInstance(); //Φορτώνει το MySQL JDBC driver";
                } else if (er_dbtype==2) {
                        installStrings[18] = "Class.forName(" + quoteMark +
"org.postgresql.Driver" + quoteMark + "); //Φορτώνει το PostgreSQL JDBC driver";
                } else {
```

```
                    installStrings[18] = "Class.forName(" + quoteMark + "org.hsql.jdbcDriver"
+ quoteMark + "); //Φορτώνει το Hypersonic SQL JDBC driver";
                }
                if (er_dbtype==1) {
                    installStrings[19] = "new JDCConnectionDriver(" + quoteMark +
"org.gjt.mm.mysql.Driver" + quoteMark + "," + quoteMark + "jdbc:mysql://" + er_dbhostname + "/" +
er_dbname + quoteMark + ",username, password);";
                } else if (er_dbtype==2) {
                    installStrings[19] = "new JDCConnectionDriver(" + quoteMark +
"org.postgresql.Driver" + quoteMark + "," + quoteMark + "jdbc:postgresql://" + er_dbhostname + "/" +
er_dbname + quoteMark + ",username, password);";
                } else {
                    installStrings[19] = "new JDCConnectionDriver(" + quoteMark +
"org.hsql.jdbcDriver" + quoteMark + "," + quoteMark + "jdbc:HypersonicSQL:IBBPMAServer" +
quoteMark + "," + quoteMark + "sa" + quoteMark + "," + quoteMark + quoteMark + ");";
                }
                /* IBBPMA.DBase strings Εγκατάστασης*/
                if (rt_dbtype==3) {
                    installStrings[20] = "import org.hsql.*;";
                } else {
                    installStrings[20] = "";
                }
                installStrings[21] = "private static final String username = " + quoteMark + rt_dbuser
+ quoteMark + ";";
                installStrings[22] = "private static final String password = " + quoteMark + rt_dbpass
+ quoteMark + ";";
                if (rt_dbtype==1) {
                    installStrings[23] = "Class.forName(" + quoteMark +
"org.gjt.mm.mysql.Driver" + quoteMark + ").newInstance(); //Φορτώνει το MySQL JDBC driver";
                } else if (rt_dbtype==2) {
                    installStrings[23] = "Class.forName(" + quoteMark +
"org.postgresql.Driver" + quoteMark + "); //Φορτώνει το PostgreSQL JDBC driver";
                } else {
                    installStrings[23] = "Class.forName(" + quoteMark + "org.hsql.jdbcDriver"
+ quoteMark + "); //Φορτώνει το Hypersonic SQL JDBC driver";
                }
                if (rt_dbtype==1) {
                    installStrings[24] = "new JDCConnectionDriver(" + quoteMark +
"org.gjt.mm.mysql.Driver" + quoteMark + "," + quoteMark + "jdbc:mysql://" + rt_dbhostname + "/" +
rt_dbname + quoteMark + ",username, password);";
                } else if (rt_dbtype==2) {
                    installStrings[24] = "new JDCConnectionDriver(" + quoteMark +
"org.postgresql.Driver" + quoteMark + "," + quoteMark + "jdbc:postgresql://" + rt_dbhostname + "/" +
rt_dbname + quoteMark + ",username, password);";
                } else {
                    installStrings[24] = "new JDCConnectionDriver(" + quoteMark +
"org.hsql.jdbcDriver" + quoteMark + "," + quoteMark + "jdbc:HypersonicSQL:IBBPMAServer" +
quoteMark + "," + quoteMark + "sa" + quoteMark + "," + quoteMark + quoteMark + ");";
                }
                /* IBBPMA.AuthKey strings Εγκατάστασης*/
                installStrings[25] = "private static final String skey1 = " + quoteMark + secret_key1
+ quoteMark + ";";
                installStrings[26] = "private static final String skey2 = " + quoteMark + secret_key2
+ quoteMark + ";";
                installStrings[27] = "private static final String MACpass = " + quoteMark +
MAC_key + quoteMark + ";";
                /* IBBPMA.Comms strings Εγκατάστασης*/
                installStrings[28] = "protected static final String T_address = " + quoteMark +
totaller_name + quoteMark + ";";
```

```
                installStrings[29] = "protected static final String ER_address = " + quoteMark +
er_name + quoteMark + ";";
                installStrings[30] = "protected static final int freePort = " + freePort + ";";
                installStrings[31] = "protected static final int freeRTPort = " + freeRTPort + ";";
                /* IBB/PMA.RTServer strings Εγκατάστασης*/
                installStrings[32] = "protected static final String er_public_key = " + quoteMark +
er_public_key + quoteMark + ";";
                installStrings[33] = "protected static final String fc_public_key = " + quoteMark +
fc_public_key + quoteMark + ";";
                installStrings[34] = "protected static final String rt_private_key = " + quoteMark +
rt_private_key + quoteMark + ";";
                installStrings[35] = "protected static final String rt_public_key = " + quoteMark +
rt_public_key + quoteMark + ";";
                installStrings[36] = "protected static final String vote_system = " + quoteMark +
edl_vote_system_attr[0] + "-" + edl_vote_system_attr[1] + quoteMark + ";";
                installStrings[37] = "protected static final String count_system = " + quoteMark +
edl_count_system_attr[0] + quoteMark + ";";
                installStrings[38] = "protected static final int number_of_choices = " +
edl_number_of_choices + ";";
                installStrings[39] = "protected static final String write_in = " + quoteMark +
edl_write_in + quoteMark + ";";
                /* IBBPMA.util.AuthSys strings Εγκατάστασης*/
                installStrings[40] = "private static final String phrase = " + quoteMark + passphrase +
quoteMark + ";";
                /* PollManager.Comms strings Εγκατάστασης*/
                installStrings[41] = "protected static final String ER_address = " + quoteMark +
er_name + quoteMark + ";";
                installStrings[42] = "protected static final int freePort = " + freePort + ";";
                installStrings[43] = "protected static final String er_public_key = " + quoteMark +
er_public_key + quoteMark + ";";
                installStrings[44] = "protected static final String pm_private_key = " + quoteMark +
pm_private_key + quoteMark + ";";
                /* IBBPMAClient.IBBPMAClient strings Εγκατάστασης*/
                installStrings[45] = "protected static final String vote_system = " + quoteMark +
edl_vote_system_attr[0] + "-" + edl_vote_system_attr[1] + quoteMark + ";";
                installStrings[46] = "protected static final int number_of_choices = " +
edl_number_of_choices + ";";
                installStrings[47] = "protected static final String write_in = " + quoteMark +
edl_write_in + quoteMark + ";";
                /* IBBPMAClient.VoteFrame strings Εγκατάστασης*/
                installStrings[48] = "private final String cand1name = " + quoteMark + edl_choice[0]
+ quoteMark + ";";
                installStrings[49] = "private final String cand1code = " + quoteMark +
edl_choice_attr[0][0] + quoteMark + ";";
                installStrings[50] = "private final String cand2name = " + quoteMark + edl_choice[1]
+ quoteMark + ";";
                installStrings[51] = "private final String cand2code = " + quoteMark +
edl_choice_attr[1][0] + quoteMark + ";";
                installStrings[52] = "private final String cand3name = " + quoteMark + edl_choice[2]
+ quoteMark + ";";
                installStrings[53] = "private final String cand3code = " + quoteMark +
edl_choice_attr[2][0] + quoteMark + ";";
                installStrings[54] = "private final String cand4name = " + quoteMark + edl_choice[3]
+ quoteMark + ";";
                installStrings[55] = "private final String cand4code = " + quoteMark +
edl_choice_attr[3][0] + quoteMark + ";";


                boolean t = false;
```

```
                t = processFile("IBBPMAClient/Comms.java");
                t = processFile("IBBPMAServer/IBBPMAServer.java");
                t = processFile("IBBPMAServer/AuthKey.java");
                t = processFile("IBBPMAServer/DBase.java");
                t = processFile("IBBPMA/DBase.java");
                t = processFile("IBBPMA/AuthKey.java");
                t = processFile("IBBPMA/Comms.java");
                t = processFile("IBBPMA/RTServer.java");
                t = processFile("IBBPMA/util/AuthSys.java");
                t = processFile("PollManager/Comms.java");
                t = processFile("IBBPMAClient/IBBPMAClient.java");
                t = processFile("IBBPMAClient/VoteFrame.java");

                NORM.info("INSTALLATION COMPLETE!");

                frame.jButton1.setVisible(true);

        } //EOF doInstallation()
```

```
        /**
         * Αυτός ο κώδικας βασίζεται στο <code>CodeSwitcher</code> από την διανομή
    * της Hypersonic SQL
         *
         * @param name Όνομα αρχείου που ελέγχεται για tags και τροποποιείται
         * @returns                Boolean Επιτυχίας ή αποτυχίας εκτέλεσης
         */

        private static boolean processFile(String name) {

                File f=new File(name);
                File fnew=new File(name+".new");
                int state=0; // 0=normal 1=inside_if 2=inside_else
                boolean working=false;
                try {
                        LineNumberReader read=new LineNumberReader(new FileReader(f));
                        FileWriter write=new FileWriter(fnew);
                        NORM.info("Opened " + name + " for editing.");
                        int i=1;
                        while(true) {
                                String line=read.readLine();
                                if(line==null) {
                                        break;
                                }
                                if(working) {
                                        if(line.equals("/*") || line.equals("*/")) {
                                                continue;
                                        }
                                }
                                if(!line.startsWith("//#")) {
                                        if (working) {
                                                write.write(installStrings[installCounter] +
"\r\n");
                                                DEV.info("Wrote: " +
installStrings[installCounter]);
                                                installCounter++;
                                                working = false;
                                                state = 0;
                                        } else {
```

```
                                        write.write(line+"\r\n");
                                    }
                                } else {  //otherwise we've hit a token
                                    int t=0;
                                    if(line.startsWith("//#ifdef ")) {
                                        if(state!=0) {
                                            printError("'#ifdef' not allowed inside
'#ifdef'");
                                            return false;
                                        }
                                        write.write(line+"\r\n");
                                        state=1;
                                        String s=line.substring(9);
                                        if(s.equals("INSTALL")) {
                                            working=true;
                                        }
                                    }
                                }
                            }

                            read.close();
                            write.flush();
                            write.close();
                            File fbak=new File(name+".bak");
                            fbak.delete();
                            f.renameTo(fbak);
                            File fcopy=new File(name);
                            fnew.renameTo(fcopy);
                            fbak.delete();
                            NORM.info(name + " edited successfully!");
                            return true;
                        } catch(Exception e) {
                            printError(e.getMessage());
                            return false;
                        }

                } //EOF processFile()


                static void printError(String e) {

                        NORM.error("Problem with: " + e);

                } //EOF printError()


        } //EOF Install
```

vii. KeyInstallFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Μια οθόνη που επιτρέπει στους χρήστες να ορίσουν τις μεταβλητές του IBB/PMA
 * με ευκολία.
 */

public class KeyInstallFrame extends Frame {

        Label jLabel1 = new Label();
        Label jLabel2 = new Label();
        Label jLabel3 = new Label();
        Label jLabel4 = new Label();
        Label jLabel6 = new Label();
        TextField jTextField1 = new TextField();
        TextField jTextField2 = new TextField();
        TextField jTextField3 = new TextField();
        TextField jTextField4 = new TextField();
        Button jButton1 = new Button();
        Label jLabel5 = new Label();

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}

        protected KeyInstallFrame() {

                jLabel4.setText("Please fill out fields for setup:");
                jLabel1.setText("Password of PMA #1:");
                jLabel2.setText("Password of PMA #2:");
                jLabel3.setText("Password of PMA #3:");
                jLabel5.setText("Please fill out ALL the fields!");
                jButton1.setLabel("OK");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("Key Installer");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel4);
                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel2);
                scrollPanel.addComponent( 4, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField2);
                scrollPanel.addComponent( 5, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel3);
                scrollPanel.addComponent( 6, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField3);
                scrollPanel.addComponent( 7, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel5);
                scrollPanel.addComponent( 7, 1, 1, 1, 0, 0, GridBagConstraints.NONE,
```

```
                1.0f, 0.0f, jButton1);

        add(scrollPanel);
        setTitle("IBB/PMA: Setup (4)");
        setResizable(true);
        setSize(new java.awt.Dimension(320, 380));

    }

    protected void initComponents() throws Exception
    {

        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                jButton1ActionPerformed(e);
            }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                thisWindowClosing(e);
            }
        });
    }

    void thisWindowClosing(java.awt.event.WindowEvent e)
    {
    }

    protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

        if
(jTextField1.getText().equals("")|jTextField2.getText().equals("")|jTextField3.getText().equals("")) {
            jLabel5.setVisible(true);
        } else {
            if
(Install.isSafe(jTextField1.getText())|Install.isSafe(jTextField2.getText())|Install.isSafe(jTextField3.get
Text())) {
                setVisible(false);
                setVisible(false);
                jLabel5.setVisible(false);

                Install.secret_key1 = jTextField1.getText();
                Install.secret_key2 = jTextField2.getText();
                Install.MAC_key = jTextField3.getText();

                Install.frame6.setVisible(true);
                dispose();
            } else {
                jLabel5.setVisible(true);
            }
        }

}

} //EOF class
```

viii. RTDBFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Μια οθόνη που επιτρέπει στους χρήστες να ορίσουν τις μεταβλητές του IBB/PMA
 * με ευκολία.
 */

public class RTDBFrame extends Frame {

        Label jLabel1 = new Label();
        Label jLabel2 = new Label();
        Label jLabel3 = new Label();
        Label jLabel4 = new Label();
        Label jLabel6 = new Label();
        Label jLabel7 = new Label();
        TextField jTextField1 = new TextField();
        TextField jTextField2 = new TextField();
        TextField jTextField3 = new TextField();
        TextField jTextField4 = new TextField();
        Button jButton1 = new Button();
        Label jLabel5 = new Label();

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}

        protected RTDBFrame() {

                jLabel4.setText("Please fill out fields for setup:");
                jLabel1.setText("IBB/PMA Database username:");
                jLabel2.setText("IBB/PMA Database password:");
                jLabel3.setText("IBB/PMA Database name:");
                jLabel7.setText("IBB/PMA Database server name:");
                jLabel5.setText("Please fill out ALL the fields!");
                jButton1.setLabel("OK");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("IBB/PMA Database Setup");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel4);
                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel2);
                scrollPanel.addComponent( 4, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField2);
                scrollPanel.addComponent( 5, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel3);
                scrollPanel.addComponent( 6, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField3);
                scrollPanel.addComponent( 7, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
```

```
                1.0f, 0.0f, jLabel7);
        scrollPanel.addComponent( 8, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jTextField4);
        scrollPanel.addComponent( 9, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jLabel5);
        scrollPanel.addComponent( 9, 1, 1, 1, 0, 0, GridBagConstraints.NONE,
                1.0f, 0.0f, jButton1);

        add(scrollPanel);
        setTitle("IBB/PMA: Setup (5b)");
        setResizable(true);
        setSize(new java.awt.Dimension(320, 380));

    }

    protected void initComponents() throws Exception
    {

        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                jButton1ActionPerformed(e);
            }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                thisWindowClosing(e);
            }
        });
    }

    void thisWindowClosing(java.awt.event.WindowEvent e)
    {
    }

    protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

        if
(jTextField1.getText().equals("")|jTextField2.getText().equals("")|jTextField3.getText().equals("")|jTe
xtField4.getText().equals("")) {
            jLabel5.setVisible(true);
        } else {
            if
(Install.isSafe(jTextField1.getText())|Install.isSafe(jTextField2.getText())|Install.isSafe(jTextField3.get
Text())|Install.isSafe(jTextField4.getText())) {
                setVisible(false);
                jLabel5.setVisible(false);

                Install.rt_dbuser = jTextField1.getText();
                Install.rt_dbpass = jTextField2.getText();
                Install.rt_dbname = jTextField3.getText();
                Install.rt_dbhostname = jTextField4.getText();

                setVisible(false);
                Install.frame.setVisible(true);
                Install.doInstallation();
                dispose();
            } else {
                jLabel5.setVisible(true);
            }
        }
```

```
}
} //EOF class
```

ix. ScreenAppender.java

```
package IBBPMAInstall;

import org.apache.log4j.*;
import org.apache.log4j.spi.*;

/**
 * Υλοποιεί την <code>com.ibm.log4j.Appender</code> διεπαφή για να παρέχει μια
 * έξοδο των δεδομένων monitoring .
 *
 */

public class ScreenAppender implements Appender {

	private String name;

	public void doAppend(LoggingEvent le) {

		if (le.priority.isGreaterOrEqual(Priority.WARN)) {
			Install.frame.showError("[" + le.priority.toString() + "] " + le.message);
		} else {  // else INFO or DEBUG
			Install.frame.showInfo("[" + le.priority.toString() + "] " + le.message);
		}

	} //EOF doAppend


	public void addFilter(Filter newFilter) {



	}


	public void clearFilters() {

	}


	public void close() {


	}


	public boolean requiresLayout()	{


		return false;

	}


	public void setErrorHandler(ErrorHandler errorHandler) {


	}
```

```
        public void setLayout(Layout l) {

        }

        public void activateOptions() {
        }

        public String[] getOptionStrings() {
                String [] s = new String[1];
                s[0] = name;
                return s;
        }

        public void setOption(String option, String value) {
                name = value;
        }

        public String getName() {
                return name;
        }

        public void setName(String n) {
                name = n;
        }
} //EOF ScreenAppender
```

x. StatusFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Η κύρια IBBPMAInstall οθόνη κατάστασης.
 */

public class StatusFrame extends Frame
{

    TextView tv;
        ScrollView sv;
        Font f;
        Font f2;
        TextStyle normal;
        TextStyle red;
        TextStyle blue;
        TextStyle green;
        Button jButton1 = new Button();

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}

        protected StatusFrame() {

                tv = new TextView(true);
    sv = new ScrollView(tv);
    f = new Font("Courier", Font.PLAIN, 12);
    f2 = new Font("Courier", Font.BOLD, 12);
    normal = new TextStyle(f, Color.black);
    red = new TextStyle(f2, Color.red);
    blue = new TextStyle(f, Color.blue);
    green = new TextStyle(f, Color.green);

                jButton1.setLabel("Quit");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("Installation in Progress");


                scrollPanel.setInsets(3, 3, 3, 3);
                add( scrollPanel);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, sv);

                scrollPanel.addComponent( 1, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton1);

                setTitle("IBB/PMA System: Installation progress");
                setSize(new java.awt.Dimension(500, 260));

        }
```

```
	public void initComponents() throws Exception
	{
		jButton1.addActionListener(new java.awt.event.ActionListener() {
			public void actionPerformed(java.awt.event.ActionEvent e) {
				jButton1ActionPerformed(e);
			}
		});
		addWindowListener(new java.awt.event.WindowAdapter() {
			public void windowClosing(java.awt.event.WindowEvent e) {
				thisWindowClosing(e);
			}
		});
	}

	// Εμφάνιση πληροφοριακού μηνύματος (black, plain text)
	protected void showInfo(String msg) {
		showMsg(msg);
	}

	// Εμφάνιση προειδοποιητικού μηνύματος (red, bold text)
	protected void showError(String msg) {
		msg += "\n";
		tv.append(msg, red);
	}

	// Εμφάνιση μηνύματος με ένα προκαθορισμένο AttributeSet
	protected void showMsg(String msg) {
		msg += "\n";
		tv.append(msg, normal);
	}

	void thisWindowClosing(java.awt.event.WindowEvent e)
	{
		setVisible(false);
		dispose();
		System.exit(0);
	}

	protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {
		setVisible(false);
		dispose();
		System.exit(0);
	}

	public static void main( String args[]){
		StatusFrame frame4 = new StatusFrame();
		frame4.setVisible(true);
		try{
			frame4.initComponents();
		}catch(Exception e){}
	}

}
```

xi. VoteInstallFrame.java

```
package IBBPMAInstall;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;


public class VoteInstallFrame extends Frame
{

        Label jLabel1 = new Label();
        Label jLabel2 = new Label();
        Label jLabel3 = new Label();
        Label jLabel4 = new Label();
        Label jLabel6 = new Label();
        TextField jTextField1 = new TextField();
        TextField jTextField2 = new TextField();
        TextField jTextField3 = new TextField();
        TextField jTextField4 = new TextField();
        Button jButton1 = new Button();
        Label jLabel5 = new Label();

        protected VoteInstallFrame() {

                jLabel4.setText("Please fill out fields for setup:");
                jLabel1.setText("Candidate #1:");
                jLabel2.setText("Candidate #2:");
                jLabel3.setText("Candidate #3:");
                jLabel6.setText("Candidate #4:");
                jLabel5.setText("Please fill out ALL the fields!");
                jButton1.setLabel("OK");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("Vote Choices");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel4);
                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel2);
                scrollPanel.addComponent( 4, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField2);
                scrollPanel.addComponent( 5, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel3);
                scrollPanel.addComponent( 6, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField3);
                scrollPanel.addComponent( 7, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel6);
                scrollPanel.addComponent( 8, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField4);
                scrollPanel.addComponent( 9, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel5);
                scrollPanel.addComponent( 9, 1, 1, 1, 0, 0, GridBagConstraints.NONE,
```

```
                1.0f, 0.0f, jButton1);

        add(scrollPanel);

        setLocation(new java.awt.Point(40, 40));
        setTitle("IBB/PMA: Setup (2)");
        setResizable(true);
        setSize(new java.awt.Dimension(300, 380));

    }

    protected void initComponents() throws Exception
    {

        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                jButton1ActionPerformed(e);
            }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                thisWindowClosing(e);
            }
        });
    }

    void thisWindowClosing(java.awt.event.WindowEvent e)
    {
    }

    protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

        if
(jTextField1.getText().equals("")|jTextField2.getText().equals("")|jTextField3.getText().equals("")|jTe
xtField4.getText().equals("")) {

            jLabel5.setVisible(true);

        } else {
            if
(Install.isSafe(jTextField1.getText())|Install.isSafe(jTextField2.getText())|Install.isSafe(jTextField3.get
Text())|Install.isSafe(jTextField4.getText())) {
                setVisible(false);
                jLabel5.setVisible(false);
                                    Install.party1 = jTextField1.getText();
                Install.party2 = jTextField2.getText();
                Install.party3 = jTextField3.getText();
                Install.party4 = jTextField4.getText();

                Install.frame4.setVisible(true);
                dispose();
            } else {
                jLabel5.setVisible(true);
            }

        }

    }

} //EOF class
```

Β. Πακέτο Εξυπηρετητή Πιστοποίησης IBB/PMA

i. Authkey.java

```
package IBBPMAServer;

import cryptix.provider.key.RawSecretKey;
import cryptix.provider.padding.OneAndZeroes;
import cryptix.provider.cipher.Blowfish;
import cryptix.util.core.Hex;

import xjava.security.Cipher;
import xjava.security.FeedbackCipher;
import xjava.security.SecretKey;

import java.security.*;

import IBBPMA.util.StringByteTools;

/**
 * Το AuthKey υπολογίζει το κλειδί πιστοποίησης IBB/PMA το οποίο
 * έχει σκοπό την προστασία του προγράμματος από επιθέσεις τύπου
 * reverse engineering στο πρόγραμμα πελάτη. Επιπλέον έχει σκοπό να
 * μειώσει τον αριθμό των πελατών που μπορεί να μην ψηφίσουν εξαιτίας
 * προβλήματος στον υπολογιστή τους.
 *
 * Η κρυπτογράφηση γίνεται με ένα συμμετρικό αλγόριθμο κρυπτογράφησης
 * από τον οργανισμό <a href="http://www.cryptix.org">The Cryptix
 * Foundation</a>.
 *
*/

public class AuthKey {

        /**
         * Σημαντική σημείωση ασφάλειας:
         * Οι τρεις ακόλουθες μεταβλητές πρέπει να αλλάζουν σε κάθε
    * ψηφοφορία.
         */

        // Εδώ τοποθετούνται τα κλειδιά για το AuthKey σύστημα

//#ifdef INSTALL
        private static final String skey1 = "F1E1D2C2B4A5968778A95A4B3C221E0F003122";

//#ifdef INSTALL
        private static final String skey2 = "FB4A210D1D2C7840E1D2C3B4A5968778695A4B";

        // Αποθηκεύει το MAC τμήμα του κλειδιού

//#ifdef INSTALL
        private static final String MACpass = "p0rpleturn1p";

        /**
         * Αρχικοποιεί την κωδικοποίηση
         */
```

```
protected static void init() {

        try {
                java.security.Security.addProvider(new cryptix.provider.Cryptix());
        } catch(Exception e) {
                IBBPMAServer.NORM.error("AuthSys initialistion error: " + e.toString());
        }

} //eof init()


/**
* Κατασκευάζει το <code>AuthKey</code> από τα δεδομένα που
* έχουν τοποθετηθεί. Η ακόλουθη διαδικασία δημιουργεί το
* IBBPMA AuthKey:
* Το αποτέλεσμα της <code>Roll</code> εισαγωγής και το
* <code>MACpass</code> δημιουργείται.
* Τότε δημιουργείται το digest του μηνύματος, με το
* <code>MACpass</code>
* και ο κωδικός δημιουργείται.
* Το αποτέλεσμα τότε κρυπτογραφείται με ένα από τα δύο μυστικά
* κλειδιά όπως ορίζεται από το <code>key</code>.
*
* @param name Το όνομα χρήστη από την <code>Roll</code> βάση
* δεδομένων.
* @param code Ένα μοναδικό pin code για τον χρήστη, επιπλέον
* και από τη βάση δεδομένων
* @param pword Ένας κωδικός για τον χρήστη επίσης από τη βάση
* δεδομένων
* @param whichKey Ορίζει εάν θα χρησιμοποιηθεί το μυστικό
* κλειδί 1 ή 2
* @returns String με ένα οριστικό AuthKey
*/

protected static String build(String name, String code, String pword, int whichKey) throws
Exception {

        RawSecretKey key;
        String digest;
        byte[] ect;

        Cipher blowalg = Cipher.getInstance(new Blowfish(),null,new OneAndZeroes());

        MessageDigest md = MessageDigest.getInstance("SHA-1");

        byte dataBytes[] = StringByteTools.asciiGetBytes(name + code + pword);
        byte passBytes[] = StringByteTools.asciiGetBytes(MACpass);
        md.update(passBytes);
        md.update(dataBytes);
        byte digest1[] = md.digest();

        md.update(passBytes);
        md.update(digest1);
        byte mac[] = md.digest();

/* Κατασκευάζει ένα Blowfish (16-round) κλειδί με ECB */

        if (whichKey==1) {
                key = new RawSecretKey("Blowfish", Hex.fromString(skey1));
        } else {
```

```
                    key = new RawSecretKey("Blowfish", Hex.fromString(skey2));
        }

                blowalg.initEncrypt(key);
                ect = blowalg.crypt(mac);

                return Hex.toString(ect);

        } //eof build()


        /**
        * Αποκρυπτογραφεί ένα κλειδί, αλλά δεν αποκαλύπτει την αρχική
    * προέλευση αφού το
    * αποκρυπτογραφημένο κλειδί είναι απλά το MAC
        * και όχι απλό κείμενοο.
        *
        * @param akey  Το κλειδί για αποκρυπτογράφηση
        * @param whichKey  Δείχνει εάν μπορεί να χρησιμοποιηθεί το
    * μυστικό κλειδί 1 ή 2
        * @returns String με ένα αποκρυπτογραφημένο AuthKey
        */

        protected static String decrypt(String akey, int whichKey) throws Exception {

                RawSecretKey key;
                byte[] dect;
                Cipher blowalg = Cipher.getInstance(new Blowfish(),null,new OneAndZeroes());

/* build a Blowfish (16-round) key using ECB */

        if (whichKey==1) {
                        key = new RawSecretKey("Blowfish", Hex.fromString(skey1));
                } else {
                        key = new RawSecretKey("Blowfish", Hex.fromString(skey2));
        }

                blowalg.initDecrypt(key);
                dect = blowalg.crypt(Hex.fromString(akey));

                return Hex.toString(dect);

        } //eof decrypt()


        /**
        * Κρυπτογραφεί ένα κλειδί - για χρήση κατά την εξαγωγή των
    * κλειδιών
        *
        * @param akey  Το κλειδί για την κρυπτογράφηση
        * @param whichKey  Δείχνει εάν θα χρησιμοποιηθεί το μυστικό
    * κλειδί 1 ή 2
        * @returns String με ένα κρυπτογραφημένο AuthKey
        */

        protected static String encrypt(String akey, int whichKey) throws Exception {

                RawSecretKey key;
                byte[] ect;
                Cipher blowalg = Cipher.getInstance(new Blowfish(),null,new OneAndZeroes());
```

```
                if (whichKey==1) {
                        key = new RawSecretKey("Blowfish", Hex.fromString(skey1));
                } else {
                        key = new RawSecretKey("Blowfish", Hex.fromString(skey2));
        }

                blowalg.initEncrypt(key);
                ect = blowalg.crypt(Hex.fromString(akey));

                return Hex.toString(ect);

        } //eof encrypt()


} //eof AuthKey
```

ii. CryptFrame.java

```
package IBBPMAServer;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;
import IBBPMA.util.*;

/**
 * Ζητάει από τον χρήστη να εισάγει ένα κλειδί για την Βάση Δεδομένων
 *
 * @version 0.1 20 February 2001
 * @author Jason Kitcat, Rajagopal C.V
 */

public class CryptFrame extends Frame {

        Label jLabel1 = new Label();
        TextField jTextField1 = new TextField();
        Button jButton1 = new Button();

        public Insets getInsets() {

                return new Insets( 30, 10, 10, 10);
        }

        protected CryptFrame() {

                setTitle("IBB/PMA Ciphering Key: Password");
                jButton1.setLabel("OK");
                jTextField1.setEchoChar('*');
                jLabel1.setText("Please enter you 4 Digit PMA Key for the encryption of the
databases");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("IBB/PMA Ciphering Key:
Encryption Password");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton1);

                add(scrollPanel, BorderLayout.CENTER);
                setSize(new java.awt.Dimension(400, 175));

        }

        public void initComponents() throws Exception {

                jButton1.addActionListener(new java.awt.event.ActionListener() {
                        public void actionPerformed(java.awt.event.ActionEvent e) {
                                jButton1ActionPerformed(e);
                        }
                });
                addWindowListener(new java.awt.event.WindowAdapter() {
```

```
                public void windowClosing(java.awt.event.WindowEvent e) {
                    thisWindowClosing(e);
                }
            });
        }

        // Κλείνει το παράθυρο όταν επιλεγεί το close box

        void thisWindowClosing(java.awt.event.WindowEvent e)
        {
        }

        public void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

            String input = new String(jTextField1.getText());

            if (!input.equals("")) {
                try {
                    if (DBase.isSafe(input)) {
                        DBase.cryptword=AuthSys.makeDigest(input);

                        setVisible(false);
                        IBBPMAServer.frame.setVisible(true);
                        IBBPMAServer.startDaemon();

                        dispose();
                    }

                } catch (Exception ex) { // bail out as it's too early to deal with
                    setVisible(false);
                    dispose();
                    System.exit(1);
                }
            }

        }

} //EOF Class
```

iii. Dbase.java

```
package IBBPMAServer;

import java.io.*;
import java.sql.*;
import java.util.*;
//#ifdef INSTALL
import org.hsql.*;

import cryptix.provider.key.RawSecretKey;
import cryptix.provider.padding.OneAndZeroes;
import cryptix.provider.cipher.Blowfish;
import cryptix.util.core.Hex;

import xjava.security.Cipher;
import xjava.security.FeedbackCipher;
import xjava.security.SecretKey;

import java.security.*;

import IBBPMA.util.*;
import IBBPMA.DBPool.*;

/**
 * Το DBase δημιουργεί τις βάσεις δεδομένων, τους πίνακες και
 * αλληλεπιδρά
 * με αυτές μέσω JDBC και SQL.
 *
 * Η υλοποίηση των βάσεων έχει πραγματοποιηθεί με
 * <a href="http://hsql.oron.ch">Hypersonic SQL</a> η οποία είναι εξ'
 * ολοκλήρου γραμμένη σε Java.
 * @version 1.8  28 August 2001
 * @author Jason Kitcat
 */

public class DBase {

//#ifdef INSTALL
        private static final String username = "sa";

//#ifdef INSTALL
        private static final String password = "";

        // USER SUPPLIED SESSION WORD
        protected static String cryptword = "";


        /**
         * Αρχικοποιεί την Βάση Hypersonic SQL.
         *
         */

        protected static void init() {

                try {

//#ifdef INSTALL
                        Class.forName("org.hsql.jdbcDriver"); // Φορτώνει το Hypersonic SQL
JDBC driver
```

```
                        /* Ετοιμάζει το pool με 10 συνδέσεις */
//#ifdef INSTALL
                        new
JDCConnectionDriver("org.hsql.jdbcDriver","jdbc:HypersonicSQL:IBBPMAServer",username,
password);

                        /* Get a connection from the pool */
                        Connection FreeConn = getConnection();
                        Statement freeStat = freeConn.createStatement();

                        /* Δημιουργία του πίνακα Electoral Roll */
                        try {
                                ResultSet r = freeStat.executeQuery("SELECT name FROM
Roll"); //Ελεγχος για τον πίνακα
                        } catch (SQLException sqle) {
                                if(IBBPMAServer.DEV.isDebugEnabled()) {
                                        IBBPMAServer.DEV.debug("Roll table didn't exist so
creating it");
                                }
                                freeStat.execute("CREATE TABLE Roll_1(name
VARCHAR(255), code VARCHAR(255), pword VARCHAR(255), voted CHAR(1), vote_where
CHAR(1), info_sent CHAR(1), other VARCHAR(255))");
                        }

                        DatabaseMetaData dbMetaData = freeConn.getMetaData();

                        freeStat.close();

                        IBBPMAServer.NORM.info("Database intialised");
                        IBBPMAServer.DEV.debug("DB Using: " +
dbMetaData.getDatabaseProductName() + " " + dbMetaData.getDatabaseProductVersion());

                } catch(Exception e) {
                        IBBPMAServer.NORM.error("Database error: " + e.toString());
                }

        } //EOF init()


        /**
         * Παρέχει μια σύνδεση με την βάση δεδομένων από το pool.
         *
         * @returns  Μια ενεργή σύνδεση
         */

        private static Connection getConnection() throws SQLException {
    return DriverManager.getConnection("jdbc:jdc:jdcpool");
  }


        /**
         * Ελέγχει την String είσοδο για να βεβαιωθεί ότι περιέχει
    * ασφαλείς χαρακτήρες.
         *
         * @param input  To String που θα ελεγχθεί
         * @returns  True εάν το String είναι OK
         */

        protected static boolean isSafe(String input) throws Exception{
```

```
boolean ok = true;

for(int i = 0; (i<input.length())&&(ok==true); i++) {

        switch (input.charAt(i)) {
                case 'a': ok = true; break;
                case 'b': ok = true; break;
                case 'c': ok = true; break;
                case 'd': ok = true; break;
                case 'e': ok = true; break;
                case 'f': ok = true; break;
                case 'g': ok = true; break;
                case 'h': ok = true; break;
                case 'i': ok = true; break;
                case 'j': ok = true; break;
                case 'k': ok = true; break;
                case 'l': ok = true; break;
                case 'm': ok = true; break;
                case 'n': ok = true; break;
                case 'o': ok = true; break;
                case 'p': ok = true; break;
                case 'q': ok = true; break;
                case 'r': ok = true; break;
                case 's': ok = true; break;
                case 't': ok = true; break;
                case 'u': ok = true; break;
                case 'v': ok = true; break;
                case 'w': ok = true; break;
                case 'x': ok = true; break;
                case 'y': ok = true; break;
                case 'z': ok = true; break;
                case 'A': ok = true; break;
                case 'B': ok = true; break;
                case 'C': ok = true; break;
                case 'D': ok = true; break;
                case 'E': ok = true; break;
                case 'F': ok = true; break;
                case 'G': ok = true; break;
                case 'H': ok = true; break;
                case 'I': ok = true; break;
                case 'J': ok = true; break;
                case 'K': ok = true; break;
                case 'L': ok = true; break;
                case 'M': ok = true; break;
                case 'N': ok = true; break;
                case 'O': ok = true; break;
                case 'P': ok = true; break;
                case 'Q': ok = true; break;
                case 'R': ok = true; break;
                case 'S': ok = true; break;
                case 'T': ok = true; break;
                case 'U': ok = true; break;
                case 'V': ok = true; break;
                case 'W': ok = true; break;
                case 'X': ok = true; break;
                case 'Y': ok = true; break;
                case 'Z': ok = true; break;
                case '1': ok = true; break;
                case '2': ok = true; break;
```

```
                                case '3': ok = true; break;
                                case '4': ok = true; break;
                                case '5': ok = true; break;
                                case '6': ok = true; break;
                                case '7': ok = true; break;
                                case '8': ok = true; break;
                                case '9': ok = true; break;
                                case '0': ok = true; break;
                                case ';': ok = true; break;
                                case ')': ok = true; break;
                                case '(': ok = true; break;
                                case '*': ok = true; break;
                                case '\': ok = true; break;
                                case ' ': ok = true; break;
                                case '=': ok = true; break;
                                case '-': ok = true; break;
                                case '_': ok = true; break;
                                case '+': ok = true; break;
                                case '@': ok = true; break;
                                case ',': ok = true; break;
                                case '.': ok = true; break;
                                case '<': ok = true; break;
                                case '>': ok = true; break;
                                case '?': ok = true; break;
                                default:  ok = false; break;
                        } //eof case

                } //eof for

                return ok;

        } //eof isSafe


        /**
         * Κρυπτογραφεί τα αλφαριθμητικά δεδομένα προτού τα εισάγει
    * στην βάση δεδομένων
         *
         * @param data  Το string που θα κρυπτογραφηθεί
         * @returns  Κρυπτογραφημένο string
         */

        private static String encrypt(String data) throws Exception {

                RawSecretKey key;
                byte[] ect;
                Cipher blowalg = Cipher.getInstance(new Blowfish(),null,new OneAndZeroes());

                /* build a Blowfish (16-round) key using ECB */
                key = new RawSecretKey("Blowfish", Hex.fromString(cryptword));

                blowalg.initEncrypt(key);
                ect = blowalg.crypt(StringByteTools.asciiGetBytes(data));

                return Hex.toString(ect);

        } //eof encrypt()


        /**
```

```
        * Αποκρυτογραφεί τα δεδομένα που βγαίνουν από τη βάση και
    * επιστρέφει κανονικά αλφαριθμητικά.
        *
        * @param data  Το κρυπτογραφημένο string
        * @returns  Αποκρυπτογραφημένο string
        */

        protected static String decrypt(String data) throws Exception {

                RawSecretKey key;
                byte[] dect;
                Cipher blowalg = Cipher.getInstance(new Blowfish(),null,new OneAndZeroes());

                /* Φτιάχνει ένα Blowfish (16-round) */
                key = new RawSecretKey("Blowfish", Hex.fromString(cryptword));

                blowalg.initDecrypt(key);
                dect = blowalg.crypt(Hex.fromString(data));

                return new String(dect);

        } //eof decrypt()


        /**
         * Μετράει πόσοι άνθρωποι έχουν ψηφίσει.
         *
         * @returns  Ένας integer των ανθρώπων που έχουν ψηφίσει
         */
        protected static int usersVoted() throws Exception {

                ResultSet result;
                boolean res;
                int count;

                /* Παίρνει μια σύνδεση από το pool */
                Connection freeConn = getConnection();
                Statement freeStat = freeConn.createStatement();

                /* Ψάχνει για τα δεδομένα*/
                res = freeStat.execute("SELECT COUNT(voted) FROM Roll WHERE voted='" +
encrypt("T") + "'");
                result = freeStat.getResultSet();

                /*έλεγχος αποτελέσματος*/
                res = result.next();
                if (!res) {
                        count = 0;
                } else {
                        count = new Integer(result.getString(1)).intValue();
                }

                freeStat.close();

                return count;

        } //EOF usersVoted()


        /**
```

```
    * ελέγχει τις πληροφορίες στο Electoral Roll σε αντιπαράθεση
    * με την βάση δεδομένων για το IBBPMAClient
         *
         * @param name  String αναπαραστεί το όνομα του ψηφοφόρου.
         * @param code  String Αναπαριστά τον προσωπικό κωδικό του
    *          ψηφοφόρου
         * @param pword String Αναπαριστά το password του ψηφοφόρου
         * @returns    Μια boolean, true Υπάρχει μια αντίστοιχη εγγραφή
         */
        protected static boolean checkER(String name, String code, String pword) throws Exception {

                ResultSet result;
                boolean temp, res;

                /* boundary check τα δεδομένα αρχικά*/
                // μόνο εάν τα δεδομένα είναι αλφαριθμητικά
                if (isSafe(name)&&isSafe(code)&&isSafe(pword)) {

                        /* παίρνουμμια μια σύνδεση από το pool */
                        Connection freeConn = getConnection();
                        Statement freeStat = freeConn.createStatement();

                        /* ψάχνουμε για τα δεδομένα*/
                        res = freeStat.execute("SELECT voted, vote_where FROM Roll WHERE
name = '" + encrypt(name) + "' AND code = '" + encrypt(code) + "' AND pword = '" + encrypt(pword)
+ "'");
                        result = freeStat.getResultSet();

                        /* έλεγχος απάντησης*/
                        res = result.next();
                        if (!res) {
                                temp = false;
                        } else {
                                if (decrypt(result.getString(2)).equals("I")) { // ο χρήστης
πιστοποιείται για την διαδικτυακή ψηφοφορία
                                        if (decrypt(result.getString(1)).equals("F")) {  // Δεν έχει
ψηφίσει ακόμα ο χρήστης
                                                temp = true;
                                                freeStat.execute("UPDATE Roll SET voted = '"
+ encrypt("C") + "' WHERE name = '" + encrypt(name) + "' AND code = '" + encrypt(code) + "' AND
pword = '" + encrypt(pword) + "'");
                                        } else if (decrypt(result.getString(1)).equals("C")) {  // Η
ψήφος τους δεν έχει καταγραφεί αλλά έχουν ξαναπροσπαθήσει να συνδεθούν
                                                temp = true;
                                                IBBPMAServer.NORM.info("DBase.checkER():
User already tried to vote but not confirmed");
                                        } else {  // αλλίως έχουν ήδη ψηφίσει
                                                temp = false;
                                        }
                                } else {
                                        IBBPMAServer.NORM.info("DBase.checkER(): User not
authorised to vote online.");
                                        temp = false;
                                }
                        }

                        freeStat.close();

                } else {
```

```
                temp = false;
                throw new Exception("DBase.checkER() - invalid character(s) used");

            }

            return temp;

        } //EOF checkER()


        /**
     * ελέγχει τις πληροφορίες στο Electoral Roll σε αντιπαράθεση
     * με την βάση δεδομένων για το IBBPMAClient
         *
         * @param name  String αναπαραστεί το όνομα του ψηφοφόρου.
         * @param code  String Αναπαριστά τον προσωπικό κωδικό του
     *          ψηφοφόρου
         * @param pword String Αναπαριστά το password του ψηφοφόρου
         * @returns Έναν πίνακα από String που επιστρέφει: voted
     *          status και vote_where
         */

        protected static String[] PMcheckER(String name, String code, String pword) throws
Exception {

            ResultSet result;
            boolean temp, res;
            String[] results = new String[] {"", ""};

            if (isSafe(name)&&isSafe(code)&&isSafe(pword)) {

                /* Κάνει μια σύνδεση από το pool*/
                Connection freeConn = getConnection();
                Statement freeStat = freeConn.createStatement();

                /* έλεγχος για τα δεδομένα*/
                res = freeStat.execute("SELECT voted, vote_where FROM Roll WHERE
name = '" + encrypt(name) + "' AND code = '" + encrypt(code) + "' AND pword = '" + encrypt(pword)
+ "'");
                result = freeStat.getResultSet();

                /* έλεγχος απάντησης*/
                res = result.next();
                if (!res) {
                    temp = false;
                } else {
                    if (decrypt(result.getString(1)).equals("F")) {  // Δέν έχουν ψηφίσει
ακόμα
                        freeStat.execute("UPDATE Roll SET voted = '" +
encrypt("C") + "' WHERE name = '" + encrypt(name) + "' AND code = '" + encrypt(code) + "' AND
pword = '" + encrypt(pword) + "'");
                    } else if (decrypt(result.getString(1)).equals("C")) {  // Η ψήφος
δεν έχει καταμετρηθεί ακόμα αλλά έχει ξανασυνδεθεί
                        IBBPMAServer.NORM.info("DBase.PMcheckER(): User
already tried to vote but not confirmed");
                    } else {  // Διαφορετικά έχουν ήδη ψηφίσει
                    }
                    results[0] = decrypt(result.getString(1));
                    results[1] = decrypt(result.getString(2));
                }
```

```
                freeStat.close();

            } else {

                results[0] = "ERROR";
                results[1] = "ERROR";
                throw new Exception("DBase.PMcheckER() - invalid character(s) used");

            }

            return results;

        } //EOF PMcheckER()


        /**
         * Αυτή η μέθοδος ψάχνει μια είσοδο στο Electoral Roll που να
         * ταιριάζει στο κλειδί που
         * έλαβε και ορίζει ότι ψήφισε
         *
         * @param akey String που αναπαραστεί το AuthKey του ψηφοφόρου
         * @returns   Μια boolean, true οι λεπτομέρειες ταιριάζουν με
     * τις εγγραφές
         */

        protected static boolean confirmVoted(String akey) throws Exception {

            ResultSet results;
            int count;
            boolean carryon = true;
            boolean res, retVal;

            /* Παίρνει μια σύνδεση από το pool*/
            Connection freeConn = getConnection();
            Statement freeStat = freeConn.createStatement();

            /* Ψάχνει για τα δεδομένα*/
            res = freeStat.execute("SELECT * FROM Roll WHERE voted = '" + encrypt("C") +
"'");

            count = freeStat.getUpdateCount();
            results = freeStat.getResultSet();
            if (!res) {  // Εάν δεν βρεθούν εγγραφές
                carryon=true;
            } else {  // αλλιώς υπολογίσε και βρές
                while(results.next()&&carryon) {
                    if
(akey.equals(AuthKey.build(decrypt(results.getString(1)),decrypt(results.getString(2)),decrypt(results.
getString(3)),1))) {
                        // Ανανεώνει τη βάση δεδομένων και εισάγει ότι ο
χρήστης ψήφισε
                        freeStat.execute("UPDATE Roll SET voted = '" +
encrypt("T") + "' WHERE name = '" + results.getString(1) + "' AND code = '" + results.getString(2) +
"' AND pword = '" + results.getString(3) + "'");
                        carryon = false;
                    }
                }
            }
```

```
                // Εάν δεν βρεθεί κάποια εγγραφή κάποιο λάθος έγινε
                if (carryon) {
                        retVal=false;
                } else {
                        retVal=true;
                }

                freeStat.close();

                return retVal;

        } //EOF confirmVoted()


        /**
         * Χρησιμοποιώντας δεδομένα από τον PollManager δείχνει έναν
    * ψηφοφόρο σαν να έχει ψηφίσει
         *
         * @param name. String αναπαριστά το όνομα του ψηφοφόρου.
         * @param code  String Αναπαριστά τον προσωπικό κωδικό του
    * ψηφοφόρου
         * @param pword String Αναπαριστά το password του ψηφοφόρου
         * @returns   Μια boolean, true εάν οι λεπτομέρειες ταιριάζουν
    * με μια εγγραφή στη βάση
         */

        protected static void PMconfirmVoted(String name, String code, String pword) throws
Exception {

                /* Παίρνει μια σύνδεση από το pool */
                Connection freeConn = getConnection();
                Statement freeStat = freeConn.createStatement();

                freeStat.execute("UPDATE Roll SET voted = '" + encrypt("T") + "' WHERE name =
'" + encrypt(name) + "' AND code = '" + encrypt(code) + "' AND pword = '" + encrypt(pword) + "'");

                freeStat.close();

        } //EOF PMconfirmVoted()


        /**
         * Περνάει όλα τα Electoral Roll δεδομένα για να δημιουργήσει
    * κλειδιά που
    * φτιάχνονται με το κλειδί 1 και τα γράφει σε ένα αρχείο για
    * έξοδο
         *
         */

        protected static void makeAllKeys() throws Exception {

                IBBPMAServer.NORM.info("Building all keys...");

                File keyFile = new File("rtserver.keys");
                File testFile = new File("rtserver.test.keys"); //για τις δοκιμαστικές ψήφους
                FileWriter out = new FileWriter(keyFile);
                FileWriter t_out = new FileWriter(testFile);
                ResultSet results;
                int count;
                boolean res, res2;
```

```
		boolean test=false;
		String temp;

		/* Παίρνει μια σύνδεση από το pool*/
		Connection freeConn = getConnection();
		Statement freeStat = freeConn.createStatement();

		/* Έλεγχος για τα δεδομένα*/
		// only select data which has been through stage one authentication
		res = freeStat.execute("SELECT * FROM Roll");

		/* Κατασκευή κάθε κλειδιού*/
		count = freeStat.getUpdateCount();
		results = freeStat.getResultSet();
		if (!res) {  // Εάν δεν βρεθούν δεδομένα για να γίνει το κλειδί
			throw new Exception("No data to make keys with");
		} else {  // Αλλιώς θα δουλεύει
			while(results.next()) {
				// build + encrypt output
				if (decrypt(results.getString(4)).equals("B")) {
					/* test ballot */
					temp =
AuthKey.build(decrypt(results.getString(1)),decrypt(results.getString(2)),decrypt(results.getString(3)),
1);
					temp = AuthKey.encrypt(temp,1);
					t_out.write(temp + "\r\n");
					// Διαγραφή όλων των δεδομένων που δείχνει ότι ήταν μια
δοκιμαστική ψήφος
					Connection freeConn2 = getConnection();
					Statement freeStat2 = freeConn2.createStatement();
					res2 = freeStat2.execute("UPDATE Roll SET voted = '" +
encrypt("F") + "' WHERE name = '" + results.getString(1) + "' AND code = '" + results.getString(2) +
"' AND pword = '" + results.getString(3) + "'");
					freeStat2.close();
					test=true;
				} else {
					/* Κανονικός χρήστης*/
					temp =
AuthKey.build(decrypt(results.getString(1)),decrypt(results.getString(2)),decrypt(results.getString(3)),
1);
					temp = AuthKey.encrypt(temp,1);
					out.write(temp + "\r\n");
				}
			}
		}

		out.close();
		t_out.close();
		freeStat.close();

		IBBPMAServer.NORM.info("Done! Wrote rtserver.keys for export.");
		if (test) {
			IBBPMAServer.NORM.info("Wrote rtserver.test.keys for export.");
		}

	} //EOF makeAllKeys()


	/**
	 * Εισάγει τους Νόμιμους ψηφοφόρους από ένα αρχείο CVS και τους
```

```
* εισάγει στην IBBPMAServer βάση.
     *
     */

     protected static void importUsers() throws Exception {
          int normal=0;
          int test=0;
          int line=0;
          ResultSet result;
          boolean res;
          String inData = "";

          IBBPMAServer.NORM.info("Importing users from IBBPMAServer.users");

          BufferedReader in = new BufferedReader(new FileReader("IBBPMAServer.users"));

          inData = in.readLine();

          while (inData!=null) {
               line++;
               /* Έλεγχος αρχικά των δεδομένων */
               // συνέχεια μόνο εάν τα δεδομένα είναι αλφαριθμητικά, κενά ή σημεία
στίξης
               if (isSafe(inData)) {
                    // if a blank line then skip
                    if ((inData.equals(""))||(inData.equals(" "))||(inData.equals(null))) {
                         // do nothing
                         IBBPMAServer.NORM.info("importUsers - blank line
skipped on line " + line);
                    } else {

                         // Πέρασμα για την λήψη διαφορετικών πεδίων
δεδομένων
                         int i = 0;
                         String[] pp = new String[] {"", "", "", "", "", "", "", "", "",
""};

                         StringTokenizer splitter = new StringTokenizer(inData,
",", false);

                         while (splitter.hasMoreTokens()) {
                              pp[i] = splitter.nextToken();
                              i++;
                         }

                         String name = pp[0];
                         String code = pp[1];
                         String pword = pp[2];
                         String status = pp[3];
                         String vote_where = pp[4];

                         // έλεγχος εγκυρότητας
                         switch(status.charAt(0)) {
                              case 'F': break; // normal voter
                              case 'B': break; // test ballot
                              default : throw new Exception("importUsers -
bad voter status in record, only F or B allowed. On line " + line);
                         }

                         switch(vote_where.charAt(0)) {
```

```
                                        case 'I': break;  // internt
                                        case 'M': break;  // mail (postal)
                                        case 'P': break;  // polling station
                                        case 'O': break;  // other
                                        default : throw new Exception("importUsers -
bad vote_where in record, only I,M,P or O allowed. On line " + line);
                                    }

                                    /* Έλεγχος της βάσης δεδομένων για διπλότυπα */

                                    // Λήψη σύνδεσης από το pool
                                    Connection freeConn = getConnection();
                                    Statement freeStat = freeConn.createStatement();
                                    res = freeStat.execute("SELECT voted FROM Roll
WHERE name = '" + encrypt(name) + "' AND code = '" + encrypt(AuthSys.makeDigest(code)) + "'
AND pword = '" + encrypt(AuthSys.makeDigest(pword)) + "'");
                                    result = freeStat.getResultSet();

                                    // Έλεγχος απάντησης
                                    res = result.next();
                                    if (!res) {
                                        doSQL("INSERT INTO Roll VALUES ('" +
encrypt(name) + "', '" + encrypt(AuthSys.makeDigest(code)) + "', '" +
encrypt(AuthSys.makeDigest(pword)) + "', '" + encrypt(status) + "', '" + encrypt(vote_where) +
"','');");
                                    } else {
                                        // Έχουμε μια διπλοεγγραφή οπότε
ενημερώνουμε και την απορρίπτουμε
                                        IBBPMAServer.NORM.info("importUsers -
record on line " + line + " already exists in DB.");
                                    }

                                    freeStat.close();


                                    if (status.equals("B")) {
                                        // Test Ballot
                                        test++;
                                    } else {
                                        // Normal user
                                        normal++;
                                    }
                                }
                                    inData = in.readLine();
                            } else {
                                throw new Exception("importUsers - invalid character(s) used on
line " + line);
                            }
                        }

                        float proportions = new Float(test).floatValue() / (new Float(test).floatValue() + new
Float(normal).floatValue());
                        if (proportions>0.29) {
                            IBBPMAServer.NORM.warn("Proportion of Test Ballots is rather high: " +
new Float(proportions * 100).toString() + "%");
                        }
                        IBBPMAServer.NORM.info("Import complete.");

                } //eof importUsers
```

```
        /**
         * Εκτελεί την SQL εντολή, που έχει στείλει ώς παράμετρο και
    * επιστρέφει το αποτέλεσμα στην DB Κονσόλα.
         *
         * @param  sqlCommand Ένα string που περιέχει SQL
         */

        protected static void doSQL(String sqlCommand) {

                ResultSet results;
                boolean res;
                int count;

                try {
                        /* Είναι custom εντολή? */
                        if (sqlCommand.charAt(0) == '~') {
                                // parse into SQL
                                String data, name, code, pword;
                                int i=0;
                                int t=0;

                                data = sqlCommand.substring(2);

                                // Πέρασμα για να βρεί ξεχωριστά πεδία εγγραφών
                                while (data.charAt(i)!='-') {
                                        i++;
                                }
                                name = data.substring(0,i);

                                t = i+2;

                                while (data.charAt(t)!='-') {
                                        t++;
                                }

                                code = data.substring((i+1),t);

                                pword = data.substring(t+1);

                                sqlCommand = "INSERT INTO Roll VALUES ('" +
encrypt(name) + "', '" + encrypt(AuthSys.makeDigest(code)) + "', '" +
encrypt(AuthSys.makeDigest(pword)) + "', '" + encrypt("F") + "', '" + encrypt("I") + "','');";
                        }

                        /* Έλεγχος ορίων για τα δεδομένα στην αρχή */
                        // Μόνο εάν τα δεδομένα είναι αλφαριθμητικά, συνεχίζουμε
                        if (isSafe(sqlCommand)) {

                                IBBPMAServer.DEV.info("Executing: " + sqlCommand);

                                /* Πέρνουμε μια σύνδεση από το pool*/
                                Connection freeConn = getConnection();
                                Statement freeStat = freeConn.createStatement();

                                /* Εκτέλεσε το SQL */
                                res = freeStat.execute(sqlCommand);

                                /* Εμφάνισε τα αποτελέσματα*/
                                count = freeStat.getUpdateCount();
```

```
                                        results = freeStat.getResultSet();
                                        if (!res) {  // Εάν δεν υπάρχουν αποτελέσματα στις εγγραφές
δεδομένων
                                                IBBPMAServer.frame2.showInfo(count + " record(s)
affected.");
                                                IBBPMAServer.frame2.showInfo("-DONE-");
                                        } else {  // Αλλιώς δείξτε τα αποτελέσματα
                                                while(results.next()) {

        IBBPMAServer.frame2.showInfo(decrypt(results.getString(1)) + "  " +
decrypt(results.getString(2)) + "  " + decrypt(results.getString(3)) + "  " + decrypt(results.getString(4))
+ "  " + decrypt(results.getString(5)) + "  " + decrypt(results.getString(6)) + "  " +
decrypt(results.getString(7)));
                                                } //FIXME: Βελτιώνει ευελιξία καθώς υπολογίζει 7
στήλες από αποτελέσματα
                                                IBBPMAServer.frame2.showInfo("-DONE-");
                                        }

                                        freeStat.close();

                        } else {

                                throw new Exception("doSQL - invalid character(s) used");

                        }

                } catch(Exception sqle) {
                        IBBPMAServer.frame2.showError("Database error: " + sqle.toString());
                        IBBPMAServer.NORM.error("Database error: " + sqle.toString());
                }

        } //EOF doSQL()


} //EOF Class
```

iv. IBBPMAFrame1

```
package IBBPMAServer;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Κύρια οθόνη κατάστασης
 *
 */
public class ERFrame1 extends Frame implements ActionListener
{

        ScrollPane jScrollPane1 = new ScrollPane();
    TextView tv;
        ScrollView sv;
        Font f;
        Font f2;
        TextStyle normal;
        TextStyle red;
        TextStyle blue;
        TextStyle green;
        Button jButton1 = new Button();
        Button jButton2 = new Button();
        Button jButton3 = new Button();

        // Μεταβλητες για το menu
        MenuBar mb;
        Menu m1;
        MenuItem item1, item2, item3, item4;

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}


        protected ERFrame1() {

                // Ετοίμασε το textView
                tv = new TextView(true);
    sv = new ScrollView(tv);
    f = new Font("Courier", Font.PLAIN, 12);
    f2 = new Font("Courier", Font.BOLD, 12);
    normal = new TextStyle(f, Color.black);
    red = new TextStyle(f2, Color.red);
    blue = new TextStyle(f, Color.blue);
    green = new TextStyle(f, Color.green);

                // Κατασκευή του frame
                jButton1.setLabel("IBB Database Console");
                jButton2.setLabel("Build Keys");
                jButton3.setLabel("Import Users");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("IBB/PMA Server Messages");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 3, 0, 0, GridBagConstraints.BOTH,
```

```
                1.0f, 1.0f, sv);
        IBBPMAPanel buttonPane = new IBBPMAPanel();
        scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jButton1);
        scrollPanel.addComponent( 1, 1, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jButton2);
        scrollPanel.addComponent( 1, 2, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jButton3);

        setTitle("IBB/PMA System: IBBPMAServer");
        setSize(new java.awt.Dimension(450, 300));
        add(scrollPanel);

        // Κατσκευή του menu
        mb = new MenuBar();
        setMenuBar(mb);

        m1 = new Menu("Actions", true);
        mb.add(m1);
        item1 = new MenuItem("Import Users");
        m1.add(item1);
        item2 = new MenuItem("Export Keys");
        m1.add(item2);
        item3 = new MenuItem("IBB Database Console");
        m1.add(item3);
        m1.addSeparator();
        item4 = new MenuItem("Exit");
        m1.add(item4);

}

protected void initComponents() throws Exception
{

        jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton1ActionPerformed(e);
                }
        });

        jButton2.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton2ActionPerformed(e);
                }
        });

        jButton3.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton3ActionPerformed(e);
                }
        });

        addWindowListener(new java.awt.event.WindowAdapter() {
                public void windowClosing(java.awt.event.WindowEvent e) {
                        thisWindowClosing(e);
                }
        });

        m1.addActionListener(this);
```

```
            item1.setActionCommand("item1");
            item2.setActionCommand("item2");
            item3.setActionCommand("item3");
            item4.setActionCommand("item4");

    }

    // Εμφάνιση πληροφοριακού μηνύματος (black, plain text)
    protected void showInfo(String msg) {
            showMsg(msg);
    }

    // Εμφάνιση μηνύματος προσοχής (red, bold text)
    protected void showError(String msg) {
            msg += "\n";
            tv.append(msg, red);
    }

    // Εμφάνιση κειμένου με ειδικό AttributeSet
    protected void showMsg(String msg) {
            msg += "\n";
            tv.append(msg, normal);
    }

    void thisWindowClosing(java.awt.event.WindowEvent e)
    {
            IBBPMAServer.serv.stopServer();

            setVisible(false);
            dispose();
            System.exit(0);
    }

    public void jButton1ActionPerformed(java.awt.event.ActionEvent e) {
            jButton1.setEnabled(false);
            IBBPMAServer.frame2.setVisible(true);
    }

    public void jButton2ActionPerformed(java.awt.event.ActionEvent e) {
            try {
                    DBase.makeAllKeys();
            } catch (Exception ex) {
                    IBBPMAServer.NORM.error("Build keys error: " + ex.getMessage());
            }
    }

    public void jButton3ActionPerformed(java.awt.event.ActionEvent e) {
            try {
                    DBase.importUsers();
            } catch (Exception ex) {
                    IBBPMAServer.NORM.error("Import users error: " + ex.getMessage());
            }
    }

    public void actionPerformed(ActionEvent e) {
            if (e.getActionCommand()=="item1") {
                    jButton3ActionPerformed(e);
            } else if (e.getActionCommand()=="item2") {
                    jButton2ActionPerformed(e);
            } else if (e.getActionCommand()=="item3") {
```

```
                    jButton1ActionPerformed(e);
                } else {
                    IBBPMAServer.serv.stopServer();

                    setVisible(false);
                    dispose();
                    System.exit(0);
                }
            }

} //EOF class
```

v. IBBPMAFrame1.jfrm

```
Metrowerks RAD Data V 1
BeginClasses
  {9BE3D9E0-166E-11D1-B2D8-0060081C5489} "#base-object-class"
  {24A693F8-14E5-11D2-B334-00600819ADE3} "IBBPMAServer"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "ERFrame1"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JTextPane"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JProgressBar"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JLabel"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JMenuBar"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JMenu"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JMenuItem"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JSeparator"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JScrollPane"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JButton"
EndClasses
BeginObject IBBPMAServer.ERFrame1 "ERFrame1"
  BeginProperties
    enabled = True
    JMenuBar = <none>
    foreground = black
    location = "0, 0"
    title = "IBB/PMA Voting System"
    resizable = True
    background = "204, 204, 204"
    font = Application-PLAIN-10
    layout = None
    cursor = Default
    menuBar = <none>
    size = "349, 266"
  EndProperties
  BeginInternalProperties
    UseSwing 11 = 1
    IsApplet 11 = 0
  EndInternalProperties
  BeginEventConnections
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAServer.ERFrame1"
      SinkPath = "\\\\IBBPMAServer.ERFrame1"
      EventName = windowClosing
      EventSetName = window
      HandlerName = thisWindowClosing
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAServer.ERFrame1\\jButton1"
      SinkPath = "\\\\IBBPMAServer.ERFrame1"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton1ActionPerformed
    EndEventConnection
  EndEventConnections
  BeginObject jScrollPane1 "javax.swing.JScrollPane"
    BeginProperties
      requestFocusEnabled = True
      maximumSize = "32767, 32767"
      enabled = True
      foreground = black
      location = "10, 10"
      autoscrolls = False
```

```
      visible = True
      background = "204, 204, 204"
      font = Dialog-PLAIN-12
      verticalScrollBarPolicy = 20
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = False
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "22, 22"
      size = "330, 220"
      doubleBuffered = False
      horizontalScrollBarPolicy = 30
    EndProperties
    BeginInternalProperties
    EndInternalProperties
    BeginObject jTextPane1 "javax.swing.JTextPane"
      BeginProperties
        requestFocusEnabled = True
        contentType = text/plain
        selectionStart = 0
        maximumSize = "2147483647, 2147483647"
        enabled = True
        text = ""
        selectedTextColor = black
        foreground = black
        location = "0, 0"
        autoscrolls = True
        visible = True
        background = white
        selectionColor = "204, 204, 255"
        caretColor = black
        font = Application-PLAIN-10
        cursor = Default
        alignmentY = 0.5
        alignmentX = 0.5
        opaque = True
        caretPosition = 0
        selectionEnd = 0
        disabledTextColor = "153, 153, 153"
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "11, 6"
        size = "327, 217"
        doubleBuffered = False
        editable = True
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
  EndObject
  BeginObject jButton1 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = "Database Console"
      actionCommand = ""
```

```
            contentAreaFilled = True
            location = "10, 240"
            foreground = black
            autoscrolls = False
            visible = True
            rolloverEnabled = False
            background = "204, 204, 204"
            borderPainted = True
            font = Application-BOLD-10
            horizontalAlignment = 0
            cursor = Default
            alignmentY = 0.0
            alignmentX = 0.0
            opaque = True
            defaultCapable = True
            selected = False
            verticalTextPosition = 0
            debugGraphicsOptions = 0
            toolTipText = ""
            minimumSize = "35, 11"
            size = "170, 20"
            focusPainted = True
            doubleBuffered = False
            horizontalTextPosition = 4
        EndProperties
        BeginInternalProperties
        EndInternalProperties
    EndObject
EndObject
```

vi. IBBPMAFrame2.java

```
package IBBPMAServer;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/*
 * Απλή κονσόλα βάσης δεδομένων για την εκτέλεση εντολών SQL.
 *
 */
public class ERFrame2 extends Frame implements ActionListener
{

        Label jLabel1 = new Label();
        TextField jTextField1 = new TextField();
        Button jButton1 = new Button();
        Label jLabel2 = new Label();
  TextView tv;
        ScrollView sv;
        Font f;
        Font f2;
        TextStyle normal;
        TextStyle red;
        TextStyle blue;
        TextStyle green;

        // Μεταβλητές του menu
        MenuBar mb;
        Menu m1;
        MenuItem item1, item2;

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}

        public ERFrame2() {

                // Ετοιμασία του textView
                tv = new TextView(true);
    sv = new ScrollView(tv);
    f = new Font("Courier", Font.PLAIN, 12);
    f2 = new Font("Courier", Font.BOLD, 12);
    normal = new TextStyle(f, Color.black);
    red = new TextStyle(f2, Color.red);
    blue = new TextStyle(f, Color.blue);
    green = new TextStyle(f, Color.green);

                // Δημιουργία του frame
                setTitle("IBB/PMA: Database Console");
                jLabel1.setText("Results:");
                jButton1.setLabel("Go");
                jLabel2.setText("Command entry:");

                IBBPMAPanel scrollPanel = new IBBPMAPanel("IBB/PMA Server Results");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, sv);
                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
```

```
                1.0f, 0.0f, jLabel2);
        scrollPanel.addComponent( 2, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jTextField1);
        scrollPanel.addComponent( 2, 1, 1, 1, 0, 0, GridBagConstraints.BOTH,
                1.0f, 0.0f, jButton1);

        add(scrollPanel, BorderLayout.CENTER);
        setSize(new java.awt.Dimension(472, 315));

        // Δημιουργία του menu
        mb = new MenuBar();
        setMenuBar(mb);

        m1 = new Menu("Actions", true);
        mb.add(m1);
        item1 = new MenuItem("Close window");
        m1.add(item1);
        m1.addSeparator();
        item2 = new MenuItem("Exit");
        m1.add(item2);
}

public void initComponents() throws Exception
{
        jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton1ActionPerformed(e);
                }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
                public void windowClosing(java.awt.event.WindowEvent e) {
                        thisWindowClosing(e);
                }
        });

        m1.addActionListener(this);

        item1.setActionCommand("item1");
        item2.setActionCommand("item2");
}

// Εμφάνιση πληροφοριακού μηνύματος (black, plain text)
protected void showInfo(String msg) {
        showMsg(msg);
}

// Εμφάνιση προειδοποιητικού μηνύματος (red, bold text)
protected void showError(String msg) {
        msg += "\n";
        tv.append(msg, red);
}

// Εμφάνιση μηνύματος με συγκεκριμένο AttributeSet
protected void showMsg(String msg) {
        msg += "\n";
        tv.append(msg, normal);
}

// Κλείσιμο του παραθύρου όταν επιλεγεί το close button
void thisWindowClosing(java.awt.event.WindowEvent e)
```

```
        {
                IBBPMAServer.frame.jButton1.setEnabled(true);
                setVisible(false);
        }

        public void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

                String userCommand = jTextField1.getText();

                showInfo("Executing: " + userCommand);
                jTextField1.setText("");

                DBase.doSQL(userCommand);

        }

        public void actionPerformed(ActionEvent e) {
                if (e.getActionCommand()=="item1") {
                        IBBPMAServer.frame.jButton1.setEnabled(true);
                        setVisible(false);
                } else {
                        IBBPMAServer.serv.stopServer();

                        setVisible(false);
                        IBBPMAServer.frame.setVisible(false);
                        dispose();
                        System.exit(0);
                }
        }

} //EOF class
```

vii. IBBPMAFrame2.jfrm

```
Metrowerks RAD Data V 1
BeginClasses
  {9BE3D9E0-166E-11D1-B2D8-0060081C5489} "#base-object-class"
  {24A693F8-14E5-11D2-B334-00600819ADE3} "IBBPMAServer"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "ERFrame1"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JTextPane"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "DBFrame"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JLabel"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JScrollPane"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "ERFrame2"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JTextField"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JButton"
EndClasses
BeginObject IBBPMAServer.ERFrame2 "ERFrame2"
  BeginProperties
    enabled = True
    JMenuBar = <none>
    foreground = black
    location = "0, 0"
    title = "IBB/PMA Voting System: Database Console"
    resizable = True
    background = "204, 204, 204"
    font = Application-PLAIN-10
    layout = None
    cursor = Default
    menuBar = <none>
    size = "471, 315"
  EndProperties
  BeginInternalProperties
    UseSwing 11 = 1
    IsApplet 11 = 0
  EndInternalProperties
  BeginEventConnections
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAServer.ERFrame2"
      SinkPath = "\\\\IBBPMAServer.ERFrame2"
      EventName = windowClosing
      EventSetName = window
      HandlerName = thisWindowClosing
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAServer.ERFrame2\\jButton1"
      SinkPath = "\\\\IBBPMAServer.ERFrame2"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton1ActionPerformed
    EndEventConnection
  EndEventConnections
  BeginObject jLabel1 "javax.swing.JLabel"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      iconTextGap = 4
      maximumSize = "0, 0"
      enabled = True
      text = Results:
      foreground = "102, 102, 153"
      location = "10, 0"
```

```
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Application-BOLD-10
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.0
      opaque = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "0, 0"
      size = "280, 20"
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jTextField1 "javax.swing.JTextField"
    BeginProperties
      requestFocusEnabled = True
      selectionStart = 0
      maximumSize = "2147483647, 2147483647"
      enabled = True
      scrollOffset = 0
      text = ""
      selectedTextColor = black
      location = "10, 290"
      foreground = black
      autoscrolls = True
      visible = True
      background = white
      selectionColor = "204, 204, 255"
      caretColor = black
      font = Application-PLAIN-10
      columns = 0
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = True
      caretPosition = 0
      selectionEnd = 0
      disabledTextColor = "153, 153, 153"
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "4, 17"
      size = "390, 20"
      doubleBuffered = False
      editable = True
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jButton1 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
```

```
        maximumSize = "50, 20"
        enabled = True
        text = Go
        actionCommand = ""
        contentAreaFilled = True
        location = "410, 290"
        foreground = black
        autoscrolls = False
        visible = True
        rolloverEnabled = False
        background = "204, 204, 204"
        borderPainted = True
        font = Application-BOLD-10
        horizontalAlignment = 0
        cursor = Default
        alignmentY = 0.0
        alignmentX = 0.0
        opaque = True
        defaultCapable = True
        selected = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = "Click here to execute the SQL code you have entered in the IBB/PMA
Database"
        minimumSize = "50, 20"
        size = "50, 20"
        focusPainted = True
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
    BeginObject jLabel2 "javax.swing.JLabel"
      BeginProperties
        requestFocusEnabled = True
        verticalAlignment = 0
        iconTextGap = 4
        maximumSize = "0, 0"
        enabled = True
        text = "Command Entry:"
        foreground = "102, 102, 153"
        location = "10, 270"
        autoscrolls = False
        visible = True
        background = "204, 204, 204"
        font = Application-BOLD-10
        horizontalAlignment = 2
        cursor = Default
        alignmentY = 0.5
        alignmentX = 0.0
        opaque = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "0, 0"
        size = "260, 20"
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
```

```
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jScrollPane1 "javax.swing.JScrollPane"
    BeginProperties
      requestFocusEnabled = True
      maximumSize = "32767, 32767"
      enabled = True
      foreground = black
      location = "10, 20"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Dialog-PLAIN-12
      verticalScrollBarPolicy = 20
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = False
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "22, 22"
      size = "450, 250"
      doubleBuffered = False
      horizontalScrollBarPolicy = 30
    EndProperties
    BeginInternalProperties
    EndInternalProperties
    BeginObject jTextPane1 "javax.swing.JTextPane"
      BeginProperties
        requestFocusEnabled = True
        contentType = text/plain
        selectionStart = 0
        maximumSize = "2147483647, 2147483647"
        enabled = True
        text = ""
        selectedTextColor = black
        foreground = black
        location = "0, 0"
        autoscrolls = True
        visible = True
        background = white
        selectionColor = "204, 204, 255"
        caretColor = black
        font = Application-PLAIN-10
        cursor = Default
        alignmentY = 0.5
        alignmentX = 0.5
        opaque = True
        caretPosition = 0
        selectionEnd = 0
        disabledTextColor = "153, 153, 153"
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "11, 6"
        size = "447, 247"
        doubleBuffered = False
        editable = True
      EndProperties
      BeginInternalProperties
```

```
        EndInternalProperties
      EndObject
    EndObject
EndObject
```

viii. IBBPMAServer.java

```
package IBBPMAServer;

import java.net.*;
import java.io.*;
import java.awt.*;

import IBBPMA.awt.*;
import IBBPMA.util.*;

import org.apache.log4j.*;

/**
 * Εκτελεί το πρόγραμμα IBBPMAServer και αρχικοποιεί το GUI.
 *
 * Αρχικοποιείται η βάση δεδομένων και προετοιμάζεται το σύστημα
 * logging
 * Το Logging γίνεται με <code>log4j</code>, και περισσότερες
 * πληροφορίες έχει στο
 * <a href="http://www.log4j.org">www.log4j.org</a>.
 *
 *
 */

public class IBBPMAServer
{
        /* ΣΗΜΕΙΩΣΗ : ΠΟΛΥ ΣΗΜΑΝΤΙΚΟ! */
        // Η constant που χρησιμοποιείται για να αποθηκευτεί η τιμή που
    // χρησιμοποιείται σαν ελεύθερος αριθμός θήρας
        // Κάθε εκλογή θα πρέπει να αλλάζονται οι τιμές

//#ifdef INSTALL
        private static final int freePort = 1111;

        /** Δημόσια και ιδιωτικά κλειδιά για την επικοινωνία */
//#ifdef INSTALL
        protected static final String rt_public_key =
"00110100010800819374 5AA3A2733EF1140758495439CB4CD188C4F2B5D2FA84C3C533465991
257EAFC2BCAC0AC07F1531C97489723A353E7A1C911E687A5E834CD872B534480D8B4F9107
E48C994A0D5B68870DE4DC3A51C604572446F166E02B8F234C44158FD373A06CFF30E3BBA2
F1E670743D7B855CC2D3EF395D440A702D57013A5146171E4B7C78343FB2DAA47E56C63A2B
5CFA57FF8892399F0A7D8E47A0203C687A158D3E0802AE02D3C1EA4AC625B641430FA9A1A5
0FC095461EED1DF401F93365CD650D02CFD4392BE1349B29900A0DCA648658FAFA3E8C3300
E44C705AF3930F5EDF28B5D40C89251F440C9038DD9F533A849965EB71617D91A5CFD5621ED
6A5D1";
//#ifdef INSTALL
        protected static final String fc_public_key = "";
//#ifdef INSTALL
        protected static final String pm_public_key = "";
//#ifdef INSTALL
        protected static final String er_private_key =
"07FF4A2B93480B1AC5AFDE0CE5FB7A00864BDB41FF1BCFC7FB322EF74BE12299A75B22FA
1E026E815BFC5EBA37E50722655F20CD0A79BCA852C649D55D01509BC54C0361ABA4BE7E7
ED6F0DE15740355EF04FC570BD2A1CAA43BF7848A1DEE4DAC2086D86ADB935909FC46CF2
DEB4D7348BD0D80833ABF45CD43C2A5D92910E96D73AA578D83BFC8787487DC74B3E2940
D8B00FA5817416E62A355E154411030520C2CFAAB83C32B09B7DD813FEFD0BEF224A34CD31
82484FBD20138E5FD1F036752013192124E42F90C6D4E9CA2EB9AA5AA5E622A4459ABFD224
F95F09CC4DA3D6EDA225FAC3035D03314635FAD1AC9E461D96251CC8497C19DA81B7323D1
E975210400E50682CE1B0734FEEC88B2AA924C277B958F73324C74489A9B20015EF61A610B43
```

```
3E14EEF6B81A1F127882093AC1E4CCC7CE9380978F2687F9C35B0C6F63C482C7DD1CFB439E
12ADF5B26BA9F5E413529272A43E3D732087DBBBDB74663C2E53CD8E22404F60F8DE1DDEC
2EACA701DAF7C3724CCA0390FCEFBC3D2B9D9C7BF950400A6CD5010197C85D2DEED12730
FE6CA466B3A4042100F1742CB0B7301EEEC2FAC2EE028ABBECC6E44BD277400693601FFEF1
39814135AC9E87AB1E05C322FCA483F35B7772FA41EE2A62E49768C9ED6BC6A8957328FFF8E
D23210FFCBBF1395E55538BAB3AEF29CF1CDA7560896AE1DA8E4B04BD14BD4CD93A7A8A
8E040101DAD0400829001A26DBAFC192B567F9F04D1BB49C5BE100D03468E19DC5B600A474
85448220E0C820EB5DF8ED6EB24304AF42D233559DBD97FDCD16AC7056BFBF9245CCBF18C
48C982BFF91137B94A9669E6B198E33B37DC8B7202E4D66E80506313E9108BE98696897AE1D4
1A7802DB23D99333D6208F187C0A9AC07FEE484D1AC37EE8";

        // AWT Πλαίσια
        protected static ERFrame1 frame;
        protected static CryptFrame frame1;
        protected static ERFrame2 frame2;

        // Ο εξυπηρετητής
        protected static TCPServer serv;

        // logging Κατηγορίες
        protected static Category NORM;
        protected static Category DEV;

        protected IBBPMAServer()
        {
                try {

                        IBBPMAPanel.showSplash(4000);

                        // Αρχικοποίηση GUI
                        frame = new ERFrame1();
                        frame.initComponents();
                        frame1 = new CryptFrame();
                        frame1.initComponents();
                        frame2 = new ERFrame2();
                        frame2.initComponents();
                        frame1.setVisible(true);

                }
                catch (Exception e) {
                        e.printStackTrace();
                }
        }

        // Κύριο σημείο εισόδου
        static protected void main(String[] args)
        {
                new IBBPMAServer();

        }

        /**
         * Αρχικοποιεί τη βάση δεδομένων, το σύστημα logging και τον TCP/IP εξυπηρετητή.
         *
         * Επίσης προσαρμόζει το GUI ανάλογα με το λειτουργικό που χρησιμοποιείται.
         */
        protected static void startDaemon() {

                try {
```

```
                        // Αρχικοποίηση του logging system
                        ScreenAppender A1 = new ScreenAppender();

                        // Υπάρχει το log ? Αν ναι ξεκινάει να πέρνει δεδομένα
                        String tempInData = "";
                        String secInitString = "STARTING VALUE";
                        try {
                                BufferedReader t_in = new BufferedReader(new
FileReader("IBBPMAServer.log"));
                                String inData = t_in.readLine();
                                while (inData!=null) { // Πήγαινε στην τελευταία σειρά
                                        tempInData = inData;
                                        inData = t_in.readLine();
                                }
                                t_in.close();
                                inData = tempInData;

                                if (inData!="" || inData!=" ") {
                                        int i=0;
                                        int t=0;

                                        while (inData.charAt(i)!=']') {
                                                i++;
                                        }

                                        t = i+5;

                                        while (inData.charAt(t)!='-') {
                                                t++;
                                        }

                                        secInitString = inData.substring(t+2); // Κατέγραψε τα
δεδομένα
                                }

                        } catch (FileNotFoundException fnfe) {

                        }

                        SecureAppender A2 = new SecureAppender(secInitString);
                        PropertyConfigurator.configure("log4j.IBBPMAServer.properties");

                        NORM = Category.getInstance("NORM");
                        NORM.addAppender(A1);
                        NORM.addAppender(A2);
                        DEV = Category.getInstance("DEV");
                        DEV.addAppender(A1);
                        DEV.addAppender(A2);

                        IBBPMAServer.NORM.info("IBB/PMA IBBPMAServer 0.0.0.01
εκκινείται...");

                        // Αρχικοποίηση της database
                        DBase.init();

                        // Αρχικοποίηση του AuthSys
                        AuthKey.init();
                        RSAEncrypt.init();

                        // Εκκίνηση των TCPServer υπηρεσιών
```

```
                    serv = new TCPServer();
                    serv.startServer(freePort);

                    IBBPMAServer.NORM.info("IBB/PMA Server daemon started");

                }
                catch (Exception e) {
                    e.printStackTrace();
                }


        } //EOF startDaemon

} //EOF Class
```

ix. IBBPMAServerProtocol.java

```
package IBBPMAServer;

import java.util.*;
import IBBPMA.util.*;

/**
 * Επεξεργάζεται όλα τα αλφαριθμητικά που έρχονται από το
 * <code>TCPServer</code> αλλά στέλνει
 * όλα τα λάθη στο <code>TCPServer</code> που διαχειρίζεται τα λάθη
 * κλείνοντας τις απαραίτητες συνδέσεις.
 */

public class IBBPMAServerProtocol {


        private static Vector checkDates = new Vector(1000);
        private static Vector checkUsers = new Vector(1000);

        private static Vector session_data = new Vector(1000);
        private static Vector client_hmac_keys = new Vector(1000);
        private static Vector server_hmac_keys = new Vector(1000);
        private static Vector session_count = new Vector(1000);


        /**
         * Προσθέτει μια εγγραφή στο Users Vector για χρήση αργότερα.
         *
         * @param  n  To index στο οποίο γίνεται η είσοδος
         * @param  data  πακέτο μηνύματος για αποθήκευση
         */
        private static void setUsers(int n, String data) {

                checkUsers.insertElementAt(data, n);

        } //EOF Users


        /**
         * Εισάγει μια εγγραφή στο checkDate Vector για μετέπειτα
    * χρήση.
         *
         * @param cdate   Date που εισάγεται
         * @returns    To index στο οποίο εισείχθει η Date
         */
        private static int setCheck(String cdate) {

                int r = checkDates.size();

                checkDates.insertElementAt(cdate, r);

                return r;

        } //EOF setCheck


        /**
         * Προσπαθεί να βρει έναν χρήστη για ελέγχους ασφαλείας.
         *
```

```
        * @param user   User για να βρει και να συγκρίνει
        *
        * @returns  Ένας string πίνακας με τα αποτελέσματα των
    * ταιριασμένων date και τον συνδεδεμένο χρήστη
        */

        private static String[] getUsers(String user) {

                int i=0;
                String[] r = new String[] {"","",""};

                i = checkUsers.indexOf(user);

                if (i==-1) {
                        r[0] = "FALSE";
                } else {
                        r[0] = checkUsers.elementAt(i).toString(); // Συλλογή user
                        r[1] = checkDates.elementAt(i).toString(); // Συλλογή Date
                        checkDates.removeElementAt(i); // καθαρισμός data
                        checkUsers.removeElementAt(i);
                }

                return r;

        } //EOF getUsers


        /**
        * Εισάγει δεδομένα στο Vectors για να κρατάει τα session_data
        *
        * @param session_key  Το session_key κρυπτογράφησης με το
    * οποίο γίνεται το index
        * @param client_hmac_key  Κλειδί Client HMAC
        * @param server_hmac_key  Κλειδί Server HMAC
        * @param session_counter  Packet counter
        */

        private static void setSession(String session_key, String client_hmac_key, String
server_hmac_key, int session_counter) {

                int r = session_data.size();

                session_data.insertElementAt(session_key, r);
                client_hmac_keys.insertElementAt(client_hmac_key, r);
                server_hmac_keys.insertElementAt(server_hmac_key, r);
                session_count.insertElementAt(new Integer(session_counter).toString(), r);

        } //EOF setSession


        /**
        * Προσπαθεί να βρει δεδομένα session που ταιριάζουν.
        *
        * @param session_key
        * @return Ένας string πίνακας με τα αποτελέσματα του match
        */
        private static String[] getSession(String session_key) {

                int i=0;
                String[] r = new String[] {"","",""};
```

```
                i = session_data.indexOf(session_key);

                if (i==-1) {
                        r[0] = "FALSE";
                } else {
                        r[0] = client_hmac_keys.elementAt(i).toString(); // συλλέγει data
                        r[1] = server_hmac_keys.elementAt(i).toString();
                        r[2] = session_count.elementAt(i).toString();
                        session_data.removeElementAt(i);  // καθαρίζει τα δεδομένα
                        client_hmac_keys.removeElementAt(i);
                        server_hmac_keys.removeElementAt(i);
                        session_count.removeElementAt(i);
                }

                return r;

        } //EOF getSession
```

```
        /**
         * Το <code>process</code> αναλύει τα αλφαριθμητικά που
    * στέλνονται από το <code>TCPServer</code>
         * και αποφασίζει εάν αυτά τα δεδομένα είναι έγκυρα πακέτα.
         *
         * Εάν είναι στέλνονται τα αντίστοιχα αποτελέσματα στο
         * <code>TCPServer</code> για αποστολή στον πελάτη.
         *
         * @param inputData  Περιέχει τα περιεχόμενα του πακέτου που
    *                   παρέλαβε
         * @param session_key  Το session κλειδί για κρυπτογράφηση
         * @returns  ΤΟ String με τον κώδικα για να επιστραφεί στον
    *           πελάτη
         */

        protected static String process(String inputData, String session_key) throws Exception {

                String outputData="";

                if (inputData.charAt(0) == 'C') {

                        /* Ορισμός σαν έγκυρη ψήφος*/
                        Packet p = new Packet(inputData);

                        String[] sess_data = getSession(session_key);

                        if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) {  // Έλεγχος tamper

                                // Χωρισμός του πακέτου μηνύματος
                                String[] pp = p.splitMessage();
                                String check = pp[0];
                                String name = pp[1];
                                String code = pp[2];
                                String pword = pp[3];

                                String[] t = getUsers(name + "-" + code + "-" + pword); //Σύγκριση
του δεύτερου time stamp
```

```
                        if (!t[0].equals("FALSE")&&check.equals(t[1])) {  //Ασφάλεια
date ok
                                /* register vote */

                                try {
                                        DBase.PMconfirmVoted(name, code,
pword); //Ορισμός σαν voted
                                        outputData = new
Packet('S',"OK",session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);

        setSession(session_key,sess_data[0],sess_data[1],(new Integer(sess_data[2]).intValue()+2));
                                } catch (Exception e) {
                                        IBBPMAServer.NORM.error("Voted
registration error: " + e.getMessage());
                                        outputData = "ERROR";
                                }
                        } else {
                                IBBPMAServer.NORM.warn("Voted security
failure");
                                outputData = "ERROR";
                        }

                } else {
                        IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                        outputData = "ERROR";
                }

        } else if (inputData.charAt(0) == 'D') {

                /* Διαγνωστικό */
                Packet p = new Packet(inputData);

                String[] sess_data = getSession(session_key);

                if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) {  // tamper check

                        IBBPMAServer.NORM.info("Diagnostic packet received");

                        if (p.getMessage().equals("TEST")) {
                                outputData = new
Packet('D',"OK",session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);
                                setSession(session_key,sess_data[0],sess_data[1],(new
Integer(sess_data[2]).intValue()+2));
                        } else {
                                IBBPMAServer.NORM.warn("Unrecognized diagnostic
command.");
                                outputData = "ERROR";
                        }

                } else {
                        IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                        outputData = "ERROR";
                }

        } else if (inputData.charAt(0) == 'E') {

                /* ER Έλεγχος */
```

```
                    Packet p = new Packet(inputData);

                    /*
                     * Το MAC είναι πάντα 31 χαρακτήρες!
                     */
                    if (inputData.length() >= 47) {  // simple validity check to save work

                            String[] sess_data = getSession(session_key);

                            if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) {

                                    try {

                                            // Πέρασμα για να πάρει ξεχωριστά πεδία
δεδομένων

                                            String[] pp = p.splitMessage();
                                            String name = pp[0];
                                            String code = pp[1];
                                            String pword = pp[2];

                                            if (code.equals("")|code.equals(" ")) {
                                                    IBBPMAServer.NORM.warn("Bad auth
code data in packet.");
                                                    outputData = "ERROR";
                                            } else if (pword.equals("")|pword.equals(" ")) {
                                                    IBBPMAServer.NORM.warn("Bad auth
pword data in packet.");
                                                    outputData = "ERROR";
                                            } else if (name.equals("")|name.equals(" ")) {
                                                    IBBPMAServer.NORM.warn("Bad auth
name data in packet.");
                                                    outputData = "ERROR";
                                            } else {
                                                    /* Τα δεδομένα φαίνονται σωστά οπότε
συνεχίζει στην εγγραφή */
                                                    try {
                                                            if (DBase.checkER(name,
code, pword)) {
                                                                    // Πριν να σταλεί η
επιβεβαίωση κάνε το auth key
                                                                    String ak =
AuthKey.build(name,code,pword,2);

                                                                    outputData = new
Packet('A',"TRUE-" + ak,session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);

        setSession(session_key,sess_data[0],sess_data[1],(new Integer(sess_data[2]).intValue()+2));

        IBBPMAServer.NORM.info("Voter authorised! You may Procceed to the Voting Phase");
                                                            } else {
                                                                    // Αν δεν υπήρχε
εγγραφή στη βάση
                                                                    outputData = new
Packet('A',"FALSE-NOKEY",session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);

        setSession(session_key,sess_data[0],sess_data[1],(new Integer(sess_data[2]).intValue()+2));
```

```
IBBPMAServer.NORM.info("Voter authorisation failed!");
                                                        }
                                                } catch (Exception e) {
                                                        outputData = "ERROR";

IBBPMAServer.NORM.error("Auth dbase/auth key exception: " + e.getMessage());
                                                }

                                        }

                                } catch (IndexOutOfBoundsException e) {
                                        IBBPMAServer.NORM.error("Unrecognized or
bad packet received.");
                                        outputData = "ERROR";
                                }

                        } else {
                                IBBPMAServer.NORM.error("Packet corrupted or
altered!");
                                outputData = "ERROR";
                        }

                } else {
                        IBBPMAServer.NORM.warn("Bad Packet!");
                        outputData = "ERROR";
                }

        } else if (inputData.charAt(0) == 'I') {


                Packet p = new Packet(inputData);

                if (AuthSys.checkDigest(p)) {  // tamper check

                        String client_key = AuthSys.seedToKey(p.getMessage());  //
Δημιουργία κλειδιών
                        String server_seed = RSAEncrypt.makeSessionKey();
                        String server_key = AuthSys.seedToKey(server_seed);

                        setSession(session_key,client_key,server_key,1);  // Αποθήκευση
δεδομένων συνεδρίας

                        outputData = new Packet('I',server_seed + "-" +
AuthSys.makeHMAC("I|" +
p.getMessage(),client_key,0),session_key,"",server_key,1).toCryptString(false);
                } else {
                        IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                        outputData = "ERROR";
                }

        } else if (inputData.charAt(0) == 'K') {

                /* auth key πακέτο */
                IBBPMAServer.NORM.info("Auth Key packet received");

                Packet p = new Packet(inputData);

                if (inputData.length() >= 88) {
```

```
                        String[] sess_data = getSession(session_key);

                        if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) { // tamper check
                                if (DBase.confirmVoted(p.getMessage())) {
                                        // Υπάρχει ταυτοποίηση
                                        outputData = new Packet('K',p.getMessage() + "-
OK",session_key,"",sess_data[1],(new Integer(sess_data[2]).intValue()+2)).toCryptString(false);

        setSession(session_key,sess_data[0],sess_data[1],(new Integer(sess_data[2]).intValue()+2));
                                } else {
                                        // Δεν βρέθηκε εγγραφη
                                        IBBPMAServer.NORM.warn("No Key match: "
+ p.getMessage());
                                        outputData = new Packet('K',p.getMessage() + "-
FALSE",session_key,"",sess_data[1],(new Integer(sess_data[2]).intValue()+2)).toCryptString(false);

        setSession(session_key,sess_data[0],sess_data[1],(new Integer(sess_data[2]).intValue()+2));
                                }
                        } else {
                                IBBPMAServer.NORM.warn("Packet corrupted or
altered!");
                                outputData = "ERROR";
                        }

                } else {
                        IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                        outputData = "ERROR";
                }

        } else if (inputData.charAt(0) == 'P') {

                /* IBBPMAServer ER Έλεγχος */
                IBBPMAServer.NORM.info("PMERCheck packet received");

                Packet p = new Packet(inputData);

                if (inputData.length() >= 47) {

                        String[] sess_data = getSession(session_key);

                        if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) {
                                try {

                                        // Πέρασμα για να βρεθούν πεδία δεδομένων
                                        String[] pp = p.splitMessage();
                                        String name = pp[0];
                                        String code = pp[1];
                                        String pword = pp[2];


                                        if (code.equals("")|code.equals(" ")) {
                                                IBBPMAServer.NORM.warn("Bad auth
code data in packet.");
                                                outputData = "ERROR";
                                        } else if (pword.equals("")|pword.equals(" ")) {
                                                IBBPMAServer.NORM.warn("Bad auth
pword data in packet.");
                                                outputData = "ERROR";
```

```
                                        } else if (name.equals("")|name.equals(" ")) {
                                                IBBPMAServer.NORM.warn("Bad auth
name data in packet.");
                                                outputData = "ERROR";
                                        } else {
                                                try {
                                                        String[] pmer =
DBase.PMcheckER(name, code, pword);
                                                        if (pmer[0].equals("ERROR"))
{
                                                                // datbase check didn't
match
                                                                outputData = new
Packet('P',"FALSE-FALSE",session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);

        setSession(session_key,sess_data[0],sess_data[1],(new Integer(sess_data[2]).intValue()+2));

        IBBPMAServer.NORM.info("No PMERCheck match.");
                                                        } else {
                                                                outputData = new
Packet('P',pmer[0] + "-" + pmer[1],session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);

        setSession(session_key,sess_data[0],sess_data[1],(new Integer(sess_data[2]).intValue()+2));

        IBBPMAServer.NORM.info("PMERCheck match.");
                                                        }
                                                } catch (Exception e) {
                                                        outputData = "ERROR";

        IBBPMAServer.NORM.error("PMERChecl dbase exception: " + e.getMessage());
                                                }

                                        }

                                } catch (IndexOutOfBoundsException e) {
                                        IBBPMAServer.NORM.error("Unrecognized or
bad packet received.");
                                        outputData = "ERROR";
                                }
                        } else {
                                IBBPMAServer.NORM.warn("Packet corrupted or
altered!");
                                outputData = "ERROR";
                        }

                } else {
                        IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                        outputData = "ERROR";
                }

        } else if (inputData.charAt(0) == 'Q') {

                /* query πιστοποίησης */
                Packet p = new Packet(inputData);

                String[] sess_data = getSession(session_key);
```

```
                        if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) {
                                if (p.getMessage().equals("TOTALVOTED")) {
                                        int total = DBase.usersVoted();
                                        outputData = new
Packet('Q',Integer.toString(total),session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);
                                        setSession(session_key,sess_data[0],sess_data[1],(new
Integer(sess_data[2]).intValue()+2));
                                        IBBPMAServer.NORM.info("Total users voted calculated
and sent.");
                                } else {
                                        IBBPMAServer.NORM.warn("Unrecognized verification
query.");
                                        outputData = "ERROR";
                                }
                        } else {
                                IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                                outputData = "ERROR";
                        }

                } else if (inputData.charAt(0) == 'S') {

                        /* Ορισμός ως ΨΗΦΗΣΕ */
                        Packet p = new Packet(inputData);

                        String[] sess_data = getSession(session_key);

                        if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) {
                                String[] pp = p.splitMessage();
                                String check = pp[0];
                                String name = pp[1];
                                String code = pp[2];
                                String pword = pp[3];

                                if (check.equals("")|check.equals(" ")) {
                                        IBBPMAServer.NORM.warn("No data in packet");
                                        outputData = "ERROR";
                                } else {
                                        // Αρχίζει ο έλεγχος ασφάλειας

                                        // Αποθήκευση των time stamp, AuthKey και ER data
packet
                                        int checkIndex = setCheck(check);
                                        setUsers(checkIndex, name+"-"+code+"-"+pword);
                                        outputData = new
Packet('C',"STAMP",session_key,"",sess_data[1],(new
Integer(sess_data[2]).intValue()+2)).toCryptString(false);
                                        setSession(session_key,sess_data[0],sess_data[1],(new
Integer(sess_data[2]).intValue()+2));
                                }
                        } else {
                                IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                                outputData = "ERROR";
                        }

                } else if (inputData.charAt(0) == 'X') {

                        /* Τέλος επικοινωνίας*/
```

```
                        Packet p = new Packet(inputData);

                        String[] sess_data = getSession(session_key);

                        if (AuthSys.checkHMAC(p,sess_data[0],new
Integer(sess_data[2]).intValue()+1)) {
                                outputData = "DONE";
                        } else {
                                IBBPMAServer.NORM.warn("Packet corrupted or altered!");
                                outputData = "ERROR";
                        }

                } else {
                                IBBPMAServer.NORM.warn("Unrecognized packet received.");
                                outputData = "ERROR";
                }

                return outputData;
-------------------------------------------------------------------------------------------------------
        } //EOF process

} //EOF Class
```

x. ScreenAppender.java

```
package IBBPMAServer;

import org.apache.log4j.*;
import org.apache.log4j.spi.*;

/**
 * Υλοποιεί τη <code>org.apache.log4j.Appender</code> διεπαφή για να
 * παρέχει έξοδο για
 * τους Administrators στο logging.
 *
 */

public class ScreenAppender implements Appender {

	private String name;

	public void doAppend(LoggingEvent le) {

		if (le.priority.isGreaterOrEqual(Priority.WARN)) {  // Εάν υπάρξει το WARN ή
χειρότερο
			IBBPMAServer.frame.showError("[" + le.priority.toString() + "] " +
le.message);
		} else {  // αλλιώς INFO ή DEBUG
			IBBPMAServer.frame.showInfo("[" + le.priority.toString() + "] " +
le.message);
		}

	} //EOF doAppend


	public void addFilter(Filter newFilter) {

		// Δεν χρειάζεται υποστήριξη Φίλτρου

	}


	public void clearFilters() {

		// Όχι Φίλτρα

	}


	public void close() {

		// Ο ScreenAppender δεν θα πρέπει ποτέ να κλείσει
		// εκτός έγκυρου τερματισμού.

	}


	public boolean requiresLayout()	{


		return false;

	}
```

```
        public void setErrorHandler(ErrorHandler errorHandler) {
                // Δεν υποστηρίζεται
        }

        public void setLayout(Layout l) {

        }

        public void activateOptions() {

        }

        public String[] getOptionStrings() {
                String [] s = new String[1];
                s[0] = name;
                return s;
        }

        public void setOption(String option, String value) {
                name = value;

        }

        public String getName() {
                return name;
        }

        public void setName(String n) {
                name = n;
        }
} //EOF ScreenAppender
```

xi. SecureAppender.java

```
package IBBPMAServer;

import org.apache.log4j.*;
import org.apache.log4j.spi.*;

import java.io.*;

import IBBPMA.util.*;

/**
 * Υλοποιεί τη διεπαφή <code>org.apache.log4j.Appender</code> για να
 * προσφέρει μια ασφαλή
 * προσθήκη στο IBB/PMA logging
 */

public class SecureAppender implements Appender {

        private String previous_string;
        private String name;

        public SecureAppender(String previous_val) {

                super();
                previous_string = previous_val; // prime previous_string val

        }

        public void doAppend(LoggingEvent le) {
                try {
                        FileWriter out = new FileWriter("IBBPMAServer.sec.log", true);

                        out.write(AuthSys.makeDigest(le.message + previous_string) + "\r\n");
                        out.close();

                        previous_string = le.message;
                } catch (Exception e) {
                }
        } //EOF doAppend


        public void addFilter(Filter newFilter) {

                // Δεν χρειάζεται υποστήριξη φίλτρων

        }


        public void clearFilters() {

                //Όχι φίλτρα

        }


        public void close() {


        }
```

```
public boolean requiresLayout()     {
        return false;
}

public void setErrorHandler(ErrorHandler errorHandler) {

}

public void setLayout(Layout l) {

}

public void activateOptions() {

}

public String[] getOptionStrings() {
        String [] s = new String[1];
        s[0] = name;
        return s;
}

public void setOption(String option, String value) {
        name = value;

}

public String getName() {
        return name;
}

public void setName(String n) {
        name = n;
}
} //EOF ScreenAppender
```

xii. TCPServer.java

```
package IBBPMAServer;

import java.net.*;
import java.io.*;
import java.util.*;

import IBBPMA.util.*;

/**
 * Η κλάση TCPServer ανοίγει μια υποδοχή για να ακούει για συνδέσεις
 * πελάτη
 * οι οποίες όταν παραλαμβάνονται στέλνονται σε ένα νήμα για να
 * κρατηθεί η υποδοχή ανοιχτή
 * για τους νέους πελάτες
 *
 * Η κλάση είναι από το αρχικό "Java Threads" του O'Reilly.
 */

public class TCPServer implements Cloneable, Runnable {

        Thread runner = null;
        ServerSocket server = null;
        Socket data = null;
        boolean shouldStop = false;

        /**
         * Ο startServer δημιουργεί και εκτελεί ένα νέο νήμα για την
    * συνεδρία
         *
         * @param port  Ένας αριθμός που δείχνει την θήρα που
    * χρησιμοποιείται για την συνεδρία
         */

        protected synchronized void startServer(int port) throws IOException {

                if (runner == null) {
                        server = new ServerSocket(port);
                        runner = new Thread(this);
                        runner.start();
                }
        }


        /**
         * Το stopServer σταματάει τον εξυπηρετητή
         */
        protected synchronized void stopServer() {

                if (server != null) {
                        shouldStop = true;
                        runner.interrupt();
                        runner = null;
                        try {
                                server.close();
                        } catch (IOException ioe) {
                                IBBPMAServer.NORM.error("IO Error closing server
connection");
                        }
```

```
                server = null;
            }
        }

        /**
         * Η run ακούει και δέχεται νέες συνδέσεις και τις στέλνει σε
    * νήματα εκτέλεσης
         *
         */
        public void run() {

            if (server != null) {
                while (!shouldStop) {
                    try {
                        Socket datasocket = server.accept();
                        TCPServer newSocket = (TCPServer) clone();

                        newSocket.server = null;
                        newSocket.data = datasocket;
                        newSocket.runner = new Thread(newSocket);
                        newSocket.runner.start();
                    } catch (Exception e) {
                        IBBPMAServer.NORM.error("Error opening server
connection.");
                    }
                }
            } else {
                run(data);
            }
        }

        protected void run(Socket data) {

            try {

                BufferedReader in = new BufferedReader(new
InputStreamReader(data.getInputStream()));
                PrintWriter out = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(data.getOutputStream())), true);

                String inputData, outputData;

                while (true) {

                    if (in.ready()) {
                        inputData = in.readLine();

                        if (!inputData.equals("")) {

                            String[] tempiData = new String[] {"",""};
                            tempiData =
RSAEncrypt.splitnDecrypt(IBBPMAServer.er_private_key,inputData);
                            inputData = tempiData[1];

                            if(IBBPMAServer.DEV.isDebugEnabled()) {
                                IBBPMAServer.DEV.debug("Received:
" + inputData + " from " + data.toString());
                            }
                            outputData =
IBBPMAServerProtocol.process(inputData, tempiData[0]);
```

```
                                                    if
((outputData.equals("DONE"))|(outputData.equals("ERROR"))) {
                                                            break;
                                                    } else {
                                                            out.println(outputData);
                                                    }
                                            }
                                    }
                            }
                            IBBPMAServer.NORM.info("Closing connection");
                            out.close();
                            in.close();
                            data.close();

                    } catch (Exception e) {
                            IBBPMAServer.NORM.error("Error with connection: " + e.getMessage());
                    }

            } //EOF run

} //EOF TCPServer
```

C. Πακέτο Πελάτη IBB/PMA

i. CFrame2.java

```
package IBBPMAClient;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Αρχική είσοδος περιγραφής για το IBB/PMA σύστημα
 *
 */

public class FCFrame2 extends Frame
{

        Label jLabel1 = new Label();
   TextView tv;
        ScrollView sv;
        Font f;
        Font f2;
        TextStyle normal;
        TextStyle red;
        TextStyle blue;
        TextStyle green;
        Button jButton1 = new Button();

        protected FCFrame2() {


                tv = new TextView(true);
     sv = new ScrollView(tv);
     f = new Font("Courier", Font.PLAIN, 12);
     f2 = new Font("Courier", Font.BOLD, 12);
     normal = new TextStyle(f, Color.black);
     red = new TextStyle(f2, Color.red);
     blue = new TextStyle(f, Color.blue);
     green = new TextStyle(f, Color.green);


                setTitle(IBBPMAClient.messages.getString("intro_title"));
                jLabel1.setText(IBBPMAClient.messages.getString("intro"));
                jButton1.setLabel(IBBPMAClient.messages.getString("ok_button"));

                IBBPMAPanel scrollPanel = new
IBBPMAPanel(IBBPMAClient.messages.getString("intro"));
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, sv);
                IBBPMAPanel buttonPane = new IBBPMAPanel();
                scrollPanel.addComponent( 1, 1, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton1);

                setSize(new java.awt.Dimension(350, 450));
```

```
            add(scrollPanel);

        }

        protected void initComponents() throws Exception
        {
            jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                    jButton1ActionPerformed(e);
                }
            });
            addWindowListener(new java.awt.event.WindowAdapter() {
                public void windowOpened(java.awt.event.WindowEvent e) {
                    thisWindowOpened(e);
                }
                public void windowClosing(java.awt.event.WindowEvent e) {
                    thisWindowClosing(e);
                }
            });
        }

        // Δείχνει ένα πληροφοριακό μήνυμα (black, plain text)
        protected void showInfo(String msg) {
            showMsg(msg);
        }

        // Δείχνει ένα μήνυμα προειδοποίησης (red, bold text)
        protected void showError(String msg) {
            msg += "\n";
            tv.append(msg, red);
        }

        // Δείχνει ένα μήνυμα με τα προκαθορισμένα AttributeSet
        protected void showMsg(String msg) {
            msg += "\n";
            tv.append(msg, normal);
        }

        // Κλείνει το παράθυρο όταν επιλεγεί το close
        void thisWindowClosing(java.awt.event.WindowEvent e)
        {
            setVisible(false);
            dispose();

            IBBPMAClient.frame3.setVisible(true);

        }

        protected void thisWindowOpened(java.awt.event.WindowEvent e) {
            tv.append(IBBPMAClient.messages.getString("welcome"), normal);
        }

        protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {
            setVisible(false);
            IBBPMAClient.ercheck = true;
            IBBPMAClient.frame3.setVisible(true);
            dispose();
        }

}
```

ii. CFrame2.jfrm

```
Metrowerks RAD Data V 1
BeginClasses
  {9BE3D9E0-166E-11D1-B2D8-0060081C5489} "#base-object-class"
  {24A693F8-14E5-11D2-B334-00600819ADE3} "IBBPMAClient"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame1"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JTextPane"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame2"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JScrollPane"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JButton"
EndClasses
BeginObject IBBPMAClient.FCFrame2 "FCFrame2"
  BeginProperties
    enabled = True
    JMenuBar = <none>
    foreground = black
    location = "40, 40"
    title = "IBBPMA: Introduction"
    resizable = False
    background = "204, 204, 204"
    font = Application-PLAIN-10
    layout = None
    cursor = Default
    menuBar = <none>
    size = "346, 258"
  EndProperties
  BeginInternalProperties
    UseSwing 11 = 1
    IsApplet 11 = 0
  EndInternalProperties
  BeginEventConnections
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame2"
      SinkPath = "\\\\IBBPMAClient.FCFrame2"
      EventName = windowClosing
      EventSetName = window
      HandlerName = thisWindowClosing
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame2"
      SinkPath = "\\\\IBBPMAClient.FCFrame2"
      EventName = windowOpened
      EventSetName = window
      HandlerName = thisWindowOpened
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame2\\jButton1"
      SinkPath = "\\\\IBBPMAClient.FCFrame2"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton1ActionPerformed
    EndEventConnection
  EndEventConnections
  BeginObject jScrollPane1 "javax.swing.JScrollPane"
    BeginProperties
      requestFocusEnabled = True
      maximumSize = "32767, 32767"
```

```
      enabled = True
      foreground = black
      location = "10, 10"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Dialog-PLAIN-12
      verticalScrollBarPolicy = 20
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = False
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "22, 22"
      size = "330, 210"
      doubleBuffered = False
      horizontalScrollBarPolicy = 30
    EndProperties
    BeginInternalProperties
    EndInternalProperties
    BeginObject jTextPane1 "javax.swing.JTextPane"
      BeginProperties
        requestFocusEnabled = True
        contentType = text/plain
        selectionStart = 0
        maximumSize = "2147483647, 2147483647"
        enabled = True
        text = ""
        selectedTextColor = black
        foreground = black
        location = "0, 0"
        autoscrolls = True
        visible = True
        background = white
        selectionColor = "204, 204, 255"
        caretColor = black
        font = Application-PLAIN-10
        cursor = Default
        alignmentY = 0.5
        alignmentX = 0.5
        opaque = True
        caretPosition = 0
        selectionEnd = 0
        disabledTextColor = "153, 153, 153"
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "11, 6"
        size = "327, 207"
        doubleBuffered = False
        editable = True
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
  EndObject
  BeginObject jButton1 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
```

```
        maximumSize = "35, 11"
        enabled = True
        text = OK
        actionCommand = ""
        contentAreaFilled = True
        location = "270, 230"
        foreground = black
        autoscrolls = False
        visible = True
        rolloverEnabled = False
        background = "204, 204, 204"
        borderPainted = True
        font = Application-BOLD-10
        horizontalAlignment = 0
        cursor = Default
        alignmentY = 0.0
        alignmentX = 0.0
        opaque = True
        defaultCapable = True
        selected = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = "Click to go to user authentication"
        minimumSize = "35, 11"
        size = "70, 20"
        focusPainted = True
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
EndObject
```

iii. CFrame3.java

```
package IBBPMAClient;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Το FCFrame3 είναι μια οθόνη που προτρέπει τους χρήστες να εισάγουν
 * τις πληροφορίες τους για την πιστοποίηση. *
 */

public class FCFrame3 extends Frame
{
        Label jLabel1 = new Label();
        Label jLabel2 = new Label();
        Label jLabel3 = new Label();
        Label jLabel4 = new Label();
        TextField jTextField1 = new TextField();
        TextField jTextField2 = new TextField();
        TextField jPasswordField1 = new TextField();
        Button jButton1 = new Button();
        Label jLabel5 = new Label();

        protected FCFrame3() {

                setTitle(IBBPMAClient.messages.getString("auth_title"));
                jLabel4.setText(IBBPMAClient.messages.getString("auth_instruct"));
                jLabel1.setText(IBBPMAClient.messages.getString("name"));
                jLabel2.setText(IBBPMAClient.messages.getString("code"));
                jButton1.setLabel(IBBPMAClient.messages.getString("ok_button"));
                jLabel3.setText(IBBPMAClient.messages.getString("password"));
                jLabel5.setText(IBBPMAClient.messages.getString("fill_all"));
                jPasswordField1.setEchoChar('*');

                IBBPMAPanel scrollPanel = new
IBBPMAPanel(IBBPMAClient.messages.getString("auth"));
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel4);
                scrollPanel.addComponent( 1, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel1);
                scrollPanel.addComponent( 2, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField1);
                scrollPanel.addComponent( 3, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel2);
                scrollPanel.addComponent( 4, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jTextField2);
                scrollPanel.addComponent( 5, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel3);
                scrollPanel.addComponent( 6, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jPasswordField1);
                scrollPanel.addComponent( 7, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jLabel5);
                scrollPanel.addComponent( 7, 1, 1, 1, 0, 0, GridBagConstraints.NONE,
```

```
                1.0f, 0.0f, jButton1);
        add(scrollPanel);
        setSize(new java.awt.Dimension(350, 300));
        setResizable( false);
    }

    protected void initComponents() throws Exception
    {

        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                jButton1ActionPerformed(e);
            }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                thisWindowClosing(e);
            }
        });
    }

    // Close the window when the close box is clicked
    void thisWindowClosing(java.awt.event.WindowEvent e)
    {

    }

    protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {

        if
(jTextField1.getText().equals("")|jTextField2.getText().equals("")|jPasswordField1.getText().equals("")
) {
            jLabel5.setVisible(true);
        } else if
(!IBBPMAClient.isSafe(jTextField1.getText())|!IBBPMAClient.isSafe(jTextField2.getText())|!IBBPM
AClient.isSafe(jPasswordField1.getText())) {
            jLabel5.setText(IBBPMAClient.messages.getString("invalid_chars"));
            jLabel5.setVisible(true);
        } else {
            jLabel5.setText("");
            jLabel5.setVisible(false);
            setVisible(false);
            IBBPMAClient.frame4.setVisible(true);

    IBBPMAClient.frame4.showInfo(IBBPMAClient.messages.getString("auth_details"));
            IBBPMAClient.comms.sendAuth(jTextField1.getText(),
jTextField2.getText(), jPasswordField1.getText());
        }

    }

    protected void cleanUp() {
        dispose();
    }


}
```

iv. CFrame3.jfrm

```
Metrowerks RAD Data V 1
BeginClasses
  {9BE3D9E0-166E-11D1-B2D8-0060081C5489} "#base-object-class"
  {24A693F8-14E5-11D2-B334-00600819ADE3} "IBBPMAClient"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame1"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JTextPane"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame2"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JScrollPane"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JButton"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame3"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JOptionPane"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JPanel"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JLabel"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JTextField"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JToggleButton"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JPasswordField"
EndClasses
BeginObject IBBPMAClient.FCFrame3 "FCFrame3"
  BeginProperties
    enabled = True
    JMenuBar = <none>
    foreground = black
    location = "0, 0"
    title = "IBBPMA: User Authentication"
    resizable = False
    background = "204, 204, 204"
    font = Application-PLAIN-10
    layout = None
    cursor = Default
    menuBar = <none>
    size = "312, 156"
  EndProperties
  BeginInternalProperties
    UseSwing 11 = 1
    IsApplet 11 = 0
  EndInternalProperties
  BeginEventConnections
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame3"
      SinkPath = "\\\\IBBPMAClient.FCFrame3"
      EventName = windowClosing
      EventSetName = window
      HandlerName = thisWindowClosing
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame3\\jButton1"
      SinkPath = "\\\\IBBPMAClient.FCFrame3"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton1ActionPerformed
    EndEventConnection
  EndEventConnections
  BeginObject jLabel1 "javax.swing.JLabel"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      iconTextGap = 4
```

```
      maximumSize = "0, 0"
      enabled = True
      text = Name:
      foreground = "102, 102, 153"
      location = "10, 40"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Application-BOLD-10
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.0
      opaque = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "0, 0"
      size = "80, 20"
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jLabel2 "javax.swing.JLabel"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      iconTextGap = 4
      maximumSize = "0, 0"
      enabled = True
      text = Code:
      foreground = "102, 102, 153"
      location = "10, 70"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Application-BOLD-10
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.0
      opaque = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "0, 0"
      size = "80, 20"
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jLabel3 "javax.swing.JLabel"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      iconTextGap = 4
```

```
      maximumSize = "0, 0"
      enabled = True
      text = Password:
      foreground = "102, 102, 153"
      location = "10, 100"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Application-BOLD-10
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.0
      opaque = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "0, 0"
      size = "80, 20"
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jLabel4 "javax.swing.JLabel"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      iconTextGap = 4
      maximumSize = "0, 0"
      enabled = True
      text = "Using your voter card, fill out the fields:"
      foreground = "102, 102, 153"
      location = "10, 10"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Application-BOLD-10
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.0
      opaque = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "0, 0"
      size = "260, 20"
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jTextField1 "javax.swing.JTextField"
    BeginProperties
      requestFocusEnabled = True
      selectionStart = 0
      maximumSize = "2147483647, 2147483647"
```

```
      enabled = True
      scrollOffset = 0
      text = ""
      selectedTextColor = black
      location = "100, 40"
      foreground = black
      autoscrolls = True
      visible = True
      background = white
      selectionColor = "204, 204, 255"
      caretColor = black
      font = Application-PLAIN-10
      columns = 0
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = True
      caretPosition = 0
      selectionEnd = 0
      disabledTextColor = "153, 153, 153"
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "4, 17"
      size = "200, 20"
      doubleBuffered = False
      editable = True
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jTextField2 "javax.swing.JTextField"
    BeginProperties
      requestFocusEnabled = True
      selectionStart = 0
      maximumSize = "2147483647, 2147483647"
      enabled = True
      scrollOffset = 0
      text = ""
      selectedTextColor = black
      location = "100, 70"
      foreground = black
      autoscrolls = True
      visible = True
      background = white
      selectionColor = "204, 204, 255"
      caretColor = black
      font = Application-PLAIN-10
      columns = 0
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = True
      caretPosition = 0
      selectionEnd = 0
      disabledTextColor = "153, 153, 153"
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "4, 17"
```

```
      size = "200, 20"
      doubleBuffered = False
      editable = True
    EndProperties
    BeginInternalProperties
    EndInternalProperties
EndObject
BeginObject jPasswordField1 "javax.swing.JPasswordField"
    BeginProperties
      requestFocusEnabled = True
      selectionStart = 0
      maximumSize = "2147483647, 2147483647"
      enabled = True
      scrollOffset = 0
      text = ""
      selectedTextColor = black
      foreground = black
      location = "100, 100"
      autoscrolls = True
      visible = True
      background = white
      selectionColor = "204, 204, 255"
      caretColor = black
      font = Application-PLAIN-10
      columns = 0
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = True
      caretPosition = 0
      selectionEnd = 0
      disabledTextColor = "153, 153, 153"
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "4, 17"
      size = "200, 20"
      doubleBuffered = False
      editable = True
    EndProperties
    BeginInternalProperties
    EndInternalProperties
EndObject
BeginObject jButton1 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = OK
      actionCommand = ""
      contentAreaFilled = True
      location = "230, 130"
      foreground = black
      autoscrolls = False
      visible = True
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
```

```
        horizontalAlignment = 0
        cursor = Default
        alignmentY = 0.0
        alignmentX = 0.0
        opaque = True
        defaultCapable = True
        selected = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "35, 11"
        size = "70, 20"
        focusPainted = True
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
    BeginObject jLabel5 "javax.swing.JLabel"
      BeginProperties
        requestFocusEnabled = True
        verticalAlignment = 0
        iconTextGap = 4
        maximumSize = "0, 0"
        enabled = True
        text = "Please fill out ALL the fields!"
        foreground = "102, 102, 153"
        location = "10, 130"
        autoscrolls = False
        visible = False
        background = "204, 204, 204"
        font = Application-BOLD-10
        horizontalAlignment = 2
        cursor = Default
        alignmentY = 0.5
        alignmentX = 0.0
        opaque = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "0, 0"
        size = "210, 20"
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
EndObject
```

v. CFrame4.java

```
package IBBPMAClient;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Μια οθόνη που δείχνει την κατάσταση της πιστοποίησης του ψηφοφόρου.
 *
 */

public class FCFrame4 extends Frame
{
        protected static boolean status;

  TextView tv;
        ScrollView sv;
        Font f;
        Font f2;
        TextStyle normal;
        TextStyle red;
        TextStyle blue;
        TextStyle green;
        Button jButton1 = new Button();
        Button jButton2 = new Button();

        protected FCFrame4() {

                // Ετοιμασία textView
                tv = new TextView(true);
    sv = new ScrollView(tv);
    f = new Font("Courier", Font.PLAIN, 12);
    f2 = new Font("Courier", Font.BOLD, 12);
    normal = new TextStyle(f, Color.black);
    red = new TextStyle(f2, Color.red);
    blue = new TextStyle(f, Color.blue);
    green = new TextStyle(f, Color.green);

                // Δημιουργία του frame
                jButton1.setLabel(IBBPMAClient.messages.getString("ok_button"));
                jButton2.setLabel(IBBPMAClient.messages.getString("quit_button"));
                setTitle(IBBPMAClient.messages.getString("auth_title"));

                IBBPMAPanel scrollPanel = new IBBPMAPanel(" ");
                scrollPanel.setInsets(3, 3, 3, 3);

                scrollPanel.addComponent( 0, 0, 1, 2, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, sv);
                scrollPanel.addComponent( 2, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton1);
                scrollPanel.addComponent( 2, 1, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton2);

                add(scrollPanel, BorderLayout.CENTER);
                setSize(new java.awt.Dimension(350, 300));
```

```
}

protected void initComponents() throws Exception
{

        jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton1ActionPerformed(e);
                }
        });
        jButton2.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton2ActionPerformed(e);
                }
        });
        addWindowListener(new java.awt.event.WindowAdapter() {
                public void windowClosing(java.awt.event.WindowEvent e) {
                        thisWindowClosing(e);
                }
        });

        jButton1.setEnabled(false);
        jButton2.setEnabled(false);
}

// Εμφάνιση μηνύματος πληροφορίας(black, plain text)
protected void showInfo(String msg) {
        showMsg(msg);
}

// Εμφάνιση προειδοποιητικού μηνύματος (red, bold text)
protected void showError(String msg) {
        msg += "\n";
        tv.append(msg, red);
}

// Εμφάνιση κειμένου με συγκεκριμένο AttributeSet
protected void showMsg(String msg) {
        msg += "\n";
        tv.append(msg, normal);
}


// Κλείσιμο του παραθύρου όταν επιλεγεί το Close
void thisWindowClosing(java.awt.event.WindowEvent e)
{
}

protected void jButton2ActionPerformed(java.awt.event.ActionEvent e) {

        IBBPMAClient.frame3.cleanUp();
        setVisible(false);
        dispose();
        System.exit(0);

}

protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {
```

```
                if (status == true) {
                        setVisible(false);
                        IBBPMAClient.frame3.cleanUp();
                        IBBPMAClient.ercheck = false;
                        IBBPMAClient.frame.setVisible(true);
                        dispose();
                } else {
                        setVisible(false);
                        IBBPMAClient.frame4.jButton1.setEnabled(false);
                        IBBPMAClient.frame4.jButton2.setEnabled(false);
                        IBBPMAClient.frame3.setVisible(true);
                }

        }


}
```

vi. CFrame4.jfrm

```
Metrowerks RAD Data V 1
BeginClasses
  {9BE3D9E0-166E-11D1-B2D8-0060081C5489} "#base-object-class"
  {24A693F8-14E5-11D2-B334-00600819ADE3} "IBBPMAClient"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame1"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JTextPane"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame2"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JScrollPane"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JButton"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame3"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JOptionPane"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JPanel"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JLabel"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JTextField"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JToggleButton"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JPasswordField"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame4"
EndClasses
BeginObject IBBPMAClient.FCFrame4 "FCFrame4"
  BeginProperties
    enabled = True
    JMenuBar = <none>
    foreground = black
    location = "0, 0"
    title = "IBB/PMA Authorisation"
    resizable = True
    background = "204, 204, 204"
    font = Application-PLAIN-10
    layout = None
    cursor = Default
    menuBar = <none>
    size = "350, 235"
  EndProperties
  BeginInternalProperties
    UseSwing 11 = 1
    IsApplet 11 = 0
  EndInternalProperties
  BeginEventConnections
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame4"
      SinkPath = "\\\\IBBPMAClient.FCFrame4"
      EventName = windowClosing
      EventSetName = window
      HandlerName = thisWindowClosing
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame4\\jButton2"
      SinkPath = "\\\\IBBPMAClient.FCFrame4"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton2ActionPerformed
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.FCFrame4\\jButton1"
      SinkPath = "\\\\IBBPMAClient.FCFrame4"
      EventName = actionPerformed
      EventSetName = action
```

```
      HandlerName = jButton1ActionPerformed
    EndEventConnection
  EndEventConnections
  BeginObject jTextPane1 "javax.swing.JTextPane"
    BeginProperties
      requestFocusEnabled = True
      contentType = text/plain
      selectionStart = 0
      maximumSize = "2147483647, 2147483647"
      enabled = True
      text = ""
      selectedTextColor = black
      foreground = black
      location = "10, 10"
      autoscrolls = True
      visible = True
      background = white
      selectionColor = "204, 204, 255"
      caretColor = black
      font = Application-PLAIN-10
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = True
      caretPosition = 0
      selectionEnd = 0
      disabledTextColor = "153, 153, 153"
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "11, 6"
      size = "330, 190"
      doubleBuffered = False
      editable = True
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jButton1 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = OK
      actionCommand = ""
      contentAreaFilled = True
      location = "270, 210"
      foreground = black
      autoscrolls = False
      visible = False
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      defaultCapable = True
```

```
        selected = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = "click to continue"
        minimumSize = "35, 11"
        size = "70, 20"
        focusPainted = True
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
    BeginObject jButton2 "javax.swing.JButton"
      BeginProperties
        requestFocusEnabled = True
        verticalAlignment = 0
        maximumSize = "35, 11"
        enabled = True
        text = Quit
        actionCommand = ""
        contentAreaFilled = True
        location = "190, 210"
        foreground = black
        autoscrolls = False
        visible = False
        rolloverEnabled = False
        background = "204, 204, 204"
        borderPainted = True
        font = Application-BOLD-10
        horizontalAlignment = 0
        cursor = Default
        alignmentY = 0.0
        alignmentX = 0.0
        opaque = True
        defaultCapable = True
        selected = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = "click to quit"
        minimumSize = "35, 11"
        size = "70, 20"
        focusPainted = True
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
  EndObject
```

vii. IBBPMAClient.java

```
package IBBPMAClient;

import java.net.*;
import java.io.*;
import java.awt.*;
import java.util.*;

import IBBPMA.awt.*;
import IBBPMA.util.*;

/**
 * Η κλάση IBBPMAClient αρχικοποιεί και εκινεί το λογισμικό IBBPMAClient.
 *
 */

public class IBBPMAClient {

        /* Αλφαριθμητικά που αποθηκεύουν τα δεδομένα για την γλώσσα*/
        protected static String language = "en";
        protected static String country = "GB";

        /** Ορίζει το σύστημα της ψηφοφορίας*/
//#ifdef INSTALL
        protected static final String vote_system = "borda-default";
//#ifdef INSTALL
        protected static final int number_of_choices = 4;
//#ifdef INSTALL
        protected static final String write_in = "no";

        /* Ο χρόνος σε milliseconds, πρωτού τερματιστεί μια σύνδεση */
        protected static final long timeOutTime = 240000;

        /* true όταν υπάρχει σύνδεση με τον IBBPMAServer */
        protected static boolean ercheck;

        /* Μεταβλητή για τον τερματισμό της επικοινωνίας μετά από time out */
        protected static boolean carryOn = true;

        /* Αποθηκεύεται η γλώσσα στην οποία θα εκτελείται το πρόγραμμα */
        protected static Locale currentLocale;

        protected static ResourceBundle messages;

        protected static VoteFrame frame;
        protected static FCFrame2 frame2;
        protected static FCFrame3 frame3;
        protected static FCFrame4 frame4;
        protected static Comms comms;

        /**
         * Το IBBPMAClient() είναι ο constructor που δημιουργεί όλα τα AWT frames και
         * εκκινεί το splash screen.
         *
         */

        protected IBBPMAClient()
        {
```

```
                try {
                        // Εάν επιθυμείται native Look and Feel, πρέπει να τοποθετήσετε τον
ακόλουθο κώδικα στο πρόγραμμα.
                        /*
                        try {

        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
                        }
                        catch (Exception e) {
                        }
                        */

                        /* Εγκατάσταση Internationalisation */
                        currentLocale = new Locale(language, country);
                    messages = ResourceBundle.getBundle("IBBPMAClientLang",currentLocale);

                        /* AWT Frames */
                        frame2 = new FCFrame2();
                        IBBPMAPanel.showSplash(4000);
                        RSAEncrypt.init();

                        frame2.initComponents();
                        frame2.setVisible(true);
                        frame = new VoteFrame();
                        frame.initComponents();
                        frame3 = new FCFrame3();
                        frame3.initComponents();
                        frame4 = new FCFrame4();
                        frame4.initComponents();

                        comms = new Comms();

                }
                catch (Exception e) {
                        e.printStackTrace();
                }
        }

        static public void main(String[] args)
        {
                new IBBPMAClient();
        }

        /**
         * Ελέγχει το String εισόδου για να βεβαιώσει ότι περιέχει μόνο ασφαλείς
    * χαρακτήρες.
         */

        protected static boolean isSafe(String input) {

                boolean ok = true;

                for(int i = 0; (i<input.length())&&(ok==true); i++) {

                        switch (input.charAt(i)) {
                                case 'a': ok = true; break;
                                case 'b': ok = true; break;
                                case 'c': ok = true; break;
                                case 'd': ok = true; break;
```

```
case 'e': ok = true; break;
case 'f': ok = true; break;
case 'g': ok = true; break;
case 'h': ok = true; break;
case 'i': ok = true; break;
case 'j': ok = true; break;
case 'k': ok = true; break;
case 'l': ok = true; break;
case 'm': ok = true; break;
case 'n': ok = true; break;
case 'o': ok = true; break;
case 'p': ok = true; break;
case 'q': ok = true; break;
case 'r': ok = true; break;
case 's': ok = true; break;
case 't': ok = true; break;
case 'u': ok = true; break;
case 'v': ok = true; break;
case 'w': ok = true; break;
case 'x': ok = true; break;
case 'y': ok = true; break;
case 'z': ok = true; break;
case 'A': ok = true; break;
case 'B': ok = true; break;
case 'C': ok = true; break;
case 'D': ok = true; break;
case 'E': ok = true; break;
case 'F': ok = true; break;
case 'G': ok = true; break;
case 'H': ok = true; break;
case 'I': ok = true; break;
case 'J': ok = true; break;
case 'K': ok = true; break;
case 'L': ok = true; break;
case 'M': ok = true; break;
case 'N': ok = true; break;
case 'O': ok = true; break;
case 'P': ok = true; break;
case 'Q': ok = true; break;
case 'R': ok = true; break;
case 'S': ok = true; break;
case 'T': ok = true; break;
case 'U': ok = true; break;
case 'V': ok = true; break;
case 'W': ok = true; break;
case 'X': ok = true; break;
case 'Y': ok = true; break;
case 'Z': ok = true; break;
case '1': ok = true; break;
case '2': ok = true; break;
case '3': ok = true; break;
case '4': ok = true; break;
case '5': ok = true; break;
case '6': ok = true; break;
case '7': ok = true; break;
case '8': ok = true; break;
case '9': ok = true; break;
case '0': ok = true; break;
case ';': ok = true; break;
case ')': ok = true; break;
```

```
                    case '(': ok = true; break;
                    case '*': ok = true; break;
                    case '\': ok = true; break;
                    case ' ': ok = true; break;
                    case '=': ok = true; break;
                    case '_': ok = true; break;
                    case '+': ok = true; break;
                    case '@': ok = true; break;
                    case ',': ok = true; break;
                    case '.': ok = true; break;
                    case '<': ok = true; break;
                    case '>': ok = true; break;
                    case '?': ok = true; break;
                    default:  ok = false; break;
                } //eof case

            } //eof for

            return ok;

        } //eof isSafe

    } //EOF Class
```

viii. TCPClient.java

```
package IBBPMAClient;

import java.io.*;
import java.net.*;

import IBBPMA.util.*;

/**
 * Ο TCPClient ανοίγει συνδέσεις στους εξυπηρετητές και στέλνει / λαμβάνει όλα
 * τα δεδομένα.
 *
 * Ενώ ο TCPClient εκτελεί όλες τις λειτουργίες επιπέδου σύνδεσης και ασχολείται
 * με τα λάθη, όλα τα δεδομένα περνάνε στο <code>ClientProtocol</code> για
 * επεξεργασία
 */

public class TCPClient implements Runnable {

        /**
         * Η run() παραδίδει ένα πακέτο στον εξυπηρετητή του οποίου η DNS
    * διεύθυνση έχει επιλεγεί.
         *
         */

        public void run() {

                String serverName="";
                String packetMsg;
                int port = 0;
                Socket Freesocket = null;
                PrintWriter out = null;
                BufferedReader in = null;

                /* get data */
                if (IBBPMAClient.comms.getType()=='V') {
                        serverName = IBBPMAClient.comms.R_address;
                        port = IBBPMAClient.comms.freeRTPort;
                } else if (IBBPMAClient.comms.getType()=='D') {
                        serverName = IBBPMAClient.comms.R_address;
                        port = IBBPMAClient.comms.freeRTPort;
                } else if (IBBPMAClient.comms.getType()=='E') {
                        serverName = IBBPMAClient.comms.ER_address;
                        port = IBBPMAClient.comms.freePort;
                } else {

        IBBPMAClient.frame.showInfo(IBBPMAClient.messages.getString("unrec_type"));
                        if (IBBPMAClient.ercheck) {

        IBBPMAClient.frame4.showError(IBBPMAClient.messages.getString("no_IO") +
serverName);
                                IBBPMAClient.frame4.status=false;
                                IBBPMAClient.frame4.jButton1.setVisible(true);
                                IBBPMAClient.frame4.jButton2.setVisible(true);

        IBBPMAClient.frame4.showInfo(IBBPMAClient.messages.getString("try_again"));
                        } else {
```

```
        IBBPMAClient.frame.showError(IBBPMAClient.messages.getString("no_IO") +
serverName);
                                IBBPMAClient.frame.voteErr();
                        }
                }

                packetMsg = IBBPMAClient.comms.getMessage() + "\r\n";

                /* Αποστολή πακέτου*/
                try {
                        freeSocket = new Socket(serverName, port); // Άνοιγμα Σύνδεσης
                        out = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(freeSocket.getOutputStream())), true);
                        in = new BufferedReader(new
InputStreamReader(freeSocket.getInputStream()));

                        out.println(packetMsg); // Εγγραφή πακέτου
-------------------------------------------------------------------------------
                } catch (UnknownHostException e) {
                        if (IBBPMAClient.ercheck) {

        IBBPMAClient.frame4.showError(IBBPMAClient.messages.getString("no_host") +
serverName);
                                IBBPMAClient.frame4.status=false;
                                IBBPMAClient.frame4.jButton1.setEnabled(true);
                                IBBPMAClient.frame4.jButton2.setEnabled(true);

        IBBPMAClient.frame4.showInfo(IBBPMAClient.messages.getString("try_again"));
                        } else {
                                IBBPMAClient.frame.showError("Unknown host: " +
serverName);
                                IBBPMAClient.frame.voteErr();
                        }
                } catch (IOException e) {
                        if (IBBPMAClient.ercheck) {

        IBBPMAClient.frame4.showError(IBBPMAClient.messages.getString("no_IO") +
serverName);
                                IBBPMAClient.frame4.status=false;
                                IBBPMAClient.frame4.jButton1.setEnabled(true);
                                IBBPMAClient.frame4.jButton2.setEnabled(true);

        IBBPMAClient.frame4.showInfo(IBBPMAClient.messages.getString("try_again"));
                        } else {

        IBBPMAClient.frame.showError(IBBPMAClient.messages.getString("no_IO") +
serverName);
                                IBBPMAClient.frame.voteErr();
                        }
                } catch (Exception e) {
                        if (IBBPMAClient.ercheck) {

        IBBPMAClient.frame4.showError(IBBPMAClient.messages.getString("err_connect") +
e.getMessage());
                                IBBPMAClient.frame4.status=false;
                                IBBPMAClient.frame4.jButton1.setEnabled(true);
                                IBBPMAClient.frame4.jButton2.setEnabled(true);

        IBBPMAClient.frame4.showInfo(IBBPMAClient.messages.getString("try_again"));
```

```
                    } else {

        IBBPMAClient.frame.showError(IBBPMAClient.messages.getString("err_connect") +
e.getMessage());
                            IBBPMAClient.frame.voteErr();
                    }
            }

            String inputData = "NONE";
            String[] outputData = new String[] {" ","NONE"};

            /* read response */
            try {

                    while (IBBPMAClient.carryOn) {

                            if (in.ready()) {  // if there's data

                                    inputData = in.readLine();

                                    if (!inputData.equals("")) { // Έλεγχος λήψης
πραγματικών στοιχείων
                                            // decrypt data
                                            inputData =
RSAEncrypt.blowDecrypt(IBBPMAClient.comms.getsession_key(), inputData);

                                            // now process
                                            outputData = ClientProtocol.process(inputData);

                                            if
((outputData[1].equals("DONE"))|(outputData[1].equals("ERROR"))) {

                                                    break;

                                            } else if (outputData[0].charAt(0)=='X') {

                                                    out.println(outputData[1]);

                                                    break;

                                            } else {

                                                    out.println(outputData[1]);

                                            }

                                    }

                            }

                    }


            } catch (Exception e) {
                    if (IBBPMAClient.ercheck) {  //Έλεγχος της αποστολής λαθών στην σωστή
οθόνη

        IBBPMAClient.frame4.showError(IBBPMAClient.messages.getString("connect_err") +
e.getMessage());
                    } else {
```

```
        IBBPMAClient.frame.showError(IBBPMAClient.messages.getString("connect_err") +
e.getMessage());
                        }
                } finally {
                        try {
                                out.close();
                                in.close();
                                freeSocket.close();
                        } catch (Exception e) {
                                if (IBBPMAClient.ercheck) {

        IBBPMAClient.frame4.showError(IBBPMAClient.messages.getString("err_shut") +
e.getMessage());
                                } else {

        IBBPMAClient.frame.showError(IBBPMAClient.messages.getString("err_shut") +
e.getMessage());
                                }
                        }
                        IBBPMAClient.comms.timer.stopThread();
                        IBBPMAClient.comms.timer = null;
                        IBBPMAClient.comms.setFromServer(outputData[1]);
                }

        } //EOF run

} //EOF TCPClient
```

ix. TimeOutThread.java

```
package IBBPMAClient;

/**
 * Ένα νήμα που περιμένει ένα δεδομένο χρόνο σε milliseconds που ορίζεται από
 * <code>IBBPMAClient.timeOutTime</code> πρωτού να κλείνει τις επικοινωνίες.
 *
 */

public class TimeOutThread extends Thread {

	public boolean threadStatus = true;

	public void stopThread() {
		threadStatus = false;
	}

	public void run() {

		while (threadStatus) {

			try {
				sleep(IBBPMAClient.timeOutTime);
			} catch (InterruptedException e) {}

			if (threadStatus) {
				IBBPMAClient.carryOn = false;
				if (IBBPMAClient.ercheck) {

	IBBPMAClient.frame4.showError(IBBPMAClient.messages.getString("time_out"));
					IBBPMAClient.frame4.status=false;
					IBBPMAClient.frame4.jButton1.setVisible(true);
					IBBPMAClient.frame4.jButton2.setVisible(true);
					IBBPMAClient.frame4.jButton1.setEnabled(true);
					IBBPMAClient.frame4.jButton2.setEnabled(true);

	IBBPMAClient.frame4.showInfo(IBBPMAClient.messages.getString("try_again"));
				} else {

	IBBPMAClient.frame.showError(IBBPMAClient.messages.getString("time_out"));
					IBBPMAClient.frame.voteErr();
				}
			}

		} //eof while

	} //EOF run()

} //EOF TimeOutThread
```

x. VoteFrame.java

```
package IBBPMAClient;

import java.awt.*;
import java.awt.event.*;

import IBBPMA.awt.*;

/**
 * Η κύρια οθόνη ψηφοφορίας, στην οποία ο ψηφοφόρος κάνει τις επιλογές του.
 */

public class VoteFrame extends Frame {

        private String choice;
        private final CheckboxGroup group = new CheckboxGroup();

        /* Τα ονόματα του υποψηφίου*/
//#ifdef INSTALL
        private final String cand1name = "Mr Triantafillou Basilios, EPDO";
//#ifdef INSTALL
        private final String cand1code = "TRV";
//#ifdef INSTALL
        private final String cand2name = "Mr Alefragis Panagiwtis, EPDO";
//#ifdef INSTALL
        private final String cand2code = "ALP";
//#ifdef INSTALL
        private final String cand3name = "Mr Voros Nickolaos, EPDO";
//#ifdef INSTALL
        private final String cand3code = "VON";
//#ifdef INSTALL
        private final String cand4name = "Mr Illaridis Panagiwtis";
//#ifdef INSTALL
        private final String cand4code = "PIL";

        ScrollPane jScrollPane1 = new ScrollPane();
   TextView tv;
        ScrollView sv;
        Font f;
        Font f2;
        TextStyle normal;
        TextStyle red;
        TextStyle blue;
        TextStyle green;
        boolean clicked;
        Label jLabel1 = new Label();

        Button jButton1 = new Button();
        Button jButton2 = new Button();
        Label jLabel2 = new Label();
        Button jButton3 = new Button();
        Button jButton4 = new Button();

        /* Συγκεκριμένες μεταβλητές για το σύστημα ψηφοφορίας*/
        Checkbox jToggleButton1 = new Checkbox(); // fptp-default
        Checkbox jToggleButton2 = new Checkbox();
        Checkbox jToggleButton3 = new Checkbox();
```

```
        Checkbox jToggleButton4 = new Checkbox();
        TextField jTextField1 = new TextField(); // borda-default
        TextField jTextField2 = new TextField();
        TextField jTextField3 = new TextField();
        TextField jTextField4 = new TextField();
        Label bordaLabel1 = new Label();
        Label bordaLabel2 = new Label();
        Label bordaLabel3 = new Label();
        Label bordaLabel4 = new Label();

        public Insets getInsets(){ return new Insets(30, 10, 10, 10);}

        protected VoteFrame() {

                int FrameHeight = 0;


                tv = new TextView(true);
        sv = new ScrollView(tv);
        f = new Font("Courier", Font.PLAIN, 12);
        f2 = new Font("Courier", Font.BOLD, 12);
        normal = new TextStyle(f, Color.black);
        red = new TextStyle(f2, Color.red);
        blue = new TextStyle(f, Color.blue);
        green = new TextStyle(f, Color.green);

                jLabel1.setText(IBBPMAClient.messages.getString("make_choice"));
                IBBPMAPanel buttonPane = new IBBPMAPanel();
                choice = "";

                if (IBBPMAClient.vote_system.equals("fptp-default")) {  // First Past the Post default
init code

                        // build frame
                        jToggleButton1.setCheckboxGroup(group);
                        jToggleButton2.setCheckboxGroup(group);
                        jToggleButton3.setCheckboxGroup(group);
                        jToggleButton4.setCheckboxGroup(group);

                        jToggleButton1.setLabel(cand1name);
                        jToggleButton2.setLabel(cand2name);
                        jToggleButton3.setLabel(cand3name);
                        jToggleButton4.setLabel(cand4name);

                        buttonPane.setSize(100, 50);
                        buttonPane.addComponent( 0, 0, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jToggleButton1);
                        buttonPane.addComponent( 0, 1, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jToggleButton2);
                        buttonPane.addComponent( 0, 2, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jToggleButton3);
                        buttonPane.addComponent( 0, 3, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jToggleButton4);

                        FrameHeight = 300;
```

```
                } else if (IBBPMAClient.vote_system.equals("borda-default")) { // Borda default init
code

                        clicked = false;

                        bordaLabel1.setText(cand1name);
                        bordaLabel2.setText(cand2name);
                        bordaLabel3.setText(cand3name);
                        bordaLabel4.setText(cand4name);

                        jTextField1.setColumns(5);
                        jTextField2.setColumns(5);
                        jTextField3.setColumns(5);
                        jTextField4.setColumns(5);

                        buttonPane.setSize(100, 100);
                        buttonPane.addComponent( 0, 0, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, bordaLabel1);
                        buttonPane.addComponent( 0, 1, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, bordaLabel2);
                        buttonPane.addComponent( 0, 2, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, bordaLabel3);
                        buttonPane.addComponent( 0, 3, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, bordaLabel4);
                        buttonPane.addComponent( 1, 0, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jTextField1);
                        buttonPane.addComponent( 1, 1, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jTextField2);
                        buttonPane.addComponent( 1, 2, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jTextField3);
                        buttonPane.addComponent( 1, 3, 1, 1, 0, 0,
GridBagConstraints.HORIZONTAL,
                                1.0f, 0.0f, jTextField4);

                        FrameHeight = 400;

                } else {
                        System.out.println("FATAL ERROR: unrecognised vote system defined.");
                } // EOF Vote_System if

                jLabel2.setText(IBBPMAClient.messages.getString("sure"));
                jButton1.setLabel(IBBPMAClient.messages.getString("yes_button"));
                jButton2.setLabel(IBBPMAClient.messages.getString("no_button"));

                jButton3.setLabel(IBBPMAClient.messages.getString("retry_button"));
                jButton4.setLabel(IBBPMAClient.messages.getString("quit_button"));

                IBBPMAPanel scrollPanel = new
IBBPMAPanel(IBBPMAClient.messages.getString("vote_stat"));
                scrollPanel.setInsets(3, 3, 3, 3);
                scrollPanel.addComponent( 0, 0, 1, 4, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 1.0f, sv);
                scrollPanel.setInsets(1, 0, 1, 0);
```

```
                scrollPanel.addComponent( 1, 0, 1, 4, 0, 0, GridBagConstraints.HORIZONTAL,
                        1.0f, 0.0f, buttonPane);
                scrollPanel.setInsets(3, 3, 3, 3);
                scrollPanel.addComponent( 2, 0, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton1);
                scrollPanel.addComponent( 2, 1, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton2);
                scrollPanel.addComponent( 2, 2, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton3);
                scrollPanel.addComponent( 2, 3, 1, 1, 0, 0, GridBagConstraints.BOTH,
                        1.0f, 0.0f, jButton4);

                add(scrollPanel);
                setTitle(IBBPMAClient.messages.getString("vote_title"));
                setSize(new java.awt.Dimension(460, FrameHeight));

        } //eof VoteFrame()

        class focusListener extends FocusAdapter {

                Color t_col = getBackground();

                public void focusGained( FocusEvent e){
                        ((Checkbox)e.getSource()).setBackground(Color.gray);
                }

                public void focusLost( FocusEvent e){
                        ((Checkbox)e.getSource()).setBackground(t_col);
                }

        } // eof focusListener


        class itemListener implements ItemListener{

                public void itemStateChanged(ItemEvent e){

                        if(e.getSource() == jToggleButton1){
                                jToggleButton1StateChanged(e);
                        } else if(e.getSource() == jToggleButton2){
                                jToggleButton2StateChanged(e);
                        } else if(e.getSource() == jToggleButton3){
                                jToggleButton3StateChanged(e);
                        } else if(e.getSource() == jToggleButton4){
                                jToggleButton4StateChanged(e);
                        }
                }
        } //eof itemListener

        public void initComponents() throws Exception {

                if (IBBPMAClient.vote_system.equals("fptp-default")) { // First Past the Post default
init code
                        jToggleButton1.addItemListener(new itemListener());
                        jToggleButton1.addFocusListener(new focusListener());
                        jToggleButton2.addItemListener(new itemListener());
                        jToggleButton2.addFocusListener(new focusListener());
                        jToggleButton3.addItemListener(new itemListener());
                        jToggleButton3.addFocusListener(new focusListener());
```

```
                jToggleButton4.addItemListener(new itemListener());
                jToggleButton4.addFocusListener(new focusListener());

                jButton1.setEnabled(false);
                jButton2.setEnabled(false);
                jButton3.setEnabled(false);
                jButton4.setEnabled(true);
                jLabel2.setVisible(false);

        } else if (IBBPMAClient.vote_system.equals("borda-default")) { // Borda default init
code
                jButton1.setEnabled(true);
                jButton2.setEnabled(false);
                jButton3.setEnabled(false);
                jButton4.setEnabled(true);
                jLabel2.setVisible(false);
        }

        jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton1ActionPerformed(e);
                }
        });

        jButton2.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton2ActionPerformed(e);
                }
        });

        jButton3.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton3ActionPerformed(e);
                }
        });

        jButton4.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                        jButton4ActionPerformed(e);
                }
        });

        addWindowListener(new java.awt.event.WindowAdapter() {
                public void windowClosing(java.awt.event.WindowEvent e) {
                        thisWindowClosing(e);
                }
        });

} //eof initComponents()


// Εμφάνιση πληροφοριακού μηνύματος (black, plain text)
protected void showInfo(String msg) {
        showMsg(msg);
}

// Εμφάνισης προηδοποιητικού μηνύματος (red, bold text)
protected void showError(String msg) {
        msg += "\n";
```

```
            tv.append(msg, red);
}

// Εμφάνιση μηνύματος κειμένου με προκαθορισμένο AttributeSet

protected void showMsg(String msg) {
            msg += "\n";
            tv.append(msg, normal);
}

// Κλείσιμο του παραθύρου με την επιλογή του close box
void thisWindowClosing(java.awt.event.WindowEvent e)

{

}

/**
 * Εάν το <code>IBBPMAClient.number_of_choices</code> είναι 4 τότε το bordaSum
 * επιστρέφει 10
 */

private int bordaSum() {

            int res=0;

            for (int x = 1; x <= IBBPMAClient.number_of_choices; x++) {
                        res += x;
            }

            return res;

} //EOF bordaSum

protected void jToggleButton1StateChanged(ItemEvent e) {
            choice = cand1code;
            jLabel2.setVisible(true);
            showInfo(IBBPMAClient.messages.getString("sure"));
            jButton1.setEnabled(true);
            jButton2.setEnabled(true);
}

protected void jToggleButton2StateChanged(ItemEvent e) {
            choice = cand2code;
            jLabel2.setVisible(true);
            showInfo(IBBPMAClient.messages.getString("sure"));
            jButton1.setEnabled(true);
            jButton2.setEnabled(true);
}

protected void jToggleButton3StateChanged(ItemEvent e) {
            choice = cand3code;
            jLabel2.setVisible(true);
            showInfo(IBBPMAClient.messages.getString("sure"));
            jButton1.setEnabled(true);
            jButton2.setEnabled(true);
}

protected void jToggleButton4StateChanged(ItemEvent e) {
            choice = cand4code;
```

```
                jLabel2.setVisible(true);
                showInfo(IBBPMAClient.messages.getString("sure"));
                jButton1.setEnabled(true);
                jButton2.setEnabled(true);
        }

        protected void jButton1ActionPerformed(java.awt.event.ActionEvent e) {
                if (IBBPMAClient.vote_system.equals("fptp-default")) {
                        jLabel2.setVisible(false);
                        jButton1.setEnabled(false);
                        jButton2.setEnabled(false);
                        showInfo(IBBPMAClient.messages.getString("vote_send"));
                        IBBPMAClient.comms.sendVote(choice);
                        jToggleButton1.setEnabled(false);
                        jToggleButton2.setEnabled(false);
                        jToggleButton3.setEnabled(false);
                        jToggleButton4.setEnabled(false);
                } else if (IBBPMAClient.vote_system.equals("borda-default")) {
                        if (clicked) {
                                /* error checking */
                                if ((jTextField1.getText().equals("")) ||
(jTextField2.getText().equals("")) || (jTextField3.getText().equals("")) ||
(jTextField4.getText().equals(""))) {

        showError(IBBPMAClient.messages.getString("fill_all"));
                                        jLabel2.setVisible(false); // reset interface
                                        jButton2.setEnabled(false);
                                        choice = "";
                                        clicked = false;
                                } else if ((jTextField1.getText().equals("0")) ||
(jTextField2.getText().equals("0")) || (jTextField3.getText().equals("0")) ||
(jTextField4.getText().equals("0"))) {
                                        // invalid entry

        showError(IBBPMAClient.messages.getString("invalid_chars"));
                                        jLabel2.setVisible(false); // reset interface
                                        jButton2.setEnabled(false);
                                        choice = "";
                                        clicked = false;
                                } else if ((new Integer(jTextField1.getText()).intValue() + new
Integer(jTextField2.getText()).intValue() + new Integer(jTextField3.getText()).intValue() + new
Integer(jTextField4.getText()).intValue()) != bordaSum()) {
                                        // invalid entry

        showError(IBBPMAClient.messages.getString("invalid_chars"));
                                        jLabel2.setVisible(false); // reset interface
                                        jButton2.setEnabled(false);
                                        choice = "";
                                        clicked = false;
                                } else {
                                        choice = new String(cand1code + "-" + new
Integer(IBBPMAClient.number_of_choices - new Integer(jTextField1.getText()).intValue()).toString()
+ "-" + cand2code + "-" + new Integer(IBBPMAClient.number_of_choices - new
Integer(jTextField2.getText()).intValue()).toString() + "-" + cand3code + "-" + new
Integer(IBBPMAClient.number_of_choices - new Integer(jTextField3.getText()).intValue()).toString()
+ "-" + cand4code + "-" + new Integer(IBBPMAClient.number_of_choices - new
Integer(jTextField4.getText()).intValue()).toString());
                                        jLabel2.setVisible(false); // update interface & send vote
                                        jButton1.setEnabled(false);
                                        jButton2.setEnabled(false);
```

```
showInfo(IBBPMAClient.messages.getString("vote_send"));
                                        IBBPMAClient.comms.sendVote(choice);
                                        jTextField1.setEnabled(false);
                                        jTextField2.setEnabled(false);
                                        jTextField3.setEnabled(false);
                                        jTextField4.setEnabled(false);
                                }
                        } else {
                                jLabel2.setVisible(true);
                                showInfo(IBBPMAClient.messages.getString("sure"));
                                jButton1.setEnabled(true);
                                jButton2.setEnabled(true);
                                clicked = true;
                        }
                }
        }

        protected void jButton2ActionPerformed(java.awt.event.ActionEvent e) {
                jLabel2.setVisible(false);
                jButton1.setEnabled(false);
                jButton2.setEnabled(false);
                choice = "";
                clicked = false;
        }

        protected void jButton3ActionPerformed(java.awt.event.ActionEvent e) {
                jButton3.setEnabled(false);
                jButton4.setEnabled(false);
                jButton1.setEnabled(false);
                jButton2.setEnabled(false);
                IBBPMAClient.comms.sendVote(choice);
                showInfo(IBBPMAClient.messages.getString("vote_send2"));
        }

        protected void jButton4ActionPerformed(java.awt.event.ActionEvent e) {
                jButton1.setEnabled(false);
                jButton2.setEnabled(false);
                setVisible(false);
                dispose();
                System.exit(0);
        }

        protected void voteOK() {
                jButton1.setEnabled(false);
                jButton2.setEnabled(false);
                jButton4.setEnabled(true);
                jButton4.setEnabled(true);
                showInfo(IBBPMAClient.messages.getString("vote_reg"));
        }

        protected void voteErr() {
                jButton1.setEnabled(false);
                jButton2.setEnabled(false);
                jButton3.setEnabled(true);
                jButton4.setEnabled(true);
                showError(IBBPMAClient.messages.getString("vote_sorry"));
        }

        public static void main( String args[]) throws Exception{
```

```
            VoteFrame v = new VoteFrame();
            v.initComponents();
            v.setVisible(true);
        }

} //EOF Class
```

xi. VoteFrame.jfrm

```
Metrowerks RAD Data V 1
BeginClasses
  {9BE3D9E0-166E-11D1-B2D8-0060081C5489} "#base-object-class"
  {24A693F8-14E5-11D2-B334-00600819ADE3} "IBBPMAClient"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame1"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JTextPane"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame2"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JScrollPane"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JButton"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame3"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JOptionPane"
  {8AC75CA0-A339-11D1-B2BE-006008A5C0A5} "javax.swing.JPanel"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JLabel"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JTextField"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JToggleButton"
  {AF105FF8-70D9-11D1-B010-00805F6114CC} "javax.swing.JPasswordField"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "FCFrame4"
  {5FA17216-8612-11D1-B2AD-006008A5C0A5} "VoteFrame"
EndClasses
BeginObject IBBPMAClient.VoteFrame "VoteFrame"
  BeginProperties
    enabled = True
    JMenuBar = <none>
    foreground = black
    location = "0, 0"
    title = "IBBPMA: Voting Screen"
    resizable = False
    background = "204, 204, 204"
    font = Application-PLAIN-10
    layout = None
    cursor = Default
    menuBar = <none>
    size = "451, 284"
  EndProperties
  BeginInternalProperties
    UseSwing 11 = 1
    IsApplet 11 = 0
  EndInternalProperties
  BeginEventConnections
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = windowClosing
      EventSetName = window
      HandlerName = thisWindowClosing
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jToggleButton1"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = stateChanged
      EventSetName = change
      HandlerName = jToggleButton1StateChanged
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jToggleButton2"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
```

```
      EventName = stateChanged
      EventSetName = change
      HandlerName = jToggleButton2StateChanged
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jToggleButton3"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = stateChanged
      EventSetName = change
      HandlerName = jToggleButton3StateChanged
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jToggleButton4"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = stateChanged
      EventSetName = change
      HandlerName = jToggleButton4StateChanged
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jButton1"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton1ActionPerformed
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jButton2"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton2ActionPerformed
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jButton3"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton3ActionPerformed
    EndEventConnection
    BeginEventConnection {796E35A1-9372-11D2-9A19-00104B70C619}
      SourcePath = "\\\\IBBPMAClient.VoteFrame\\jButton4"
      SinkPath = "\\\\IBBPMAClient.VoteFrame"
      EventName = actionPerformed
      EventSetName = action
      HandlerName = jButton4ActionPerformed
    EndEventConnection
  EndEventConnections
  BeginObject jScrollPane1 "javax.swing.JScrollPane"
    BeginProperties
      requestFocusEnabled = True
      maximumSize = "32767, 32767"
      enabled = True
      foreground = black
      location = "10, 140"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
      font = Dialog-PLAIN-12
      verticalScrollBarPolicy = 20
      cursor = Default
```

```
      alignmentY = 0.5
      alignmentX = 0.5
      opaque = False
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "22, 22"
      size = "430, 110"
      doubleBuffered = False
      horizontalScrollBarPolicy = 30
    EndProperties
    BeginInternalProperties
    EndInternalProperties
    BeginObject jTextPane1 "javax.swing.JTextPane"
      BeginProperties
        requestFocusEnabled = True
        contentType = text/plain
        selectionStart = 0
        maximumSize = "2147483647, 2147483647"
        enabled = True
        text = ""
        selectedTextColor = black
        foreground = black
        location = "0, 0"
        autoscrolls = True
        visible = True
        background = white
        selectionColor = "204, 204, 255"
        caretColor = black
        font = Application-PLAIN-10
        cursor = Default
        alignmentY = 0.5
        alignmentX = 0.5
        opaque = True
        caretPosition = 0
        selectionEnd = 0
        disabledTextColor = "153, 153, 153"
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "11, 6"
        size = "427, 107"
        doubleBuffered = False
        editable = True
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
  EndObject
  BeginObject jLabel1 "javax.swing.JLabel"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      iconTextGap = 4
      maximumSize = "0, 0"
      enabled = True
      text = "Please make your choice:"
      foreground = "102, 102, 153"
      location = "10, 10"
      autoscrolls = False
      visible = True
      background = "204, 204, 204"
```

```
      font = Application-BOLD-10
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.0
      opaque = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "0, 0"
      size = "290, 20"
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jToggleButton1 "javax.swing.JToggleButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = GREEN
      actionCommand = ""
      contentAreaFilled = True
      location = "10, 40"
      foreground = black
      autoscrolls = False
      visible = True
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      selected = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "35, 11"
      size = "100, 30"
      focusPainted = True
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jToggleButton2 "javax.swing.JToggleButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = LABOUR
      actionCommand = ""
```

```
      contentAreaFilled = True
      location = "120, 40"
      foreground = black
      autoscrolls = False
      visible = True
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      selected = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "35, 11"
      size = "100, 30"
      focusPainted = True
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jToggleButton3 "javax.swing.JToggleButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = "LIB DEM"
      actionCommand = ""
      contentAreaFilled = True
      location = "230, 40"
      foreground = black
      autoscrolls = False
      visible = True
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      selected = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "35, 11"
      size = "100, 30"
      focusPainted = True
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
```

```
    EndInternalProperties
  EndObject
  BeginObject jToggleButton4 "javax.swing.JToggleButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = TORY
      actionCommand = ""
      contentAreaFilled = True
      location = "340, 40"
      foreground = black
      autoscrolls = False
      visible = True
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      selected = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "35, 11"
      size = "100, 30"
      focusPainted = True
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jButton1 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = Yes
      actionCommand = ""
      contentAreaFilled = True
      location = "270, 110"
      foreground = black
      autoscrolls = False
      visible = False
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      defaultCapable = True
```

```
      selected = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "35, 11"
      size = "80, 20"
      focusPainted = True
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jButton2 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = No
      actionCommand = ""
      contentAreaFilled = True
      location = "360, 110"
      foreground = black
      autoscrolls = False
      visible = False
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      defaultCapable = True
      selected = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "35, 11"
      size = "80, 20"
      focusPainted = True
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jLabel2 "javax.swing.JLabel"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      iconTextGap = 4
      maximumSize = "0, 0"
      enabled = True
      text = "Are you sure?"
      foreground = "102, 102, 153"
      location = "170, 110"
      autoscrolls = False
```

```
      visible = False
      background = "204, 204, 204"
      font = Application-BOLD-10
      horizontalAlignment = 2
      cursor = Default
      alignmentY = 0.5
      alignmentX = 0.0
      opaque = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "0, 0"
      size = "90, 20"
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jButton3 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
      enabled = True
      text = Retry
      actionCommand = ""
      contentAreaFilled = True
      location = "280, 260"
      foreground = black
      autoscrolls = False
      visible = False
      rolloverEnabled = False
      background = "204, 204, 204"
      borderPainted = True
      font = Application-BOLD-10
      horizontalAlignment = 0
      cursor = Default
      alignmentY = 0.0
      alignmentX = 0.0
      opaque = True
      defaultCapable = True
      selected = False
      verticalTextPosition = 0
      debugGraphicsOptions = 0
      toolTipText = ""
      minimumSize = "35, 11"
      size = "80, 20"
      focusPainted = True
      doubleBuffered = False
      horizontalTextPosition = 4
    EndProperties
    BeginInternalProperties
    EndInternalProperties
  EndObject
  BeginObject jButton4 "javax.swing.JButton"
    BeginProperties
      requestFocusEnabled = True
      verticalAlignment = 0
      maximumSize = "35, 11"
```

```
        enabled = True
        text = Quit
        actionCommand = ""
        contentAreaFilled = True
        location = "370, 260"
        foreground = black
        autoscrolls = False
        visible = False
        rolloverEnabled = False
        background = "204, 204, 204"
        borderPainted = True
        font = Application-BOLD-10
        horizontalAlignment = 0
        cursor = Default
        alignmentY = 0.0
        alignmentX = 0.0
        opaque = True
        defaultCapable = True
        selected = False
        verticalTextPosition = 0
        debugGraphicsOptions = 0
        toolTipText = ""
        minimumSize = "35, 11"
        size = "70, 20"
        focusPainted = True
        doubleBuffered = False
        horizontalTextPosition = 4
      EndProperties
      BeginInternalProperties
      EndInternalProperties
    EndObject
EndObject
```

15. Βιβλιογραφία – Λοιπές Πηγές

[1] AN ELECTRONIC VOTING SCHEME WITH PHYSICAL MULTIPLE ADMINISTRATORS AND IDENTICAL BALLOT BOXES, *P. S. Alefragis, S.K. Lounis, V. D. Triantafillou and N. S. Voros,* IADIS WWW/Internet 2004 Madrid , Spain pp 99 - 106

[2] VOTING WITH DESIGNATED VERIFIER SIGNATURE-LIKE PROTOCOL, *Emmanuel Dall'Olio and Olivier Markowitch* IADIS WWW/Internet 2004 Madrid , Spain pp 295 - 301

[3] ELECTRONIC GOVERNMENT AND SOCIAL CONTROL OF THE STATE, *Robert Willecke, Hugo Cesar Hoeschl and Marco Aurélio Zimmermann* IADIS WWW/Internet 2004 Madrid , Spain pp

[4] SOFTWARE FOR SELF-GOVERNMENT, *Thomas B. Senior and Brian Warboy,*IADIS WWW/Internet 2004 Madrid , Spain

[5] INTEGRATED GOVERNMENTAL PUBLICATION SYSTEM *Carlos del Cuvillo, Héctor García, Diego Pérez and Eva Franco, Robert Willecke, Hugo Cesar Hoeschl and Marco Aurélio Zimmermann,*IADIS WWW/Internet 2004 Madrid , Spain

[6] ANALYSIS OF AN ELECTRONIC VOTING SYSTEM *TADAYOSHI KOHNO, ADAM STUBBLEFIELD AVIEL D. RUBIN DAN S. WALLACH* IEEE Symposium on Security and Privacy 2004, February 27, 2004

[7] CITIZEN INFORMATION SERVICES USING INTERNET TECHNOLOGIES *Bouras C., Kastaniotis S., Triantafillou V.,*European Conference on Information Systems Vienna, Austriam Jully 3 -5 pp 1123 -1130

[8] A CO-OPERATIVE ENVIROMENT FOR LOCAL GOVERNMENT : AN INTERNET – INTRANET APPROACH *Bouras C., Destounis P., Garofalakis J., Tzimas J., Triantafillou V., Zarafidis P* Journal of Telematics, PERGAMON PRESS, vol. 16/1-2, pp 75 – 84

[9] DIGITAL POLITICS 2000. THE VOTE'S IN: THE WEB'S POTENTIAL IN THE POLITICAL PROCCESS IS STILL LACKING. *Berhel H.* Communications of the ACM, Nov Vol.43, No.11, pp 17-22

[10] DOES THE INTERNET INCREASE VOTER PARTICIPATION IN ELECTIONS? Caroline Tolbert, Ramona McNeal, Annual Meeting of the American Political Science Association, San Francisco 2001

[11] LEARNING IN ELECTIONS AND VOTER TURNOUT EQUILIBRIA
Stefano DeMichelis, Amrita Dhillon, Warwick Economic Research Papers, 2001

[12] THE ITALIAN ACADEMIC COMMUNITY'S ELECTRONIC VOTING SYSTEM
Bonetti P., Ravaioli S., Piergallini S
TERENA Networking Conference 2000 "Pioneering Tommorow's Internet", Lisbon 22-25 May 2000

[13] A SECURE AND OPTIMAL EFFICIENT MULTI – AUTHORITY ELECTION SCHEME
Ronald Cramer, Rosario Gennaro, Berry Schoenmakers, Institute for Theoretical Computer Science, 1997

[14] SEQURITY CONSIDERATIONS FOR REMOTE ELECTRONIC VOTING OVER THE INTERNET
Avi Rubin, AT&T Labs

[15] MULTI – AUTHORITY SECRET – BALLOT ELECTIONS WITH LINEAR WORK
Ronald Cramer, Matthew Franklin, Berry Schoenmakers, Moti Yung
EUROCRYPT96, Berlin 1996

[16] http://java.sun.com/products/javawebstart/reference/techart/index.html

[17] http://www.hartintercivic.com/default.asp

[18] http://www.evs-j.com/English/Introduction/introduction.htm

[19] http://www.votations.com/

[20] http://www.infopoll.com/

[21] http://www.vote.caltech.edu/index.html

[22] http://www.safevote.com/index.html

[23] http://lorrie.cranor.org/voting/sensus/index.html

[24] http://www.trueballot.com/default.htm

[25] http://votehere.net/default.htm

[26] http://www.votingondemand.com/marketing/product_overview.cfm

[27] http://www.votia.com/english/index.html