Election Authority

Generated by Doxygen 1.8.9.1

Fri Jan 9 2015 20:50:16

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 Package ElectionAuthority

Classes

- class Auditor

  class is used to verify if EA did not cheat
- class Ballot

  class represents one ballot and has information about it
- class CandidateList

  class loads candidate list from txt file
- class Configuration

  loading config from txt file
- class Constants

  Constants used in project
- class ElectionAuthority

  Election authority class - responsible for generating serial numbers(SL, SR and numbers connected to them) and counting votes; main class in e-voting project
- class **Extentions**

  additional function for our program
- class Form1

  Class which shows a GUI
- class Logs

  allows to collect and display logs
- class Parser

  parsing messages recived form clients
- class Permutation

  represents all permutation's method
- class **Program**
- class SerialNumberGenerator

  Generates serial numbers used in EA
- class Server

Chapter 5

Class Documentation

5.1 ElectionAuthority.Auditor Class Reference

class is used to verify if EA did not cheat

Public Member Functions

- Auditor (Logs logs)
  Auditor's constructor
- bool checkPermutation (RsaKeyParameters privateKey, RsaKeyParameters publicKey, BigInteger[ ] explicit↩ Permutation)
  checking the correctness of permutation

Properties

- BigInteger[ ] CommitedPermatation `[get, set]`
  set and get commited permutation

5.1.1 Detailed Description

class is used to verify if EA did not cheat

5.1.2 Constructor & Destructor Documentation

5.1.2.1 ElectionAuthority.Auditor.Auditor (Logs *logs*) `[inline]`

Auditor's constructor

Parameters

logs	transfered log instance

5.1.3 Member Function Documentation

5.1.3.1 bool ElectionAuthority.Auditor.checkPermutation (RsaKeyParameters *privateKey,* RsaKeyParameters *publicKey,* BigInteger[ ] *explicitPermutation*) `[inline]`

checking the correctness of permutation

Parameters

privateKey	private key used for bit commitment
publicKey	public key used for bit commitment
explicit↩ Permutation	used permutation (as open text)

Returns

5.1.4 Property Documentation

5.1.4.1 BigInteger [ ] ElectionAuthority.Auditor.CommitedPermatation [get],[set]

set and get commited permutation

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Auditor.cs

5.2 ElectionAuthority.Ballot Class Reference

class represents one ballot and has information about it

Public Member Functions

- Ballot (BigInteger SL)
  ballot's constructor
- void signColumn ()
  Method to sing each column in ballotMatrix

Properties

- BigInteger **SL** [get]
- List< BigInteger > **TokenList** [get, set]
- List< BigInteger > **ExponentsList** [get, set]
- List< BigInteger > **SignatureFactor** [get, set]
- BigInteger[ ] **SignedColumn** [get]
- BigInteger[ ] **BlindColumn** [set]
- string[,] **UnblindedBallot** [get, set]
- List< BigInteger > **Permutation** [get, set]
- List< BigInteger > **InversePermutation** [get, set]

5.2.1 Detailed Description

class represents one ballot and has information about it

5.2.2 Constructor & Destructor Documentation

5.2.2.1 ElectionAuthority.Ballot.Ballot (BigInteger *SL*) [inline]

ballot's constructor

Parameters

SL	serial (list of candidate) number

5.2.3 Member Function Documentation

5.2.3.1 void ElectionAuthority.Ballot.signColumn ( ) [inline]

Method to sing each column in ballotMatrix

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Ballot.cs

5.3 ElectionAuthority.CandidateList Class Reference

class loads candidate list from txt file

Public Member Functions

- CandidateList (Logs logs)
  condidate list constructor
- List< string > loadCanidateList (string path)
  loading cadidate list
- string getPathToCandidateList (string path)
  gets path to txt file with candidates list

5.3.1 Detailed Description

class loads candidate list from txt file

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ElectionAuthority.CandidateList.CandidateList (Logs *logs*) [inline]

condidate list constructor

Parameters

logs	logs instance

5.3.3 Member Function Documentation

5.3.3.1 string ElectionAuthority.CandidateList.getPathToCandidateList (string *path*) [inline]

gets path to txt file with candidates list

Parameters

path	path to txt file

Returns

path to file

5.3.3.2 List<string> ElectionAuthority.CandidateList.loadCanidateList (string *path*) [inline]

loading cadidate list

Parameters

path	path to txt file

Returns

List of strings with candidates

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/CandidateList.cs

5.4 ElectionAuthority.Configuration Class Reference

loading config from txt file

Public Member Functions

- **Configuration** (Logs logs)
- bool **loadConfiguration** (string path)

Properties

- string **ElectionAuthorityID** [get]
- string **ElectionAuthorityPortClient** [get]
- string **ElectionAuthorityPortProxy** [get]
- string **NumberOfVoters** [get]

5.4.1 Detailed Description

loading config from txt file

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Configuration.cs

5.5 ElectionAuthority.Constants Class Reference

Constants used in project

Public Attributes

- const int **BALLOT_SIZE** = 4
- const int **LOG_INFO** = 0
- const int **LOG_MESSAGE** = 1
- const int **LOG_ERROR** = 2
- const string **ID** = "ID"
- const string **ELECTION_AUTHORITY_PORT_CLIENT** = "electionAuthorityPortForClient"
- const string **ELECTION_AUTHORITY_PORT_PROXY** = "electionAuthorityPortForProxy"
- const string **CONFIGURATION_LOADED_FROM** = "Configuration loaded from file: "
- const string **NUMBER_OF_VOTERS** = "numberOfVoters"
- const string **PATH_TO_CONFIG** = @"Config\ElectionAuthority.xml"
- const string **CANDIDATE_LIST** = "CandidateList.xml"
- const string **SERVER_STARTED_CORRECTLY** = "Election Authority started working correctly"
- const string **SERVER_UNABLE_TO_START** = "Election Authority unable to start working"
- const string **UNKNOWN** = "Unknown"
- const string **DISCONNECTED_NODE** = "Someone has been disconnected"
- const string **CANDIDATE_LIST_SUCCESSFUL** = "Candidate list loaded successfully"
- const string **PERMUTATION_GEN_SUCCESSFULLY** = "Permuration generated successfully"
- const string **SERIAL_NUMBER_GEN_SUCCESSFULLY** = "Serial number list generated successfully"
- const string **SL_CONNECTED_WITH_PERMUTATION** = "Serial numbers connected with permutation"
- const int **NUMBER_OF_BITS_SL** = 64
- const int **NUMBER_OF_TOKENS** = 4
- const string **TOKENS_GENERATED_SUCCESSFULLY** = "Tokens generated successfully"
- const int **NUMBER_OF_BITS_TOKEN** =512
- const string **SL_CONNECTED_WITH_TOKENS** = "Serial numbers connected with tokens"
- const string **SL_RECEIVED_SUCCESSFULLY** = "SL_RECEIVED_SUCCESSFULLY"
- const string **SL_AND_SR_SENT_SUCCESSFULLY** = "SL sent successfully to Proxy"
- const string **PROXY** ="PROXY"
- const string **GET_CANDIDATE_LIST** = "GET_CANDIDATE_LIST"
- const string **CANDIDATE_LIST_RESPONSE** = "CANDIDATE_LIST_RESPONSE"
- const string **BLIND_PROXY_BALLOT** = "BLIND_PROXY_BALLOT"
- const string **BLIND_PROXY_BALLOT_RECEIVED** = "Blind ballot received from voter with ID: "
- const string **SIGNED_PROXY_BALLOT** = "SIGNED_PROXY_BALLOT"
- const string **SIGNED_BALLOT_MATRIX_SENT** = "SIGNED_BALLOT_MATRIX_SENT"
- const string **GENERATE_INVERSE_PERMUTATION** = "Inverse permutation generated"
- const string **SL_CONNECTED_WITH_INVERSE_PERMUTATION** = "Serial numbers connected with inverse permutation"
- const string **UNBLINED_BALLOT_MATRIX** = "UNBLINED_BALLOT_MATRIX"
- const string **UNBLINED_BALLOT_MATRIX_RECEIVED** = "Unblined ballot matrix received from Proxy."
- const string **BIT_COMMITMENT_OK** = "Checking bit commitment correct"
- const string **BIT_COMMITMENT_FAIL** = "Checking bit commitment incorrect"
- const string **UNABLE_TO_STOP_VOTING** = "UNABLE_TO_STOP_VOTING"
- const string **VOTIGN_STOPPED** = "Votign stopped successfully"

Static Public Attributes

- static string **SL_TOKENS** = "SL_TOKENS"
- static string **CONNECTED** = "CONNECTED"

5.5.1 Detailed Description

Constants used in project

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Constants.cs

5.6 ElectionAuthority.ElectionAuthority Class Reference

Election authority class - responsible for generating serial numbers(SL, SR and numbers connected to them) and counting votes; main class in e-voting project

Public Member Functions

- ElectionAuthority (Logs logs, Configuration configuration, Form1 form)

  Constructor of EA
- void loadCandidateList (string pathToElectionAuthorityConfig)

  loading cadidate list from file
- void generateDate ()

  generates data for voting (serial numbers, tokens, permutations)
- void sendSLAndTokensToProxy ()

  Sends SL and Tokens to proxy
- void disableSendSLTokensAndTokensButton ()

  disable button causes sending tokens an SLs
- void getCandidateListPermuated (string name, BigInteger SL)

  permutes candidate list for concrete voter and for his/her SL number
- void saveBlindBallotMatrix (string message)

  saves blind ballot matrix recived from proxy
- void saveUnblindedBallotMatrix (string message)

  saves unblinded ballot matrix (vote)
- void disbaleProxy ()

  disables proxy
- void countVotes ()

  counting votes EA send to voter unblinded permutation (and then private key) so Audiotr can check RSA formula
- void blindPermutation (List< List< BigInteger >> permutationList)

  blinds permutations (all of them), RSA formula (bit commitment)
- void unblindPermutation (List< List< BigInteger >> permutationList)

  unblind permutation, checking permutations RSA (auditor checks all of the permutations)

Properties

- Server **ServerClient** `[get]`
- Server **ServerProxy** `[get]`

5.6.1 Detailed Description

Election authority class - responsible for generating serial numbers(SL, SR and numbers connected to them) and counting votes; main class in e-voting project

5.6.2 Constructor & Destructor Documentation

5.6.2.1 ElectionAuthority.ElectionAuthority.ElectionAuthority (Logs *logs,* Configuration *configuration,* Form1 *form*) [inline]

Constructor of EA

Parameters

logs	logs instance
configuration	configuration loaded
form	application form

5.6.3 Member Function Documentation

5.6.3.1 void ElectionAuthority.ElectionAuthority.blindPermutation (List< List< BigInteger >> *permutationList*) [inline]

blinds permutations (all of them), RSA formula (bit commitment)

Parameters

permutationList	permutation list

5.6.3.2 void ElectionAuthority.ElectionAuthority.countVotes ( ) [inline]

counting votes EA send to voter unblinded permutation (and then private key) so Audiotr can check RSA formula

5.6.3.3 void ElectionAuthority.ElectionAuthority.disableSendSLTokensAndTokensButton ( ) [inline]

disable button causes sending tokens an SLs

5.6.3.4 void ElectionAuthority.ElectionAuthority.disbaleProxy ( ) [inline]

disables proxy

5.6.3.5 void ElectionAuthority.ElectionAuthority.generateDate ( ) [inline]

generates data for voting (serial numbers, tokens, permutations)

5.6.3.6 void ElectionAuthority.ElectionAuthority.getCandidateListPermuated (string *name,* BigInteger *SL*) [inline]

permutes candidate list for concrete voter and for his/her SL number

Parameters

name	
SL	

5.6.3.7 void ElectionAuthority.ElectionAuthority.loadCandidateList (string *pathToElectionAuthorityConfig*) [inline]

loading cadidate list from file

Parameters

pathToElection↩ AuthorityConfig	path to EA configuration file

5.6.3.8 void ElectionAuthority.ElectionAuthority.saveBlindBallotMatrix (string *message*) [inline]

saves blind ballot matrix recived from proxy

Parameters

message	

5.6.3.9 void ElectionAuthority.ElectionAuthority.saveUnblindedBallotMatrix (string *message*) [inline]

saves unblinded ballot matrix (vote)

Parameters

message	string message recived from proxy

5.6.3.10 void ElectionAuthority.ElectionAuthority.sendSLAndTokensToProxy ( ) [inline]

Sends SL and Tokens to proxy

5.6.3.11 void ElectionAuthority.ElectionAuthority.unblindPermutation (List< List< BigInteger >> *permutationList*) [inline]

unblind permutation, checking permutations RSA (auditor checks all of the permutations)

Parameters

permutationList	permutation list

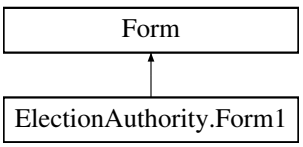
The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/ElectionAuthority.cs

5.7 ElectionAuthority.Form1 Class Reference

Class which shows a GUI

Inheritance diagram for ElectionAuthority.Form1:



Public Member Functions

- Form1 ()

  constructor which creates Graphical User interface
- void disableSendSLTokensAndTokensButton ()

  disable sendSLAndTokensButton and enable finishVoting button

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

5.7.1 Detailed Description

Class which shows a GUI

5.7.2 Constructor & Destructor Documentation

5.7.2.1 ElectionAuthority.Form1.Form1 ( ) [inline]

constructor which creates Graphical User interface

5.7.3 Member Function Documentation

5.7.3.1 void ElectionAuthority.Form1.disableSendSLTokensAndTokensButton ( ) [inline]

disable sendSLAndTokensButton and enable finishVoting button

5.7.3.2 override void ElectionAuthority.Form1.Dispose (bool *disposing*) [inline], [protected]

Clean up any resources being used.

Parameters

disposing	true if managed resources should be disposed; otherwise, false.

The documentation for this class was generated from the following files:

- ElectionAuthority/ElectionAuthority/Form1.cs
- ElectionAuthority/ElectionAuthority/Form1.Designer.cs

5.8 ElectionAuthority.Logs Class Reference

allows to collect and display logs

Public Member Functions

- Logs (ListView logsListView)
  Logs instance's constructor
- void addLog (string log, bool time, int flag, bool anotherThread=false)
  adds log

5.8.1 Detailed Description

allows to collect and display logs

5.8.2 Constructor & Destructor Documentation

5.8.2.1 ElectionAuthority.Logs.Logs (ListView *logsListView*) [inline]

Logs instance's constructor

Parameters

logsListView	logs list view

5.8.3 Member Function Documentation

5.8.3.1 void ElectionAuthority.Logs.addLog (string *log*, bool *time*, int *flag*, bool *anotherThread* = `false`) [inline]

adds log

Parameters

log	log message
time	if print time
flag	type of message (error, info...)
anotherThread	thread flag

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Logs.cs

5.9 ElectionAuthority.Parser Class Reference

parsing messages recived form clients

Public Member Functions

- Parser (Logs logs, ElectionAuthority electionAuthority)
  parser's constructor
- bool parseMessage (string msg)
  parses message

5.9.1 Detailed Description

parsing messages recived form clients

5.9.2 Constructor & Destructor Documentation

5.9.2.1 ElectionAuthority.Parser.Parser (Logs *logs*, ElectionAuthority *electionAuthority*) [inline]

parser's constructor

Parameters

logs	log instance
electionAuthority	election authority instance

5.9.3 Member Function Documentation

5.9.3.1 bool ElectionAuthority.Parser.parseMessage (string *msg*) [inline]

parses message

Parameters

msg	recived message

Returns

parsing result

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Parser.cs

5.10 ElectionAuthority.Permutation Class Reference

represents all permutation's method

Public Member Functions

- Permutation (Logs logs)
  constructor
- List< BigInteger > generatePermutation (int candidateQuantity)
  generate ONE permutation
- List< BigInteger > getInversePermutation (List< BigInteger > permutation)
  Find inverse permuatation using a table method

5.10.1 Detailed Description

represents all permutation's method

5.10.2 Constructor & Destructor Documentation

5.10.2.1 ElectionAuthority.Permutation.Permutation (Logs *logs*) [inline]

constructor

Parameters

logs	logs instance

5.10.3 Member Function Documentation

5.10.3.1 List<BigInteger> ElectionAuthority.Permutation.generatePermutation (int *candidateQuantity*) [inline]

generate ONE permutation

Parameters

candidate↩ Quantity	quantity of candidates

Returns

5.10.3.2 List<BigInteger> ElectionAuthority.Permutation.getInversePermutation (List< BigInteger > *permutation*) [inline]

Find inverse permuatation using a table method

Parameters

permutation	permutation to inverse

Returns

inverse permutation

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Permutation.cs

5.11 ElectionAuthority.SerialNumberGenerator Class Reference

Generates serial numbers used in EA

Static Public Member Functions

- static List< BigInteger > generateListOfSerialNumber (int numberOfSerials, int numberOfBits)
  generate SL for election
- static List< AsymmetricCipherKeyPair > generatePreTokens (int numberOfSerials, int numberOfBits)
  generate pre tokens (key pair) for election

5.11.1 Detailed Description

Generates serial numbers used in EA

5.11.2 Member Function Documentation

5.11.2.1 static List<BigInteger> ElectionAuthority.SerialNumberGenerator.generateListOfSerialNumber (int *numberOfSerials*, int *numberOfBits*) [inline], [static]

generate SL for election

Parameters

numberOfSerials	number of serials to generate
numberOfBits	bit size of serial

Returns

list of serial numbers

5.11.2.2 static List<AsymmetricCipherKeyPair> ElectionAuthority.SerialNumberGenerator.generatePreTokens (int *numberOfSerials*, int *numberOfBits*) [inline], [static]

generate pre tokens (key pair) for election

Parameters

numberOfSerials	number of serials to generate
numberOfBits	bit size of serial

Returns

list of pre tokens

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/SerialNumberGenerator.cs

5.12 ElectionAuthority.Server Class Reference

Public Member Functions

- Server (Logs logs, ElectionAuthority electionAuthority)
  server which allows to communicate with other processes
- bool startServer (string port)
  allow to start server
- void stopServer ()
  stops server
- void sendMessage (string name, string msg)
  sends message to client

5.12.1 Constructor & Destructor Documentation

5.12.1.1 ElectionAuthority.Server.Server (Logs *logs*, ElectionAuthority *electionAuthority*) [inline]

server which allows to communicate with other processes

Parameters

logs	allows to collect and display logs - information in console
electionAuthority	represents class where is main logic of application

5.12.2 Member Function Documentation

5.12.2.1 void ElectionAuthority.Server.sendMessage (string *name*, string *msg*) [inline]

sends message to client

Parameters

name	name of client which we want to send a message
msg	message which we want to send

5.12.2.2 bool ElectionAuthority.Server.startServer (string *port*) [inline]

allow to start server

Parameters

port	number of port on which server is running, this information comes from configuration xml file

Returns

returns true when server started successfully

5.12.2.3 void ElectionAuthority.Server.stopServer ( ) [inline]

stops server

The documentation for this class was generated from the following file:

- ElectionAuthority/ElectionAuthority/Server.cs

Index