Vault

Μία κλειδοθήκη για τις ανάγκες του σύγχρονου πολιτισμού

Ελευθεριάδης Αντώνιος 8398
Δερβίση Φωτεινή 8428
Βαβούρης Απόστολος 8443

Συνεισφορά μελών:

Ελευθεριάδης Αντώνιος

Εμπνευστής του συστήματος που περιγράφεται εν συντομία παρακάτω με επιρροές από τα OpenPGP, Bitcoin, UAF, ...

Σχεδίασε και υλοποίησε τον προσομοιωτή (οτιδήποτε υπάρχει κάτω από τον κατάλογο emulator).

OpenPGP Message Format Bitcoin: A Peer-to-Peer Electronic Cash System Namecoin fido alliance

Για τα αρχεία που υπάρχουν κάτω από τον κατάλογο demo:

Δερβίση Φωτεινή

Το demo.qrc και το main.cpp

Από το clie.hpp την class client

Από το clie.cpp operator>>, addorder, compute_price, delorder

Από το window.cpp τα MainWindow, newClient, saveClient, saveasClient, buy, addToBasket, createMenus, createToolBars, createButtons (από κοινού), confirm

Βαβούρης Απόστολος

Το demo.pro

Από το clie.hpp την struct order

Από το clie.cpp operator<<, struct order getorder, trouver, NoOfOrders

Από το window.cpp τα closeEvent, openClient, selectClient, selectShop, selectBank, About, createActions, createProductList, createButtons (από κοινού), realsaveClient

Το window.h γράφτηκε από κοινού όταν αποφασιζόταν η δομή του προγράμματος.

Εισαγωγή

Η σωστή λειτουργία των λειτουργιών της σύγχρονης κοινωνίας βασίζεται στην έννοια της εμπιστοσύνης. Για την εκτέλεση κάθε λειτουργίας απαιτείται η ύπαρξη αμοιβαίας εμπιστοσύνης είτε μεταξύ δύο μελών είτε μεταξύ κάθε μέλους, χωριστά, και ενός κοινού τρίτου μέλους.

Παραδοσιακά αυτό επιτυγχάνεται με την χρήση συμβόλων όπως ταυτότητες, υπογραφές, χαρτονομίσματα, δίπλωμα οδήγησης κ.λ.π. Ουσιαστικά, αυτά τα σύμβολα χρησιμοποιούνται για να επαληθεύσουν ιδιότητες του ατόμου που τα κατέχει π.χ. συμφωνία, ικανότητα οδήγησης, κατοχή και μεταφορά αξίας κ.λ.π.

Για να είναι αποδεκτά αυτά τα σύμβολα για την παροχή εμπιστοσύνης πρέπει να πληρούν τις παρακάτω ιδιότητες:

- Απόδειξη ύπαρξης-Αντίσταση στην παραχάραξη
- Απόδειξη ακεραιότητας-Αντίσταση στην παραποίηση
- Επαληθευσιμότητα

Γι' αυτό τον λόγο έχουν αναπτυχθεί τεχνικές, όπως, ολογράμματα, υδατογράφηση, μικροεκτύπωση, χρήση μελανιού ορατό στο υπεριώδες φάσμα κ.λ.π.

Αυτές οι τεχνικές μπορούν να παρακαμφθούν ή σε ορισμένες περιπτώσεις δεν χρειάζεται να παρακαμφθούν, διότι, η επαλήθευση τους είναι δύσκολη και παραλείπεται. Αυτή η κατάσταση κάνει το σύστημα εμπιστοσύνης να καταρρεύσει.

Δεδομένης της ευρείας διάδοσης του διαδικτύου προτείνω μία διαφορετική λύση στο πρόβλημα της εμπιστοσύνης με την μορφή ενός συστήματος που βασίζεται στην κρυπτογράφηση δημοσίου κλειδιού και υλοποιείται με την χρήση έξυπνων καρτών με έμφαση στην διαλειτουργικότητα. Αυτό το σύστημα έχει σκοπό να είναι επεκτάσιμο ορίζοντας μόνο τους βασικούς λίθους για την δημιουργία ενός πιο πολύπλοκου συστήματος εμπιστοσύνης, παρέχοντας ένα σύνολο από εύκολα, και σε πραγματικό χρόνο, επαληθεύσιμων μαθηματικών αποδείξεων.

Σε αυτό το σύστημα το βασικό σύμβολο εμπιστοσύνης είναι ένα ζεύγος δημόσιου-ιδιωτικού κλειδιού και γίνεται η υπόθεση ότι αν κάποιος προσπαθήσει να εξάγει τα κλειδιά από την έξυπνη κάρτα ή προσπαθήσει να βρει με εξαντλητική αναζήτηση το PIN της κάρτας τότε τα μυστικά κλειδιά θα αυτοκαταστραφούν.

Το βασικό κλειδί του συστήματος είναι το κύριο κλειδί “main” το οποίο αποτελεί άγκυρα εμπιστοσύνης, συνδέεται με την ταυτότητα του ατόμου και η ύπαρξη του είναι επαληθεύσιμη από ένα έμπιστο-ουδέτερο τρίτο μέλος (π.χ. κρατική βάση δεδομένων). (Στόχος του συστήματος είναι η χρήση αυτού του κλειδιού για την δημιουργία μίας αλυσίδας εμπιστοσύνης που επιτρέπει την απονομή ευθυνών και δικαιοσύνης όταν κάποιο μέλος της κοινωνίας παραμελεί το καθήκον του ή παρανομεί χωρίς να εκθέτει τα προσωπικά στοιχεία του ατόμου ή να επιτρέπει την παραβίαση της ιδιωτικότητας του στις διάφορες δραστηριότητες του).

Η συγκεκριμένη υλοποίηση του συστήματος χρησιμοποιεί για τα ζεύγη κλειδιών την ελλειπτική καμπύλη ed25519 και για την δημιουργία ψηφιακών υπογραφών τον αλγόριθμο EdDSA(Ed25519-SHA-512).

(Σημείωση1: η επιλογή κρυπτογραφίας ελλειπτικών καμπυλών έναντι πιο κλασικών συστημάτων π.χ. RSA έγινε επειδή παρέχει ισοδύναμο επίπεδο ασφάλειας με μικρότερο μέγεθος κλειδιών. Αυτό μεταφράζεται σε μικρότερες ανάγκες επεξεργαστικής ισχύος, πράγμα που αποτελεί σημαντικό παράγοντα σε ένα ενεργειακά περιορισμένο περιβάλλον, όπως αυτό μίας έξυπνης κάρτας)
(Σημείωση2: ο συγκεκριμένος αλγόριθμος επιλέχθηκε εξαιτίας της ανθεκτικότητας του σε επιθέσεις τύπου side channel και επειδή δεν χρειάζεται μία γεννήτρια τυχαίων αριθμών κατά την δημιουργία ψηφιακών υπογραφών, αφού παράγει το nonce αιτιοκρατικά π.χ. αν στον αλγόριθμο DSA η τυχαία παράμετρος είναι προβλέψιμη τότε η υπογραφή εκθέτει το μυστικό κλειδί που χρησιμοποιήθηκε)

Η εργασία αποτελείται από έναν προσομοιωτή για τις έξυπνες κάρτες και ένα demo καταστήματος.

Εγκατάσταση

Τόσο για την μεταγλώττιση του προσομοιωτή όσο και για το demo απαιτούνται ένας σύγχρονος compiler με υποστήριξη του προτύπου C++11 και η πλατφόρμα Qt, έκδοση 5 ή νεότερη. Η ανάπτυξη και οι δοκιμές έγιναν σε συστήματα GNU/Linux με τις διανομές Ubuntu 14.04.3 και Debian 8.2, με compiler τον g++ από την συλλογή gcc και Qt 5.5

Για την μεταγλώττιση και την εκτέλεση του προσομοιωτή αρκεί η εκτέλεση της εντολής “qmake && make && ./emulator” στον φάκελο emulator. Κάποιοι μεταγλωττιστές δηλώνουν ότι η εντροπία της γεννήτριας τυχαίων αριθμών είναι πάντα 0. Σε αυτήν την περίπτωση για λόγους δοκιμής χρειάζεται να γίνει comment out η γραμμή 105 στο αρχείο smartcard.cpp στον φάκελο emulator πριν την μεταγλώττιση.

Πριν από την μεταγλώττιση του δοκιμαστικού προγράμματος πρέπει να δημιουργηθεί μία εικονική έξυπνη κάρτα με τον emulator η οποία θα έχει ένα ζεύγος κλειδιών με αναγνωριστικό “transaction_confirm”. Αυτό μπορεί να γίνει χρησιμοποιώντας το κουμπί “Generate new key pair” από το μενού Debug->Run Command του προσομοιωτή(το προεπιλεγμένο PIN είναι 0000). Στην συνέχεια το δημόσιο κλειδί που θα επιστραφεί, με την μορφή pop-up, μετά από την επιτυχημένη εκτέλεση της εντολής, πρέπει να τοποθετηθεί στην γραμμή 21 του αρχείου window.cpp στον φάκελο demo.
Έπειτα για την μεταγλώττιση του δοκιμαστικού demo αρκεί η εκτέλεση της εντολής “qmake && make && ./demo” στον φάκελο demo.

Περιγραφή κλάσεων

Σε ό,τι αφορά τον προσομοιωτή της έξυπνης κάρτας:

Ορίζεται μία δομή “keypair” η οποία αποτελείται από δύο πίνακες, 64 και 32 byte, για την αποθήκευση του ιδιωτικού και δημόσιου κλειδιού αντίστοιχα, ενός ζεύγους.

Η κλάση card ορίζει τις βασικές λειτουργίες που επιτελεί η έξυπνη κάρτα, δηλαδή, υλοποιεί με την μορφή μεθόδων τις εντολές που δέχεται η smart card και περιγράφονται παρακάτω. Επίσης, διαθέτει ορισμένες επιπλέον μεθόδους που απαιτούνται για την λειτουργία του προσομοιωτή (id κάρτας, αποθήκευση σε αρχείο και ανάκτηση των δεδομένων της).

Οι κλάσεις KeyExists και KeyNotExists κληρονομούν την std::exception και ορίζουν τις εξαιρέσεις που πετάγονται όταν ένα ζεύγος υπάρχει ενώ δεν έπρεπε ή δεν υπάρχει ενώ έπρεπε, αντιστοίχως.

Η κλάση MainWindow κληρονομεί την QMainWindow, αποτελεί το βασικό παράθυρο της εφαρμογής, περιλαμβάνει τα μενού αρχείων, αποσφαλμάτωσης, βοήθειας και επιτρέπει την δημιουργία μίας νέας κάρτας, την αποθήκευση της σε αρχείο και την ανακατασκευή της από τα δεδομένα του αρχείου.

Η κλάση cardDock κληρονομεί την QDockWidget και ορίζει ένα dock widget το οποίο αντιπροσωπεύει μία κάρτα έχοντας την αντίληψη ενός αντικειμένου τύπου card. Αναλυτικότερα, δημιουργεί τον server που ακούει για νέες συνδέσεις, επεξεργάζεται τις κλησεις της διεπαφής προγραμματισμού εφαρμογών, επιβάλλοντας, ταυτόχρονα, τους αντίστοιχους κανονισμούς, και στην συνέχεια μεταβιβάζει τις εντολές στο αντικείμενο τις κάρτας που έχει υπό την αντίληψη του και προβάλει τις καταγραφές των συμβάντων.

Η κλάση commandDialog κληρονομεί την QWidget και ορίζει ένα παράθυρο το οποίο επιτρέπει στον χρήστη να συνδεθεί σε μία κάρτα, και με τα κουμπιά που διαθέτει να εκτελέσει κάποια εντολή με τα ορίσματα που επιθυμεί και να δει το αποτέλεσμα της.

Σε ό,τι αφορά το demo καταστήματος:

Στο αρχείο window το MainWindow κληρονομεί την QMainWindow, η οποία αποτελεί την βασική κλάση για τη δημιουργία του παραθύρου της εφαρμογής και περιέχει όλες τις λειτουργίες της εφαρμογής, τα κουμπιά κτλ.

Μέσω μεθόδων γίνονται τα εξής:

- Δημιουργείται νέος πελάτης
- Ανοίγει υπάρχων πελάτης
- Αποθηκεύεται πελάτης
- Αποθηκεύεται για πρώτη φορά ο καινούριο πελάτη
- Μπορώ να επιλέξω την τράπεζα
- Μπορώ να προβώ σε αγορά
- Μπορώ να δω το about του προγράμματος (που περιλαμβάνει σύντομη

περιγραφή του προγράμματος και τα ονόματα των δημιουργών) και το about Qt (που περιλαμβάνει πληροφορίες για το Qt)

- Μπορώ να προσθέσω προϊόντα στο καλάθι
- Υπάρχουν τα μενού (file, cards, help)
- Στο file υπάρχουν οι εξής υποκατηγορίες: η δημιουργία νέου πελάτη, το άνοιγμα πελάτη, η αποθήκευση και η αποθήκευση ως, και τέλος το help, το οποίο περιλαμβάνει το about application και about Qt.
- Επίσης υπάρχουν και τα κουμπιά που επιτρέπουν νέο, άνοιγμα και αποθήκευση

Στο αρχείο clie υπάρχουν το struct order και η κλάση client.

Το struct order είναι ουσιαστικά μια παραγγελία η οποία περιλαμβάνει ως στοιχεία της διανύσματα, στα οποία αποθηκεύονται τα ονόματα των αιτούμενων προς αγορά προϊόντων, οι τιμές τους και οι ποσότητές τους. Περιλαμβάνει επίσης και τον αριθμό ταυτοποίησης της παραγγελίας.

Η κλάση client ορίζει έναν πελάτη και στην συνέχεια μπορεί να προσθέτει παραγγελία, να διαγράφει παραγγελία, να βρίσκει συγκεκριμένη παραγγελία μέσω του ταυτοποιητικού αριθμού, να υπολογίζει το συνολικό κόστος παραγγελίας, τον αριθμό παραγγελιών και να κάνει operator overloading για να γράφει στο αρχείο.

Περιγραφή προγραμμάτων

Σε ό,τι αφορά τον προσομοιωτή της έξυπνης κάρτας:

Σκοπός του προσομοιωτή είναι να επιτρέπει την δημιουργία εικονικών smart cards και την εκτέλεση των εντολών στην θέση μίας πραγματικής για λόγους δοκιμών. Συγκεκριμένα με το μενού “File” ο χρήστης μπορεί να δημιουργήσει μία νέα κάρτα ή να χρησιμοποιήσει μία υπάρχουσα. Κάθε ενεργή κάρτα ακούει σε μία θύρα TCP για εισερχόμενες συνδέσεις. Όταν μία σύνδεση δημιουργηθεί τότε διαβάζεται η εντολή του API και οι παράμετροι της και αφού επαληθευτούν επιστρέφεται το αποτέλεσμα της επιτυχημένης εκτέλεσης ή, ίσως, ένα μήνυμα σφάλματος. Αν είτε η εντολή είτε οι παράμετροι δεν είναι έγκυρα επιστρέφεται το κατάλληλο μήνυμα σφάλματος. Επίσης, κάθε κάρτα αποθηκεύεται αυτόματα, αμέσως μόλις αλλάξουν τα δεδομένα της.

Το πρωτόκολλο που ακολουθείται είναι το παρακάτω:

1) Ανοίγει η TCP/IP σύνδεση με τον server

2) Ο πελάτης παραδίδει ένα (δυαδικό) μήνυμα με την παρακάτω μορφή

<εντολή> '\n' <παράμετρος 1> '\n' <παράμετρος 2> '\n' <...> '\n' <παράμετρος n> '\n'

3) Το λειτουργικό σύστημα της κάρτας ή στην περίπτωση μας ο προσομοιωτής επαληθεύει την εντολή, τον αριθμό των arguments (αν δοθούν περισσότερες παράμετροι από όσες απαιτεί η εντολή οι επιπλέον θα αγνοηθούν) και το περιεχόμενο τους

4) Αν όλα έχουν πάει καλά μέχρι στιγμής η κάρτα ζητάει μέσω ενός

δεύτερου, έμπιστου καναλιού επικοινωνίας από τον κάτοχο να επαληθεύσει την εντολή και τις παραμέτρους απαιτώντας την εισαγωγή του PIN για πιστοποίηση (στον προσομοιωτή υλοποιείται με την μορφή ενός pop-up) εκτός από την περίπτωση όπου η εντολή δεν απαιτεί πιστοποίηση, αλλιώς, πηγαίνουμε στο βήμα 6.

5) Αν είναι όλα καλά μέχρι στιγμής η εντολή εκτελείται, διαφορετικά πηγαίνουμε κατευθείαν στο βήμα 6.

6) Ο εξυπηρετητής αποστέλλει ένα μήνυμα με την παρακάτω μορφή στον πελάτη

<0 αν η λειτουργία πέτυχε ή 1 σε αποτυχία (1 byte)><αποτέλεσμα εντολής ή προαιρετικό μήνυμα σφάλματος>

7) Ο server διακόπτει την σύνδεση

Το υποστηριζόμενο API είναι:

Εντολή	**Πλήθος παραμέτρων**	**Παράμετροι**	**Αποτέλεσμα**
gen_key	1	Αναγνωριστικό νέου ζεύγους (string)	Επιστρέφει το δημόσιο κλειδί του νεόυ ζεύγους (string) (κωδικοποιημένο σε Base64)
gen_trusted_key	1	Αναγνωριστικό νέου ζεύγους (string)	Επιστρέφει το δημόσιο κλειδί του νέου ζεύγους και την ψηφιακή υπογραφή του με το κλειδί main ακολουθώντας την μορφοποίηση της εντολής sign (string)
sign	2	Αναγνωριστικό ζεύγους (string), Μήνυμα προς υπογραφή (string) (κωδικοποιημένο σε Base64)	Επιστρέφει το μήνυμα υπογεγραμμένο με την ακόλουθη μορφή (string) -----BEGIN SIGNED MESSAGE-----<μήνυμα (κωδικοποιημένο σε Base64)>-----BEGIN SIGNATURE-----<ψηφιακή υπογραφή (κωδικοποιημένη σε Base64)>-----END SIGNATURE-----
del_key	1	Αναγνωριστικό ζεύγους (string)	Διαγράφει το ζεύγος κλειδιών με το συγκεκριμένο αναγνωριστικό
set_pin	1	Νέο PIN (string)	Θέτει το PIN
get_random	1	Πλήθος bytes (unsigned int)	Επιστρέφει τον αριθμό των τυχαίων bytes που ζητήθηκαν (string) (κωδικοποιημένα σε Base64)

get_logs	0	-	Επιστρέφει τις καταγραφές του συστήματος (string)
get_version	0	-	Επιστρέφει την έκδοση του API (string)

(Σημείωση: **ΜΟΝΟ** οι εντολές get_random και get_version **ΔΕΝ** απαιτούν πιστοποίηση από τον κάτοχο της κάρτας)

Σε ό,τι αφορά το demo καταστήματος:

Το demo προσομοιάζει τη λειτουργία του ηλεκτρονικού καταστήματος.

Στο κατάστημα υπάρχει μία λίστα διαθεσίμων προϊόντων και οι τιμές τους. Ο πελάτης μπορεί να επιλέξει τα προϊόντα και σε ποιες ποσότητες τα επιθυμεί και με αυτόν τον τρόπο να προχωρήσει στην αγορά (συμπληρώνεται ένα καλάθι με τα αιτούμενα προς αγορά προϊόντα). Έπειτα από την συνολική επιλογή των προϊόντων από τον πελάτη, μόλις πατήσει το κουμπί buy το συνολικό ποσό της συναλλαγής και ο χρόνος αγοράς μετρημένος σε δευτερόλεπτα στέλνεται ως μήνυμα στην κάρτα του καταστήματος για να το υπογράψει, με το κλειδί Outgoing Order. Το αποτέλεσμα από την κάρτα του καταστήματος στέλνεται στη κάρτα του πελάτη ούτως ώστε να υπογράφει το κλειδί (ονομαζόμενο pisteos) για να γίνει γνωστό από πού πρέπει να αφαιρεθούν τα χρήματα. Το κατάστημα έπειτα παίρνει αυτό το μήνυμα και το στέλνει στην τράπεζα.

Η τράπεζα ελέγχει την ορθότητα της συναλλαγής (δηλαδή αν υπάρχουν τα χρήματα, ο λογαριασμός και τα λοιπά· για χάριν οικονομίας του προγράμματος στο πρόγραμμα που παρουσιάζουμε οι συγκεκριμένες λειτουργίες λείπουν). Τέλος, το κατάστημα παραλαμβάνει το μήνυμα από την τράπεζα υπογεγραμμένο (transaction_confirm) και συνεπώς η συναλλαγή είναι εγκεκριμένη. Οπότε στο τέλος αναδύεται ένα μήνυμα στο οποίο αναφέρεται αν ήταν επιτυχημένη η συναλλαγή.

Μελλοντικές επεκτάσεις

Σε ό,τι αφορά τον προσομοιωτή της έξυπνης κάρτας:

1) Τα κλειδιά να έχουν τις παρακάτω σημαίες

Flag	**Χρήση**
sign	**ΜΟΝΟ** για δημιουργία ψηφιακών υπογραφών
encrypt	**ΜΟΝΟ** για την αποκρυπτογράφηση μηνυμάτων
auth	**ΜΟΝΟ** για πιστοποίηση (π.χ. SSH, Password-less logins, etc)
nfc	**ΜΟΝΟ** για την υπογραφή πιστοποιητικών (ίσως X.509) που θα χρησιμοποιηθούν για την δημιουργία ενός ασφαλούς καναλιού επικοινωνίας με μία έμπιστη συσκευή

2) Εμπλουτισμός του API με τις ακόλουθες εντολές

Εντολή	**Πλήθος παραμέτρων**	**Παράμετροι**	**Αποτέλεσμα**
decrypt	2	Αναγνωριστικό ζεύγους (string), Μήνυμα προς αποκρυπτογράφηση (string)(κωδικοποιημένο σε Base64)	Επιστρέφει το αποκρυπτογραφημένο μήνυμα (string) (κωδικοποιημένο σε Base64)
connect	-	-	Επιστρέφει το υπογεγραμμένο πιστοποιητικό που θα χρησιμοποιηθεί από την έμπιστη συσκευή για να δημιουργήσει μία ασφαλή σύνδεση μέσω NFC με την κάρτα
-	-	-	Παρέχει έναν γενικό και ευέλικτο τρόπο για ολοκλήρωση κρυπτογραφικών προκλήσεων με σκοπό την πιστοποίηση του χρήστη

(Σημείωση: η εντολή connect, προφανώς, δεν απαιτεί πιστοποίηση, διότι, ο λόγος ύπαρξης της είναι η δημιουργία του ασφαλούς καναλιού για την μεταφορά των διαπιστευτηρίων)

3) Μου φαίνεται ότι υπάρχει ένα race condition, από την στιγμή που ο server δέχεται την σύνδεση, κατά τις φάσεις της ανάγνωσης των δεδομένων που έστειλε ο client και της εκτέλεσης της εντολής, μέχρι να αποσταλεί το αποτέλεσμα στον client και να κλείσει η σύνδεση, το οποίο πρέπει να διορθωθεί.

Σε ό,τι αφορά το demo καταστήματος:

1) Δυνατότητα αποθήκευσης ιστορικού πελάτη (σαν αυτό που κρατάει το αρχείο μιας τράπεζας, δηλαδή το κατάστημα να μπορεί να κρατάει στοιχεία για τις πληρωμές που έχει πραγματοποιήσει ο πελάτης, το χρονικό στίγμα στο οποίο πραγματοποιήθηκε κτλ)

2) Δυνατότητα ο χρήστης να προσθέτει νέα προϊόντα, όπως επίσης και να αφαιρεί ή να τροποποιεί ήδη υπάρχοντα.

3) Προσθήκη σήμανσης των προϊόντων ως διαθέσιμων ή σε έλλειψη, παράδοση σε χ μέρες

4) Δυνατότητα μορφοποίησης καλαθιού, δηλαδή αλλαγή της αιτούμενης προς αγορά ποσότητας προϊόντος ή και μερική ή ολική διαγραφή της παραγγελίας

5) Loyalty programme with membership key pair

6) Έλεγχος ότι ο πελάτης δεν έχει αλλάξει το μήνυμα μέσω ελέγχου του αρχικού μηνύματος που έχει στείλει το κατάστημα.