

PKRY Elektroniczne głosowanie



WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

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 poszczególnych modułów jest wczytywana z plików konfiguracyjnych (pliki z rozszerzeniem *.xml) znajdujące się w katalogu *Config*.

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

Opis przykładowego pliku konfiguracyjnego

```
<?xml version="1.0" encoding="utf-8" ?>
<Proxy ID = "Proxy" proxyPort="16000" electionAuthorityIP ="localhost" electionAuthorityPort = "15500"
numberOfVoters = "5" numberOfCandidates = "5"/>
```

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.

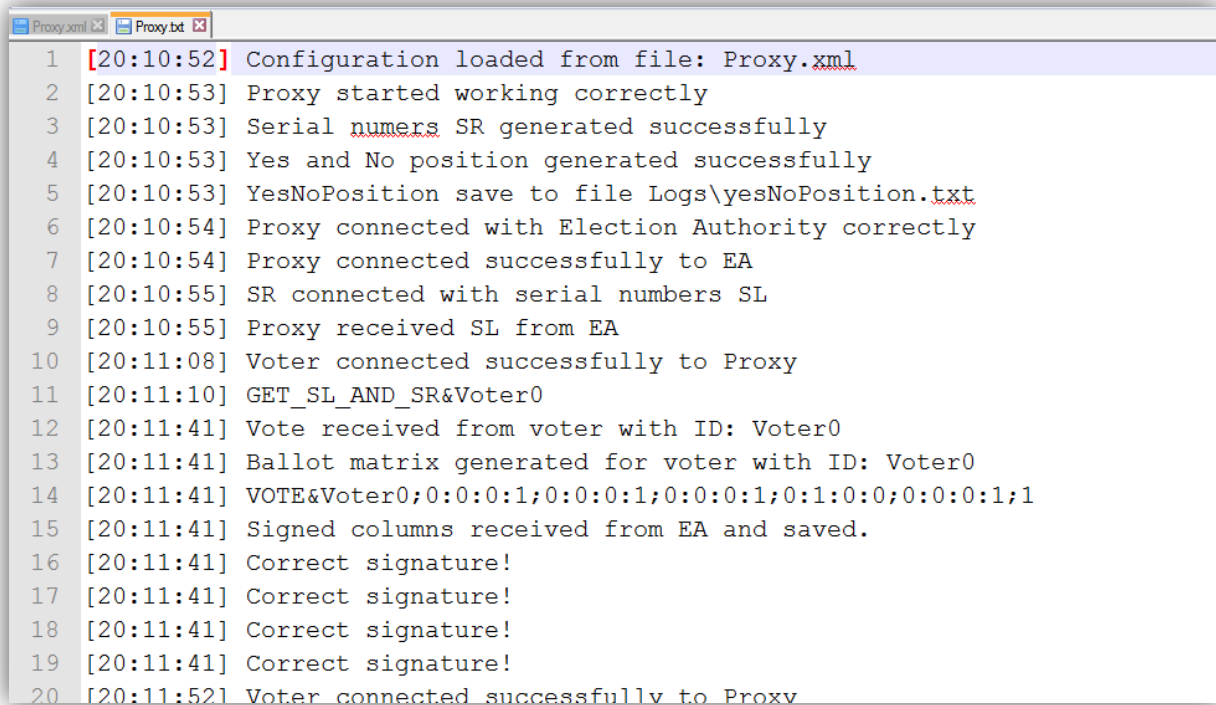
W celu prawidłowego działania projektu sugerowane jest uruchamianie go przy użyciu pliku *run_all.bat*. Dla każdej sesji uruchamiane jest po jednej aplikacji ElectionAuthority i Proxy. Użytkownik zostanie zapytany o ilość aplikacji typu Voter które powinny zostać uruchomione. Po wprowadzeniu żądanej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

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

Logowanie – informacje o tym co dzieje się w aplikacjach

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

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

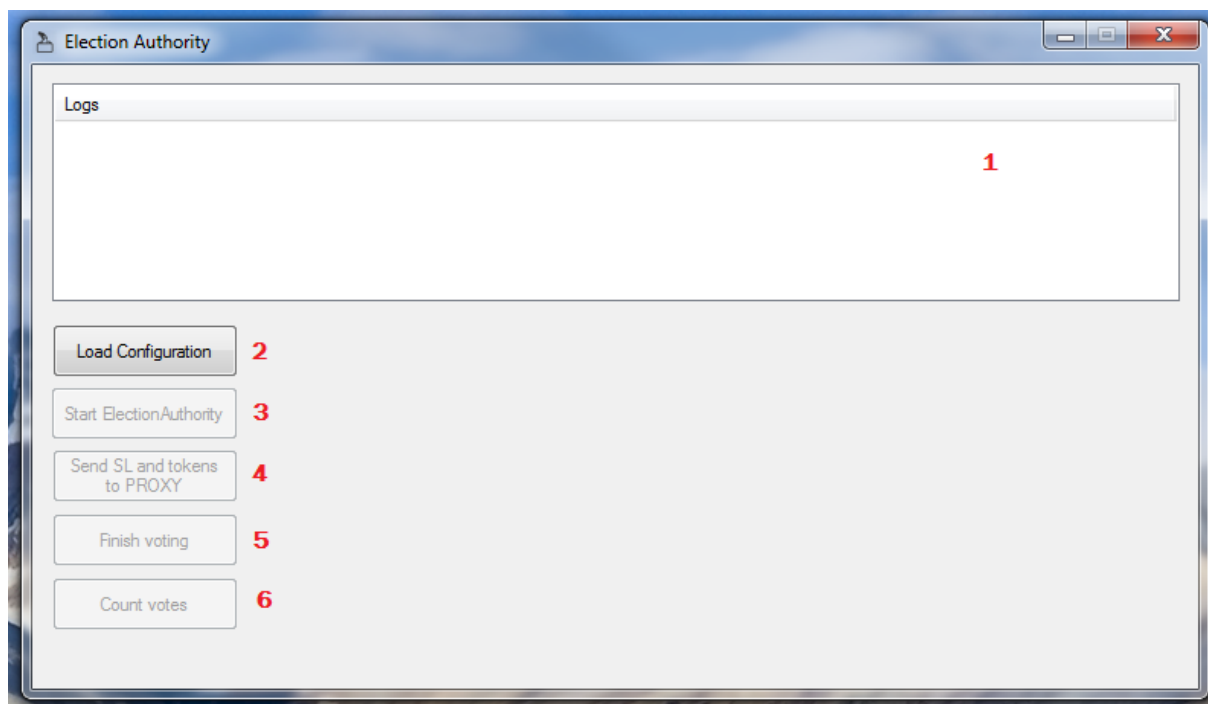


The screenshot shows a text editor window with two tabs: 'Proxy.xml' and 'Proxy.txt'. The 'Proxy.txt' tab is active, displaying a log file with 20 lines of text. Each line starts with a line number (1-20) and a timestamp in brackets, followed by a log message. The messages describe the startup and operation of a proxy system, including configuration loading, connection to an Election Authority (EA), and voter interaction.

```
1 [20:10:52] Configuration loaded from file: Proxy.xml
2 [20:10:53] Proxy started working correctly
3 [20:10:53] Serial numbers SR generated successfully
4 [20:10:53] Yes and No position generated successfully
5 [20:10:53] YesNoPosition save to file Logs\yesNoPosition.txt
6 [20:10:54] Proxy connected with Election Authority correctly
7 [20:10:54] Proxy connected successfully to EA
8 [20:10:55] SR connected with serial numbers SL
9 [20:10:55] Proxy received SL from EA
10 [20:11:08] Voter connected successfully to Proxy
11 [20:11:10] GET_SL_AND_SR&Voter0
12 [20:11:41] Vote received from voter with ID: Voter0
13 [20:11:41] Ballot matrix generated for voter with ID: Voter0
14 [20:11:41] VOTE&Voter0;0:0:0:1;0:0:0:1;0:0:0:1;0:1:0:0;0:0:0:1;1
15 [20:11:41] Signed columns received from EA and saved.
16 [20:11:41] Correct signature!
17 [20:11:41] Correct signature!
18 [20:11:41] Correct signature!
19 [20:11:41] Correct signature!
20 [20:11:52] Voter connected successfully to Proxy
```

Logs/Proxy.txt

Instrukcja 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

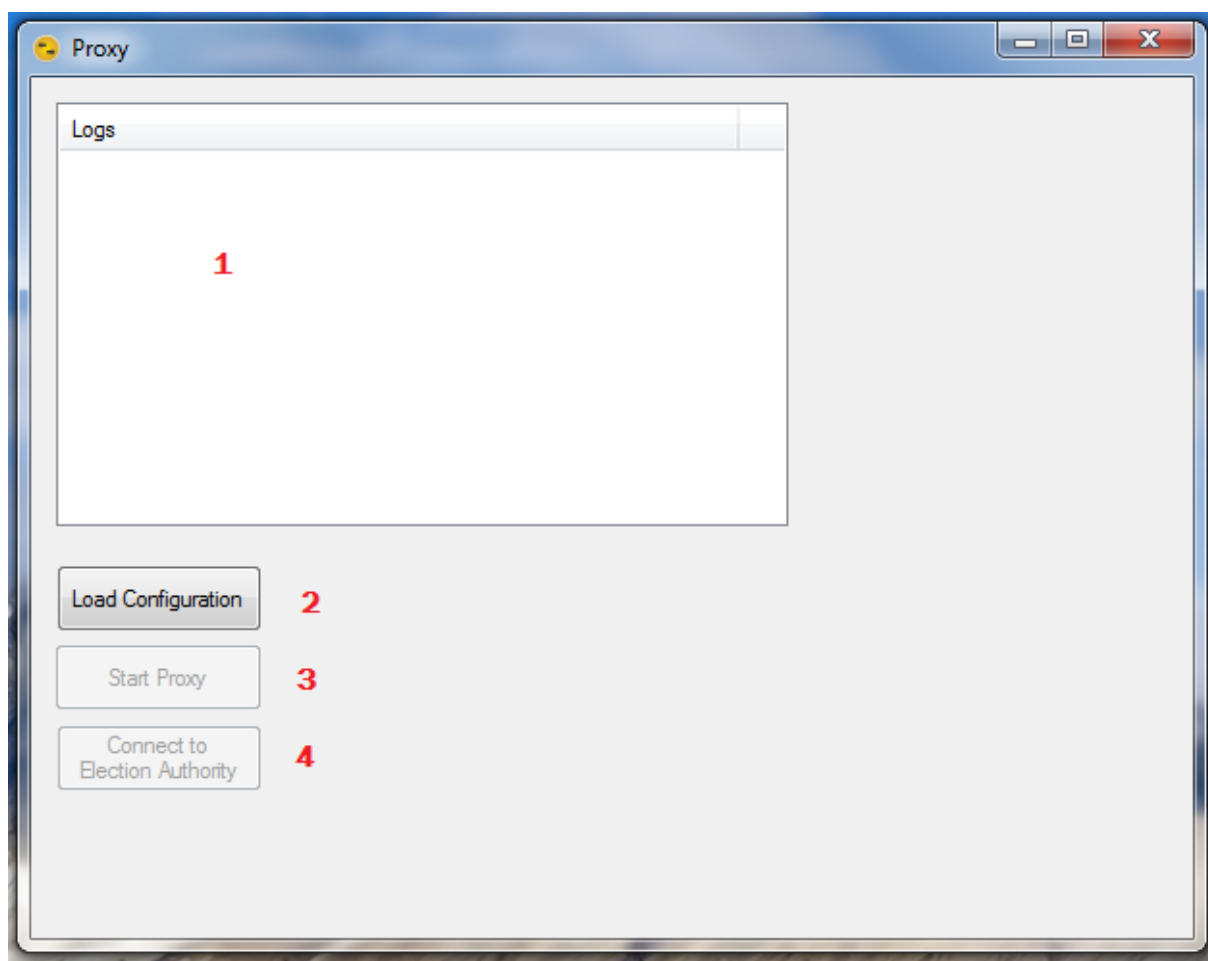
3 – przycisk do uruchomienia Election Authority, przycisk jest dostępny w momencie poprawnego załadowania konfiguracji

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

Instrukcja użytkowania aplikacji Proxy



Opis interfejsu graficznego:

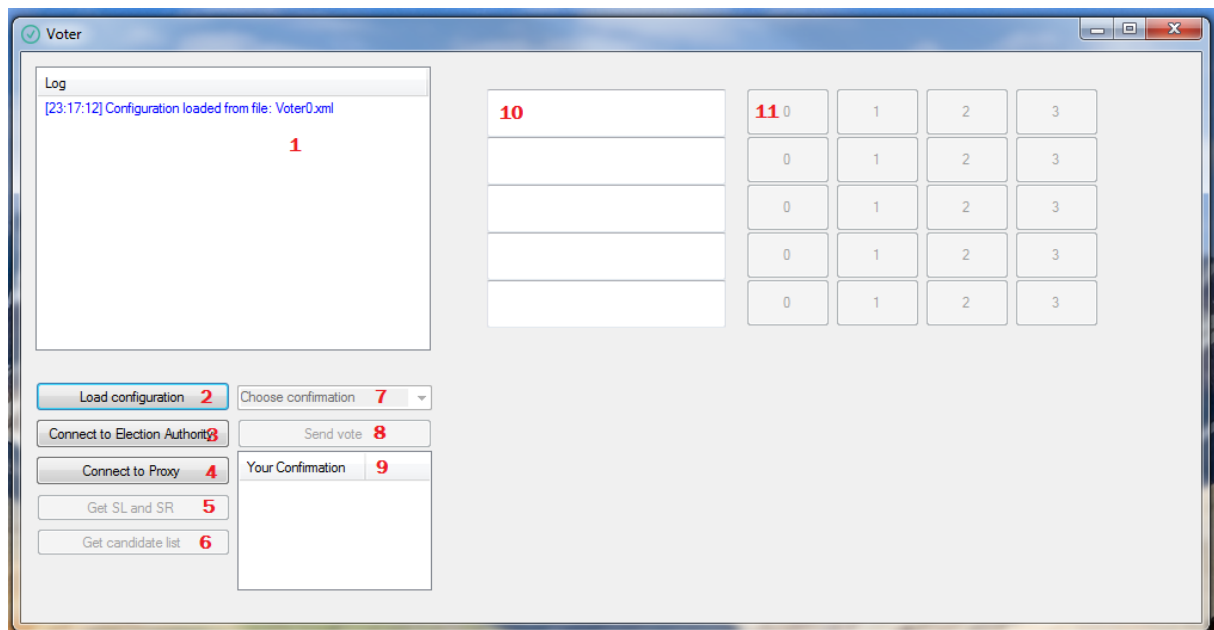
1 – konsola logowania, zawiera wszystkie istotne informacje o zdarzeniach zaistniałych w trakcie działania aplikacji

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 użytkownika aplikacji Voter



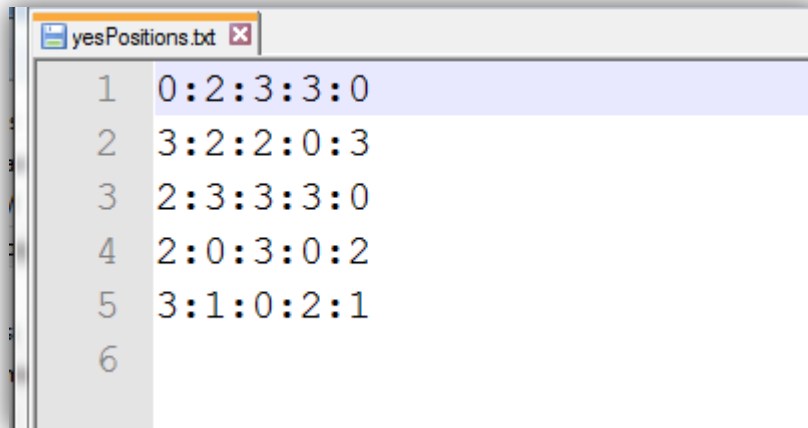
Opis interfejsu graficznego:

- 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 rozwijana umożliwiający wybór kolumny 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 kandydatów
- 11 – przyciski do głosowania

Procedura głosowania

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

Zgodnie z założeniami w konsoli do głosowania (aplikacja Voter) na przyciskach nie są jawnie umieszczone nazwy – zamiast tego w każdym rzędzie mamy przyciski z numerami od 0 do 3. Każdy z głosujących 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.



1	0:2:3:3:0
2	3:2:2:0:3
3	2:3:3:3:0
4	2:0:3:0:2
5	3:1:0:2:1
6	

yesPositions.txt

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

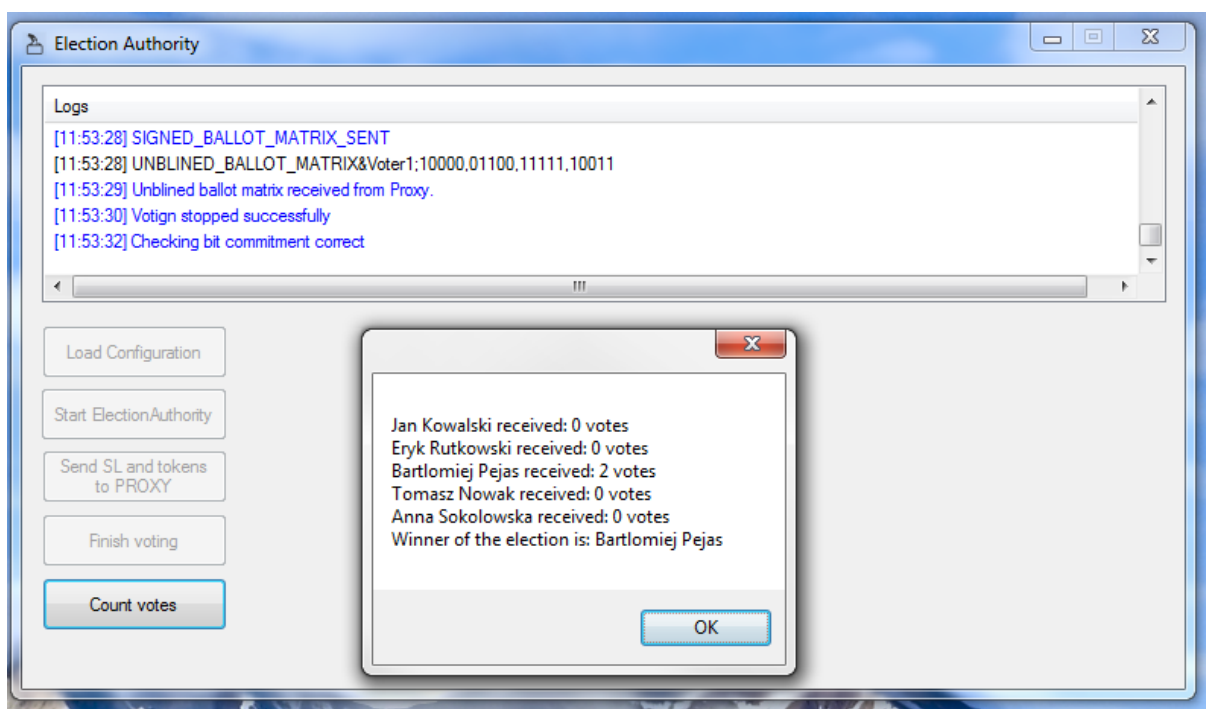
- dla pierwszego kandydata przycisk YES pod przyciskiem z napisem 0
- dla drugiego kandydata przycisk YES pod przyciskiem z napisem 2
- dla trzeciego kandydata przycisk YES pod przyciskiem z napisem 3
- itd

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.

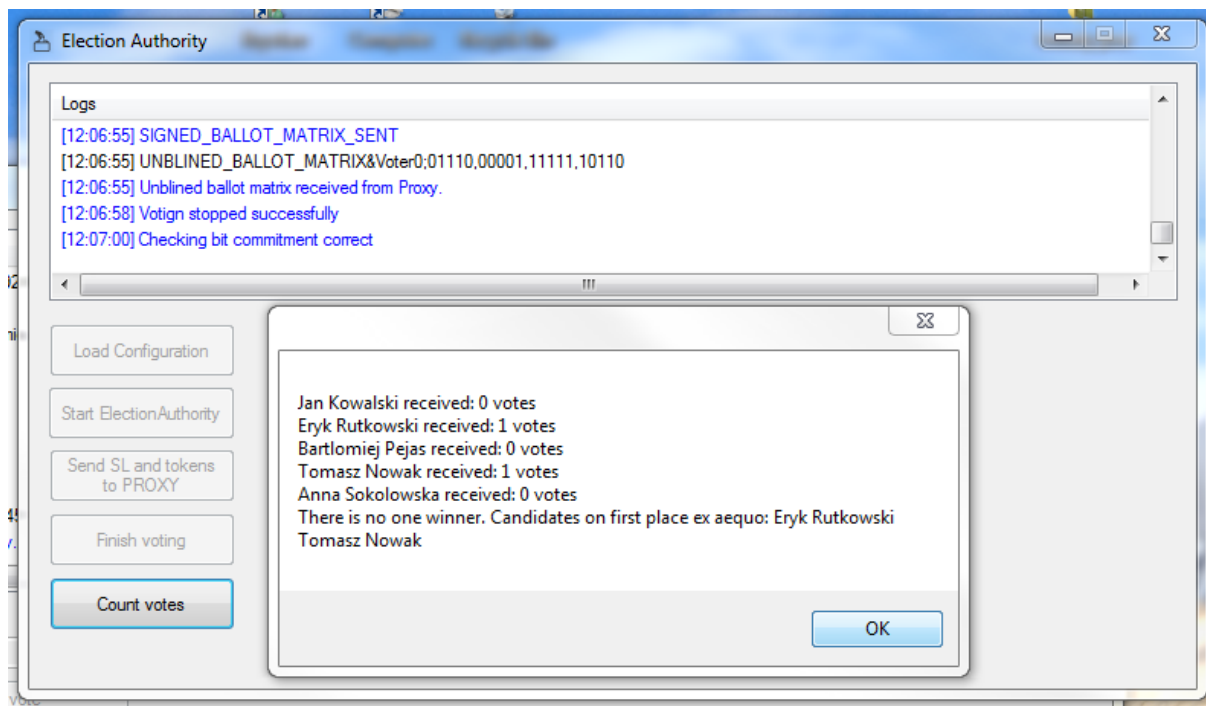
Przed rozpoczęciem głosowania ElectionAuthority przesyła do Auditora – moduł kontrolny – zobowiązanie bitowe z permutacji listy kandydatów które będzie przysyłał głosującemu. Po zakończeniu wyborów zobowiązanie bitowe jest sprawdzane czy nie doszło do zmiany, próby modyfikacji permutacji. Jeśli próba 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 części (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 zaślepieniu 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 instancja)
- Proxy (jedna instancja)
- Voter (jedna instancja)

Przeprowadzono wszystkie 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 instancja)
- 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 jednoznacznego wskazania zwycięzcy

Uruchomione:

- Election Authority (jedna instancja)
- 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 instancja)
- Proxy (jedna instancja)
- Voter (jedna instancja)

Przeprowadzono wszystkie niezbędne kroki do oddania głosu, w aplikacji Voter oddano głos na 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 się aktualne logi.

Uruchomione:

- Election Authority (jedna instancja)
- Proxy (jedna instancja)
- Voter (jedna instancja)

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

Wynik: test zaliczony

Repozytorium w serwisie 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 bezpłatnego i open-sourcow'ego narzędzia Doxygen (<http://www.stack.nl/~dimitri/doxygen/>)

Pliki z komentarzami znajdują się w katalogu *Docs/*

- *PKRY_EA* - komentarze do aplikacji Election Authority
- *PKRY_PROXY* - komentarze do aplikacji Proxy
- *PKRY_VOTER* - komentarze do aplikacji Voter

Dokumentacja ETAP I

Wprowadzenie – opis projektu

Projekt ma na celu zaimplementowanie systemu umożliwiającego przeprowadzenie e-głosowania. Aplikacja spełnia podstawowe wymagania bezpieczeństwa i gwarancji poprawności przebiegu głosowania w tym celu wykorzystany zostanie ślepy podpis. Realizując projekt bazujemy na pomysłach zatytułowanym „Scratch, Click & Vote: E2E voting over the Internet”, autorzy Mirosław Kutylowski, Filip Zagórski, Institute of Mathematics and Computer Science Wrocław University of Technology.

Aplikacja składa się z czterech modułów (podprogramów):

Election Authority (ozn EA) - odpowiada za przygotowanie list kandydatów (każda z list charakteryzuje się unikatową permutacją, wymieszaniem, kandydatów). Ponadto dba o liczenie głosów i zapewnienie głosującemu specjalny token przy pomocy którego może sprawdzić czy jego głos nie został zmieniony.

Proxy - odpowiada za przygotowanie kart do głosowania oraz pośredniczy w przekazywaniu głosu między Voter'em a Election Authority

Voter - odpowiada za prezentację danych otrzymanych od EA (lista kandydatów) oraz Proxy (karta do głosowania) oraz oddanie głosu przez wyborcę.

Auditor - pełni rolę nadzorcy, sprawdza czy w toku głosowania nie doszło do fałszerstwa.

Opracowanie teoretyczne

Projekt zostanie opracowany przy wykorzystaniu języka C# rozszerzonego o bibliotekę kryptograficzną Bouncy Castle [1] przy wykorzystaniu środowiska Microsoft Visual Studio 2010. Całość będzie składać się z oddzielnych 4 aplikacji okienkowych i wykorzystywać architekturę klient-serwer TCP do wzajemnej komunikacji.

Wykorzystane rozwiązania z dziedziny kryptografii:

Bitcommitment [2], pl zobowiązanie bitowe – jest to schemat pozwalający jednej stronie udowodnić niezmiennosć pewnego sekretu (danych, informacji) bez potrzeby ujawniania. Stron zobowiązująca nie może zmienić podanej wartości po dokonaniu zobowiązania. Po ujawnieniu sekretu strona przyjmująca zobowiązanie ma możliwość wykrycia ewentualnych nieprawidłowości wynikających z działania drugiej strony.

Blind signature [3], pl ślepy podpis - rodzaj podpisu cyfrowego w którym zawartość wiadomości jest zaślepiana przed podpisaniem. Ślepy podpis może być potem zweryfikowany z wiadomością. Są na ogół wykorzystywane w protokołach opartych na prywatności, w których autor wiadomości i podpisujący to różne osoby. Wykorzystywany jest w celu zapewnienia uczciwości jednej ze stron przy jednoczesnym zachowaniu tajności informacji.

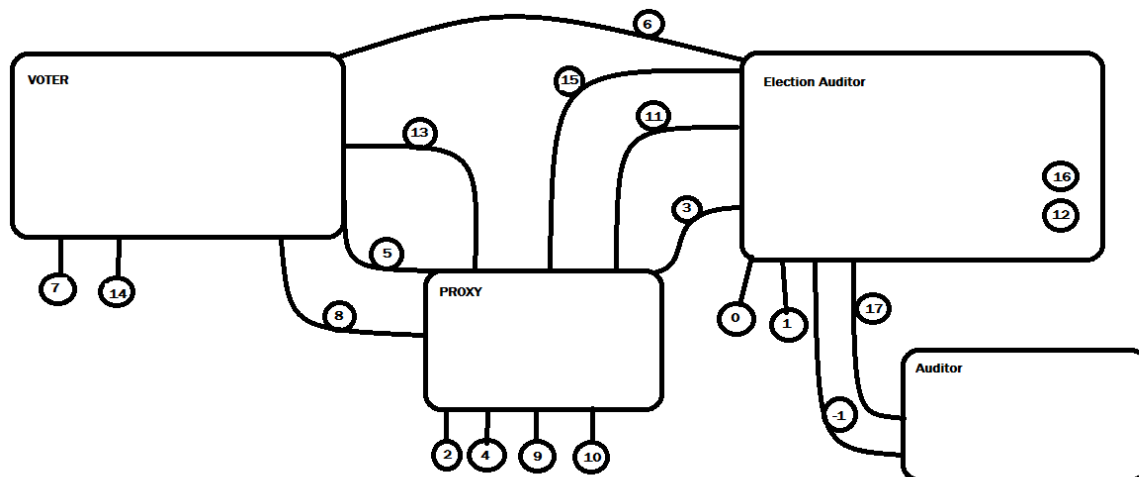
Koncepcja rozwiązania problemu [4]

SL – numer identyfikujący listę kandydatów

SR – numer karty do głosowania

π – permutacja listy kandydatów

tokeny – służą do podpisania kolumn w macierzy głosów



Opis schematu:

0 : EA wczytuje standardową listę kandydatów

1: EA generuje dla każdej karty z osobna: SL, π oraz tokeny (A,B,C,D) i przesyła listę SL i bitcommitment od π do Auditora

2: Proxy generuje dla każdej karty do głosowania SR oraz losowe pozycje „tak”. Następnie zapisuje wektor pozycji „tak” w pliku, tak aby głosujący mógł przyłożyć do listy kandydatów.

3: EA przekazuje do Proxy: SL oraz tokeny

4: Proxy paruje SL oraz odpowiadający mu SR

5: Proxy przekazuje pary SL i SR do Votera

6: Voter pobiera kartę z kandydatami na podstawie numeru SL

7: Voter dokonuje głosu

8: Głos (jako zero-jedynkowa tablica dwuwymiarowa) przekazywany jest do Proxy

- 9: Proxy przekształca głos na tzw. „ballot matrix”, czyli zaznacza wszystkie te pola “nie”, które nie zostały kliknięte przez Voter
- 10: Proxy zaślepia „ballot matrix” (para kluczy generowana losowo) [5]
- 11: Proxy przesyła: SL, tokeny oraz zaślepiony „ballot matrix”
- 12: EA podpisuje zaślepioną „ballot matrix” następnie zwraca SL, tokeny i zaślepioną, podpisaną ballot matrix
- 13: Proxy wysyła podpisaną kolumnę (wybraną przez głosującego jako potwierdzenie) do Votera
- 14: Voter wybiera kolumnę którą chce stosować jako potwierdzenie - token, kolumna, podpis od EA
- 15: Proxy przesyła odślepioną, podpisaną „ballot matrix” i odpowiadający jej SL do Election Auditora
- 16: EA odpermutowuje i liczy głosy
- 17: Auditor dostaje odślepione π od EA i dokonuje sprawdzenie czy nie uległo ono zmianie

Bibliografia

- [1] <http://www.bouncycastle.org/>
- [2] <http://www.cs.berkeley.edu/~daw/teaching/cs276-s04/19a.ps>
- [3] Z. Kotulski, Wykłady z przedmotu Protokoły Kryptograficzne, Politechnika Warszawska, 2014
- [4] M. Kutylowski, F. Zagórski Scratch, Click & Vote: E2E voting over the Internet, Politechnika Wrocławska
- [5] R. Bellare, Cryptography: Authentication, Blind Signatures, and Digital Cash