PKRY Elektroniczne głosowanie

****

## Projekt realizowany w ramach przedmiotu Protokoły kryptograficzne (PKRY) w semestrze 14Z.

## Skład grupy projektowej:

* Krystian Powójski
* Marcin Skwarek
* Anh Tuan Nguyen

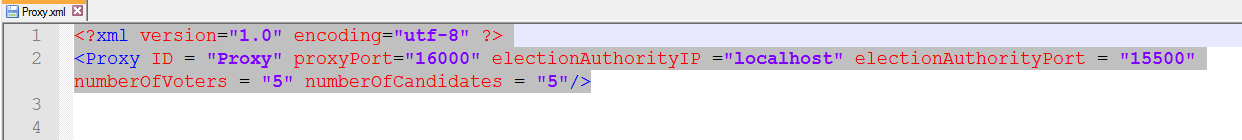
## Projekt składa się z trzech aplikacji:

* Election Authority
* Proxy
* Voter

Aplikacje te wykorzystują do komunikacji architekturę klient-server TCP. Mogą one działać na jednym komputerze ale równie dobrze mogą zostać uruchomione na różnych maszynach. Konfiguracja poszególnych modułów jest wczytywana z plików konfiguracyjnych (pliki z rozszerzeniem \*.xml) znajdujące się w katalogu *Config*.

Lista kandydatów również jest wczytywana z pliku konfiguracyjnego znajdującego sie w katalogu *Config\CandidateList.xml.*

## Opis przykładowego pliku konfiguracyjnego

*Proxy.xml*

Powyżej zaprezentowano plik konfiguracyjny aplikacji *Proxy*. Opis poszczególnych znaczników:

* ProxyID - unikatowe ID aplikacji
* proxyPort – numer portu na którym działa serwer aplikacji Proxy
* electionAuthorityIP – adres IP na którym działa aplikacja ElectionAuthority
* electionAuthrityPort – numer portu na którym działa serwer aplikacji ElectionAuthority
* numberOfVoters – liczba głosujących biorących udział w głosowaniu za pośrednictwem tego Proxy
* numberOfCandidates – liczba kandydatów na których można oddać swój głos w wyborach

Pliki konfiguracyjne pozostałych aplikacji zostały opisane w analogiczny sposób. Przed uruchomieniem konieczne jest wskazanie pliku zawierającego konfigurację.

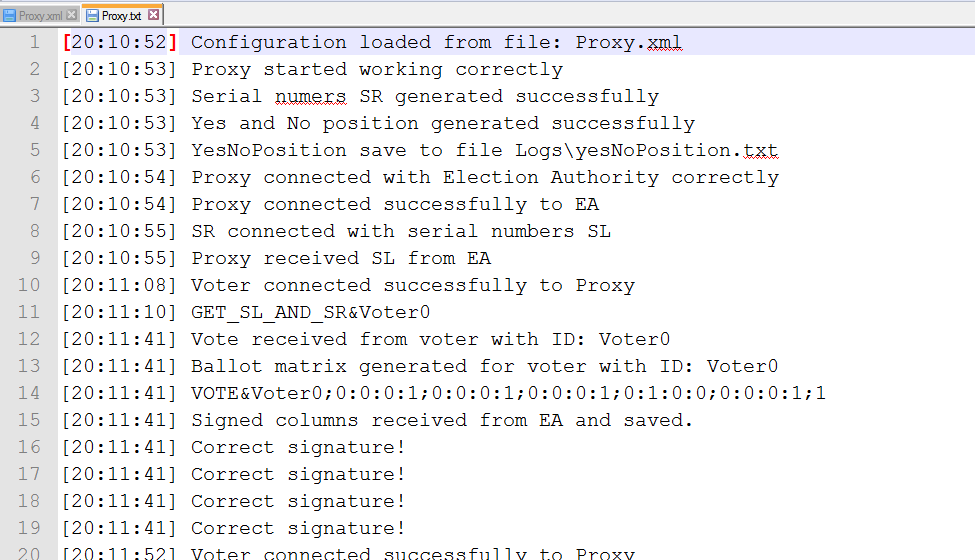
W celu prawidłowego działania projektu sugerowane jest uruchamianie go przy użyciu pliku *run\_all.bat.* Dla każdej sesji uruchamiane jest po jednej aplikacji ElectionAuthority i Proxy. Użytkownik zostanie zapytany o ilość aplikacji typu Voter które powinny zostać uruchomione. Po wprowdzeniu żadanej wartości, projekt zostanie wystartowany.

Taki sposób uruchomienia jest zalecany ze względu na możliwość zapisu logów do plików znajdujacych się w katalogu *Logs/.*

## Logowanie – informacje o tym co dzieje się w aplikacjach

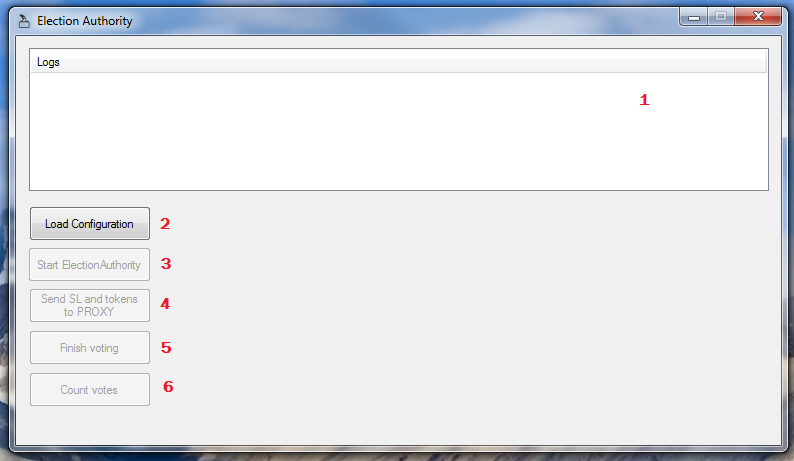
Logi, czyli informacje o akutalnie realizowanych zadaniach, odebranych danych itp. pozwalają użytkownikowi lepiej zrozumieć logikę działania aplikacji. Dodatkowo niosą informację o ewentualnych błędach i działaniach nieporządanych.

Każda z uruchomionych aplikacji posiada oddzielny plik do logowania (format \*.txt), którego nazwa jest zgodna z jej nazwą. Dane dopisuja się zawsze na końcu pliku, co za tym idzie mamy możliwość sprawdzenia informacji z poprzednich startów.



*Logs/Proxy.txt*

## Intrukcja użytkowania aplikacji Election Authority



Opis interfejsu graficznego:

1 – konsola logowania, zawiera wszystkie istotne informacje o zdarzeniach zaistniałych w trakcie działania aplikacji

2 – przycisk do załadowania konfiguracji z pliku \*.xml

3 – przycisk do uruchomienia Election Authority, przycisk jest dostępny w momencie poprawnego załadowania konfiguracji

4 – przycisk przesyłający SL i tokeny do aplikacji PROXY, przycisk jest dostępny po uruchomieniu Election Authority

5 – przycisk kończący proces zbierania głosów, przycisk jest dostępny w momencie przesłania SL i tokenów do PROXY

6 – przycisk uruchamiający zliczanie głosów i następnie wyświetlenie wyniku wyborów, przycisk jest dostępny po naciśnięciu przycisku zakończenia głosowania

## Intrukcja użytkowania aplikacji Proxy

## C:\Users\Krystian\Desktop\proxy_instruction.png

Opis interfejsu graficznego:

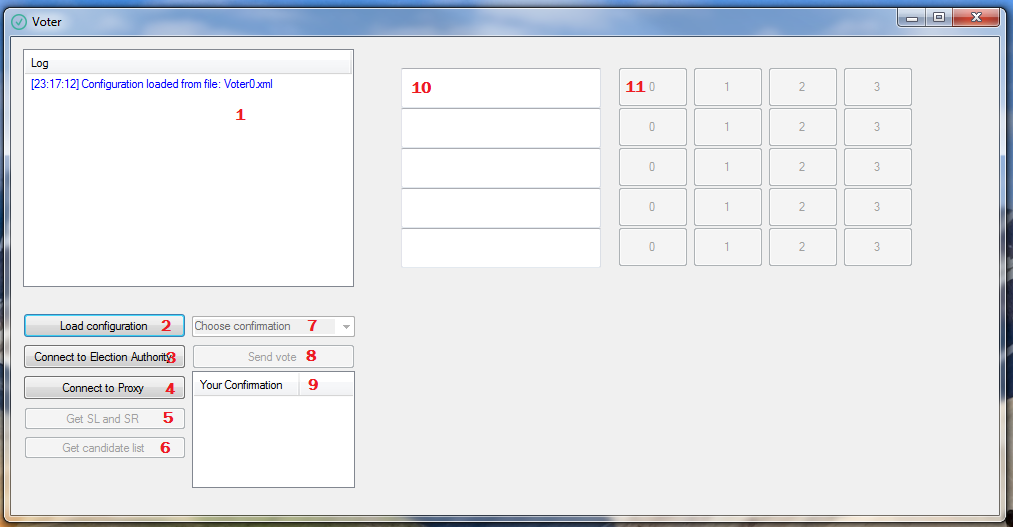
1 – konsola logowania, zawiera wszystkie istotne informacje o zdarzeniach zaistniałych w trakcie działania aplikacji

2 – przycisk do załadowania konfiguracji z pliku \*.xml

3 – przycisk do uruchomienia Proxy, przycisk jest dostępny w momencie poprawnego załadowania konfiguracji

4 – przycisk umożliwiający połączenie Proxy z Election Authority

## Instrukcja użytkownia aplikacji Voter



Opis interfejsu graficznego:

1 – konsola logowania, zawiera wszystkie istotne informacje o zdarzeniach zaistniałych w trakcie działania aplikacji

2 – przycisk do załadowania konfiguracji z pliku \*.xml

3 – przycisk umożliwiający połączenie z Election Authority

4 – przycisk umożliwiający połączenie z Proxy

5 – przycisk umożliwiający wysyłanie rządania o numer SL i SR do Proxy

6 – przycisk umożliwiający wysłanie rządania o listę kandydatów do Election Authority

7- lista rozwiajana umożliwiający wybór kolumy która ma być używana jako potwierdzenie w procesie głosowania

8 – przycisk do przesłania głosu do Proxy

9 – konsola w której wyświetlane jest potwierdzenie

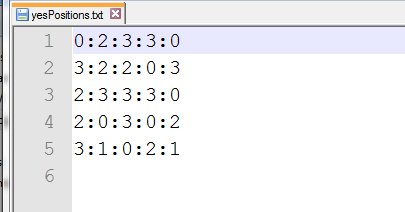
10 – Text Boxy wyświetlający dane kandydató

11 – przyciski do głosowania

## Procedura głosowania

Aby oddać głosna na wybranego kandydata należy w rzędzie przy jego nazwisku wcisnąc przycisk YES a we wszystkich pozostałych NO.

Zgodnie z założeniami w konsoli do głosowaia (aplikacja Voter) na przyciskach nie są jawnie umieszczone nazwy– zamiast tego w każdym rzędzie mamy przyciski z numerami od 0 do 3. Każdy z głosujacych może sprawdzić pozycje przycisków YES w pliku *Logs/yesPositions.txt*. Dane przeznaczone dla niego znajdują się w rzędzie o numerze n+1, gdzie n to liczba w ID głosującego.



*yesPositions.txt*

Np. Dane dla Votera o ID *Voter0* znajdują się w rzędzie numer 1. Sposób interpretacji danych:

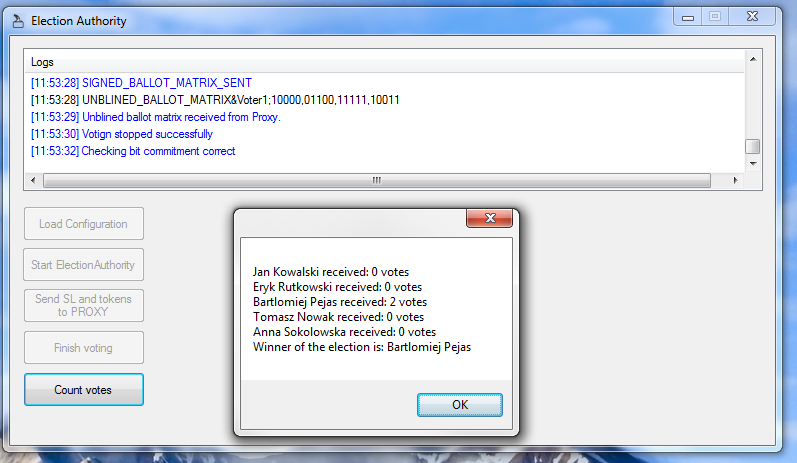
* dla pierwszego kandydata przcisk YES pod przciskiem z napisem 0
* dla drugiego kandydata przcisk YES pod przciskiem z napisem 2
* dla trzeciego kandydata przcisk YES pod przciskiem z napisem 3
* itd

Jakakolwiek próba oszustwa lub pomyłka w procesie oddawania głosu spowoduje że głos zostanie uznany za nieważny i nie będzie uwzględniany w procesie zliczania głosów. Taka sama sytuacja ma miejsce jeśli użytkownik wcisnie dwa razy przyciski w tym samym rzędzie – zostaje o tym poinformowany w konsoli logowania odpowiednim monitem.

Po prawidłowym procesie głosowania wyborca zobowiązany jest do wyboru kolumny którą będzie używał jako potwierdzenia – lista rozwijana z nazwami kolumn (od A do D). Następnie należy wcisnąć przycisk *Send vote,* który przekaże głos do Proxy.

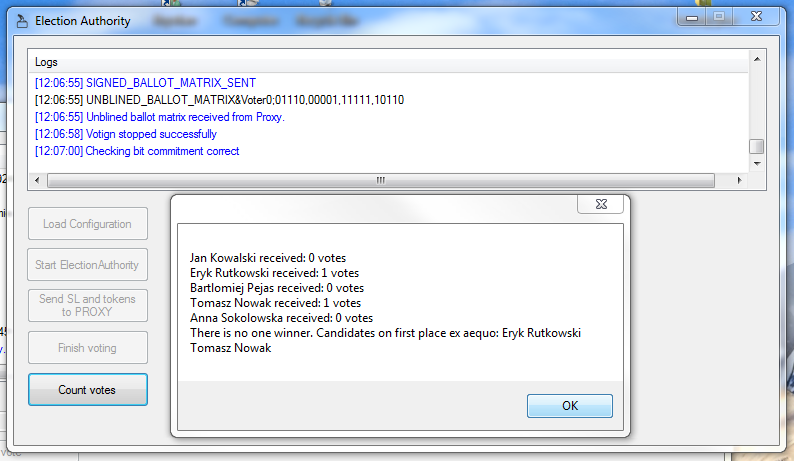
Przed rozpoczęciem głosowania ElectionAuthority przesyła do Auditora – moduł kontrolny – zz zobowiązanie bitowe z permutacji listy kadydatów które będzie przesyłał głosujacym. Po zakończeniu wyborów zobowiązanie bitowe jest sprawdzane czy nie doszło do zmiany, próby modyfikacji permutacji. Jeśli próba fałszerstwa zostanie wykryta, pojawia się odpowiedni monit w konsoli logowania Election Authority.

## Sposób prezentacji wyników głosowania



*Prezentacja wyników głosowania – jednoznaczne wskazanie zwycięzcy*

Na powyższym rysunku przedstawiono wynik procesu głosowania. Po zakończeniu głosowania i naciśnięciu przycisku *Count votes*  Election Authority przystępuje do procesu zliczania głosów. Rezultat wyborów prezentuje w postaci MessageBox’a zawierającego ilość głosów oddanych na poszczególnych kandydatów i wskazanie zwycięzcy wyborów.



*Prezentacja wyników głosowania – brak możliwości jednoznacznego wskazania zwycięzcy*

## Schematy zapewniające bezpieczeństwo i uniemożliwiające fałszerstwo

W projekcie zastosowano następujące schematy, uniemożliwiające oszustwo żadnej ze stron:

* podział karty do głosowania na dwie częsci (spermutowana lista kandydatów oraz losowo wybrane pola tak/nie); za każdą część odpowiada inna jednostka (Election Authority i Proxy) i żadna z nich nie zna informacji przechowywanej w drugiej jednostce,
* potwierdzenie oddania głosu jako część karty do głosowania – potwierdzeniem jest jedynie jedna kolumna z karty i odpowiadające jej liczby (takie jak token, czy podpisana kolumna), co nie pozwala na sprzedaż głosów, ale umożliwia zweryfikowanie oddanego głosu,
* oddany głos przekazywany jest do Proxy jako oddzielny byt, nie mający żadnego związku z listą kandydatów, więc Proxy nie ma możliwości na świadome oszukanie wyborców,
* aby upewnić się, że Election Authority nie zmieni przekazanego głosu z Proxy, wykorzystany jest ślepy podpis na bazie podpisu RSA; wygenerowane na początku pary kluczy publicznych i prywatnych (klucze 1024 bitowe) są odpowiednio wykorzystane przy podpisie, moduł liczby oraz wykładnik klucza publicznego wykorzystano przy zaślepianiu kolumn głosu, zaś wykładnik klucza prywatnego wykorzystano przy podpisie (podpis realizowany przez Election Authority); do zaślepienia wygenerowano również losowe bity (tablica 10 bajtów)
* zobowiązanie bitowe permutacji wykorzystanych przy mieszaniu list kandydatów; aby Election Authority nie mogło zmienić permutacji wykorzystanej na początku, szyfruje je i wysyła do jednostki Auditor; po zakończonych wyborach, Election Authority wysyła klucz prywatny, którym zaszyfrowano permutacje i Auditor sprawdza, czy odszyfrowanie rzeczywiście zwraca użyte permutacje; wykorzystano szyfrowanie RSA

## Testy

Test nr 1

Jeden głosujący oddaje głos na wybranego kandydata – sprawdzenie czy głos zostanie poprawnie zliczony.

Uruchomiono:

* Election Authority (jedna instacja)
* Proxy (jedna instacja)
* Voter (jedna instacja)

Przeprowadzono wszsytkie niezbędne kroki do oddania głosu, w aplikacji Voter oddano głos na wybranego kandydata. Następnie Election Authority zakończył głosowanie i zweryfikowano czy głos został poprawnie zliczony.

Wynik: test zaliczony

Test nr 2

Trzech głosujących oddaje głos na jednego wybranego kandydata – sprawdzamy czy głosy zostaną poprawnie zliczone.

Uruchomione:

* Election Authority (jedna instacja)
* Proxy (jedna instancja)
* Voter (trzy instancje)

Przeprowadzono wszystkie niezbędne kroki do oddania głosu, w aplikacjach Voter oddano głosy na wybranego kandydata. Następnie Election Authority zakończył głosowanie i zweryfikowano czy głos został poprawnie zliczony.

Wynik: test zaliczony

Test nr 3

Trzech głosujących oddaje głos na jednego różnych kandydatów – sprawdzamy czy głosy zostaną poprawnie zliczone. Spodziewany wynik – brak jednoznaczengo wskazania zwycięzcy

Uruchomione:

* Election Authority (jedna instacja)
* Proxy (jedna instancja)
* Voter (trzy instancje)

Przeprowadzono wszystkie niezbędne kroki do oddania głosu, w aplikacjach Voter oddano głosy na wybranego kandydata. Następnie Election Authority zakończył głosowanie i zweryfikowano czy głos został poprawnie zliczony.

Wynik: test zaliczony

Test nr 4

Sprawdzenie czy Auditor weryfikuje zobowiązanie bitowe z listy permutacji po zakończeniu przebiegu głosowania

Uruchomione:

* Election Authority (jedna instacja)
* Proxy (jedna instancja)
* Voter (jedna instancja)

Przeprowadzono wszystkie niezbędne kroki do oddania głosu, w aplikacji Voter oddano głosna wybranego kandydata. Następnie Election Authority zakończył głosowanie i zweryfikowano czy Auditor sprawdził zobowiązanie bitowe z listy permutacji.

Wynik: test zaliczony

Test nr 5

Sprawdzenie czy w pliku *Logs/* pojawiają sie aktualne logi.

Uruchomione:

* Election Authority (jedna instacja)
* Proxy (jedna instancja)
* Voter (jedna instancja)

Przeprowadzono wszystkie niezbędne kroki do oddania głosu, w aplikacji Voter oddano głosna wybranego kandydata. Następnie Election Authority zakończył głosowanie i zweryfikowano czy pliki w *Logs/* sa akutalizowane na bieżąco.

Wynik: test zaliczony

## Repozytorium w serwisie [www.github.com](http://www.github.com)

Nasz projekt został umieszczony na zdalnym repozytorium pod adresem: <https://github.com/kpowojski/PKRY>

## Komentarze kodu źródłowego

Komentarze kodu źródłowego zostały wygenerowane przy użyciu bezpłatengo i open-sourcow’ego narzędzia Doxygen (<http://www.stack.nl/~dimitri/doxygen/>)

Pliki z komentarzami znajduja się w katalogu *Docs/*

* *PKRY\_EA*  - komentarze do aplikacji Election Authority
* *PKRY\_PROXY*  - komentarze do aplikacji Proxy
* *PKRY\_VOTER*  - komentarze do aplikacji Voter