Grant Agreement Number: 825225

Safe-DEED

www.safe-deed.eu

Implementation of Cryptographic Building Blocks and Specialized Protocols v1/3

Deliverable number	*D5.5*
Dissemination level	*Public*
Delivery data	*due 29.11.2019*
Status	*Final*
Authors	*Lukas Helminger, Fabian Schmid, Roman Walch*

Horizon 2020
European Union Funding
for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825225.

Executive Summary

In this deliverable, we describe the implementation of a new cryptographic primitive called multi-party public-key accumulator (MPC-accumulator), which was developed in the Safe-DEED project. An MPC-accumulators is a basic building block for privacy-enhanced technology. This document should enable security experts and software engineers to use MPC-accumulators in their applications. We fully describe the functionality of MPC-accumulators and their possible applications in deliverable D5.3. Since the theoretical description of a cryptographic primitive alone is not enough to get wide acceptance by the community we implemented MPC-accumulators using an open-source MPC-framework.

The implementation uses advanced cryptographic primitives like elliptic curve cryptography and secret sharing. Therefore, we are not only providing the source code but comprehensive documentation of the code. In addition, we give easy to follow instructions on how to download, install, and edit the provided source code of MPC-accumulators.

Table of Contents

1 Introduction

In deliverable D5.3, we developed a new cryptographic primitive called multi-party public-key accumulators (MPC-accumulators). To make MPC-accumulators available to the community, we also implemented them with an open-source MPC-framework. Since we choose to use elliptic curves as a building block, we are talking about elliptic curve accumulators. This deliverable aims to provide a quick overview on how to download, install, and edit the provided source code of the elliptic curve accumulator running with Secure Multi-Party Computation. The code is written in Java and licensed under the open-source MIT license[1]. This document is organized as follows.

First, there will be a short description of the dependencies. Secondly, in Section 2, we will look at a guide on how to build and run the demonstration. Then there will be an elaboration of the program structure (Section 3), the changes made to FRESCO [1], and the location of the code of this contribution. In the end, we provide the full API of our elliptic curve accumulator (Section A).

1.1 Dependencies

1.1.1 FRESCO

For the proof-of-concept implementation of our protocols, we used FRESCO. We chose FRESCO because it allows fast prototyping of MPC protocols, is open-source, and has an active developer community. Since we needed to adapt the SPDZ protocol [3, 2] to be able to run elliptic curve operations, we made a lot of changes to the framework. Thus we provide an altered version of FRESCO ourselves, where we omitted parts we do not need and added our changed files.

1.1.2 IAIK ECCelerate

Our implementation combines Secure Multi-Party Computation with elliptic curve cryptography. As mentioned above, we rely on FRESCO for our MPC computations. For the elliptic curve calculations, we used the IAIK ECCelerate library [4] and the Java Cryptography Extension (JCE), previously developed by our team at the TU Graz. It is, therefore, necessary to get a license of the ECCerlerate, to run our demonstration. For research purposes, there exists educational license [2]. There is also an ECCelerate add-on, this includes speedups for the ECC computations. The curves used by the add-on may be patented in certain countries. When it is ensured that these curves may be used, one can integrate the add-on as well.

2 The Build Process

The project is developed using maven[3]. When using maven as a build manager, one has to install the ECCelerate .jar file in the local maven repository using the following command.

[1] `https://mit-license.org/`

[2] `https://jce.iaik.tugraz.at/sic/Sales/Licences/Educational`

[3] `https://maven.apache.org/`

```
mvn install:install-file -Dfile=<path-to-file> -DgroupId=<
    group-id> -DartifactId=<artifact-id> -Dversion=<version> -
    Dpackaging=<packaging>
```

- `<path-to-file>` has to specify the path to the `.jar` file (the java archive)
- `<group-id>` has to specify the group, in our case `iaik`.
- `<artifact-id>` the name under which the artifact will be installed: `iaik_eccelerate` for the ECC library.
- `<version>` This demonstrator was developed and tested under version `6.0`.
- `<packaging>` When following the instruction above one has to specify the packaging as `jar`.

Note that the demonstrator also depends on a JCE library , developed at IAIK. If one wants to run the program without this dependency, remove it from the `suite/spdz/pom.xml` and make sure to provide a different JCE framework. The ECCelerate add-on is an optional improvement to the ECC library. This add-on can be installed as described above. After installing it to the local maven repository, it has to be added as a dependency is the `suite/spdz/pom.xml` file.

Following the previous steps, one has installed the ECCelerate library in the local maven repository. Now the build process can be started. Navigate to the demos/MPC-ACC directory to find the Makefile for this demonstrator. There are several make targets. To make a full and clean installation, execute the command `make build`. This command will install FRESCO, the ECC library, and the demonstrator. Then the assembled jar file will be put in a specific folder so that the run target will find it later.

3 The Program Structure

The implementation of the demonstrator is embedded in the demo folder of FRESCO. We needed to adapt FRESCO to our needs in terms of elliptic curve operations. However, we kept the maven directory structure.

Core: The Core folder holds the foundation of the FRESCO framework. The general layout and fundamental definitions are given there. In there, we merely added some output functions to the Socket Network class. These output functions were needed to perform benchmarking.

Suite: FRESCO implements multiple MPC protocols. These protocols are found in the suite directory. For our demonstration, we needed the SPDZ protocol suite, since it provides security against malicious adversaries and allows more than two parties to participate in the protocol. For compactness, we omitted the other protocol suites from code. The dependencies of our adaptations can be found in the `pom.xml` file of SPDZ. Our Adaptations themselves are located in the ECCExtensions directory in the SPDZ implementation.

Tools: Tools contains various protocols needed by the protocol suites. The mascot protocol and oblivious transfer protocol can be found here. The mascot protocol had to be adapted to the changes made in the SPDZ implementation.

Demos: Here in the `MPC-ACC` directory, the actual MPC elliptic curve accumulator is positioned. When the project is successfully built, the project can be run by the provided execution scripts. As a starting point, one might look at the `make run` target. Then one could look into the scripts facilitating execution in a distributed environment.

For further information on the algorithm, the design choices, and the structure, we provide the full API (Section A) below. In addition, the source code[4] can be found online as soon as the submission process of the corresponding paper is finished.

4 Conclusion

In this deliverable, we looked at the concrete implementation of MPC-accumulators - a cryptographic primitive developed in the context of Safe-DEED. MPC-accumulators show how one can gain performance through distributed trust. Our goal is that the community can easily use MPC-accumulators. Therefore we included a step-by-step explanation of the build process (Section 2). For a complete understanding, we explained the program structure (Section 3) and put a lot of effort into describing the API of the MPC-accumulators. For the next version of this deliverable, we plan to build privacy-friendly real-world applications. These real-world applications will be based on the most recent cryptographic primitives, like MPC-accumulators. In addition, we always try to make our applications accessible to a broad audience. To achieve this, we are currently investigating how to integrate our solutions into big open-source software projects.

[4] `https://github.com/orgs/Safe-DEED/dashboard`

5 References

[1] Alexandra Institute. FRESCO - a FRamework for Efficient Secure COmputation. `https://github.com/aicis/fresco/`, 2019.

[2] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. Practical covertly secure MPC for dishonest majority - or: Breaking the SPDZ limits. In *ESORICS*, volume 8134 of *LNCS*, pages 1–18. Springer, 2013.

[3] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO*, volume 7417 of *LNCS*, pages 643–662. Springer, 2012.

[4] Christian Hanser and Sebastian Ramacher. IAIK ECCelerate, 2019. URL: `https://jce.iaik.tugraz.at/sic/Products/Core_Crypto_Toolkits/ECCelerate`.

A API

MPC-Accumulator

Generated by Doxygen 1.8.13

Contents

1 Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

2 Hierarchical Index

2.1 Class Hierarchy

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

3 Class Index

3.1 Class List

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

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

5 Namespace Documentation

5.1 Package at

Packages

- package iaik

5.2 Package at.iaik

Packages

- package mpc_acc
- package utils

5.3 Package at.iaik.mpc_acc

Classes

- class Accumulator
- class AccumulatorDemo
- class Auxillery
- class EvalResult
- class Main
- class MerkleTree
- class MPC_Acc
- class MPC_Add
- class MPC_Del
- class MPC_Eval
- class MPC_Gen

- class MPC_TripleDummy
- class MPC_WitCreate
- class MPC_WitUpdateAdd
- class MPC_WitUpdateDel
- class MPCParams
- class MPCParamsBuilder
- class Node
- class Polynomial
- class Witness

5.4 Package at.iaik.utils

Classes

- class CmdLineParser
- class MPCBuilder
- class NetworkLoggingDecorator
- class NetworkManager

5.5 Package dk

Packages

- package alexandra

5.6 Package dk.alexandra

Packages

- package fresco

5.7 Package dk.alexandra.fresco

Packages

- package suite

5.8 Package dk.alexandra.fresco.suite

Packages

- package spdz

5.9 Package dk.alexandra.fresco.suite.spdz

Packages

- package ECCExtension

5.10 Package dk.alexandra.fresco.suite.spdz.ECCExtension

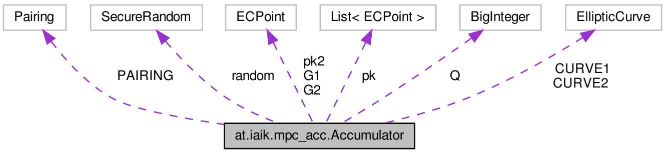
Classes

- interface SECPoint
- class SpdzECCMacCheckProtocol
- class SpdzECCOps
- class SpdzECPoint
- class SpdzKnownMultECCProtocol
- class SpdzKnownScalar
- class SpdzMultECCProtocol
- class SpdzOpenedValueECCStoreImpl
- class SpdzOutputPointProtocol

6 Class Documentation

6.1 at.iaik.mpc_acc.Accumulator Class Reference

Collaboration diagram for at.iaik.mpc_acc.Accumulator:



Public Member Functions

- Accumulator (int size, int t)
- void test (List< BigInteger > X)
- BigInteger getQ ()

Private Member Functions

- BigInteger getRandomScalar ()
- void Gen ()
- EvalResult Eval (List< BigInteger > X, BigInteger r)
- EvalResult Eval (List< BigInteger > X)
- Witness WitCreate (ECPoint acc, Auxillery aux, BigInteger x)
- Boolean Verify (ECPoint acc, Witness wit, BigInteger x)
- EvalResult Add (ECPoint acc, Auxillery aux, BigInteger x)
- EvalResult Delete (ECPoint acc, Auxillery aux, BigInteger x)
- Witness WitUpdate (Witness wit, Auxillery aux, BigInteger x)

Private Attributes

- final Pairing PAIRING
- final EllipticCurve CURVE1
- final EllipticCurve CURVE2
- final BigInteger Q
- final ECPoint G1
- final ECPoint G2
- int size
- int t
- List< ECPoint > pk
- ECPoint pk2
- SecureRandom random

6.1.1 Detailed Description

A simple class computing the non-MPC, keyless accumulator.

Author

Roman Walch

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Accumulator()

```
at.iaik.mpc_acc.Accumulator.Accumulator (
             int size,
             int t )
```

Constructor

Parameters

size	bitlength of the used pairing curves (400)
t	the threshold to precompute the public key

6.1.3 Member Function Documentation

6.1.3.1 Add()

```
EvalResult at.iaik.mpc_acc.Accumulator.Add (
             ECPoint acc,
             Auxillery aux,
             BigInteger x )  [private]
```

Adds an element to the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element

Returns

the new accumulator and AUX data

6.1.3.2 Delete()

```
EvalResult at.iaik.mpc_acc.Accumulator.Delete (
            ECPoint acc,
            Auxillery aux,
            BigInteger x )  [private]
```

Removes an element from the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element

Returns

the new accumulator and AUX data

6.1.3.3 Eval() [1/2]

```
EvalResult at.iaik.mpc_acc.Accumulator.Eval (
            List< BigInteger > X,
            BigInteger r )  [private]
```

Accumulates a set of elements into an accumulator

Parameters

X	the set of elements
r	a random element

Returns

the accumulator and the AUX data

6.1.3.4 Eval() [2/2]

```
EvalResult at.iaik.mpc_acc.Accumulator.Eval (
            List< BigInteger > X )  [private]
```

Accumulates a set of elements into an accumulator

Parameters

X	the set of elements

Returns

the accumulator and the AUX data

6.1.3.5 Gen()

```
void at.iaik.mpc_acc.Accumulator.Gen ( )  [private]
```

Generates the keys of the accumulator

6.1.3.6 getQ()

```
BigInteger at.iaik.mpc_acc.Accumulator.getQ ( )
```

6.1.3.7 getRandomScalar()

```
BigInteger at.iaik.mpc_acc.Accumulator.getRandomScalar ( )  [private]
```

Sample a random integer

Returns

a random integer

6.1.3.8 test()

```
void at.iaik.mpc_acc.Accumulator.test (
            List< BigInteger > X )
```

A simple member to test the accumulator

Parameters

X	the set to be accumulated

6.1.3.9 Verify()

```
Boolean at.iaik.mpc_acc.Accumulator.Verify (
            ECPoint acc,
            Witness wit,
            BigInteger x )  [private]
```

Verifies if an element is part of the accumulator

Parameters

acc	the accumulator
wit	the witness to the element
x	the element

Returns

true if the element is part of the accumulator, false otherwise

6.1.3.10 WitCreate()

```
Witness at.iaik.mpc_acc.Accumulator.WitCreate (
            ECPoint acc,
            Auxillery aux,
            BigInteger x )  [private]
```

Creates a Witness to an element in the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element in the accumulator

Returns

the witness to the element

6.1.3.11 WitUpdate()

```
Witness at.iaik.mpc_acc.Accumulator.WitUpdate (
            Witness wit,
            Auxillery aux,
            BigInteger x )  [private]
```

Updates a witness after element addition/removal

Parameters

wit	the previous witness of the element
aux	the AUX data
x	the element

Returns

the updated witness

6.1.4 Member Data Documentation

6.1.4.1 CURVE1

```
final EllipticCurve at.iaik.mpc_acc.Accumulator.CURVE1  [private]
```

6.1.4.2 CURVE2

```
final EllipticCurve at.iaik.mpc_acc.Accumulator.CURVE2  [private]
```

6.1.4.3 G1

```
final ECPoint at.iaik.mpc_acc.Accumulator.G1  [private]
```

6.1.4.4 G2

```
final ECPoint at.iaik.mpc_acc.Accumulator.G2  [private]
```

6.1.4.5 PAIRING

```
final Pairing at.iaik.mpc_acc.Accumulator.PAIRING  [private]
```

6.1.4.6 pk

```
List<ECPoint> at.iaik.mpc_acc.Accumulator.pk  [private]
```

6.1.4.7 pk2

```
ECPoint at.iaik.mpc_acc.Accumulator.pk2  [private]
```

6.1.4.8 Q

```
final BigInteger at.iaik.mpc_acc.Accumulator.Q  [private]
```

6.1.4.9 random

```
SecureRandom at.iaik.mpc_acc.Accumulator.random  [private]
```

6.1.4.10 size

```
int at.iaik.mpc_acc.Accumulator.size  [private]
```

6.1.4.11 t

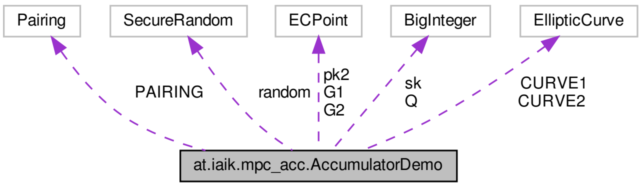
```
int at.iaik.mpc_acc.Accumulator.t  [private]
```

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

- Accumulator.java

6.2 at.iaik.mpc_acc.AccumulatorDemo Class Reference

Collaboration diagram for at.iaik.mpc_acc.AccumulatorDemo:



Public Member Functions

- AccumulatorDemo (int size)
- void test (List< BigInteger > X)

Private Member Functions

- BigInteger getRandomScalar ()
- void Gen ()
- EvalResult Eval (List< BigInteger > X)
- ECPoint WitCreate (ECPoint acc, Auxillery aux, BigInteger x)
- Boolean Verify (ECPoint acc, ECPoint wit, BigInteger x)
- EvalResult Add (ECPoint acc, Auxillery aux, BigInteger x)
- EvalResult Delete (ECPoint acc, Auxillery aux, BigInteger x)
- ECPoint WitUpdate (ECPoint wit, Auxillery aux, BigInteger x)

Private Attributes

- final Pairing PAIRING
- final EllipticCurve CURVE1
- final EllipticCurve CURVE2
- final BigInteger Q
- final ECPoint G1
- final ECPoint G2
- int size
- ECPoint pk2
- BigInteger sk
- SecureRandom random

6.2.1 Detailed Description

A simple class computing the non-MPC accumulator using the secret key.

Author

Roman Walch

6.2.2 Constructor & Destructor Documentation

6.2.2.1 AccumulatorDemo()

```
at.iaik.mpc_acc.AccumulatorDemo.AccumulatorDemo (
            int size )
```

Constructor

Parameters

size	bitlength of the used pairing curves (400)

6.2.3 Member Function Documentation

6.2.3.1 Add()

```
EvalResult at.iaik.mpc_acc.AccumulatorDemo.Add (
            ECPoint acc,
            Auxillery aux,
            BigInteger x )  [private]
```

Adds an element to the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element

Returns

the new accumulator and AUX data

6.2.3.2 Delete()

```
EvalResult at.iaik.mpc_acc.AccumulatorDemo.Delete (
            ECPoint acc,
            Auxillery aux,
            BigInteger x )  [private]
```

Removes an element from the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element

Returns

the new accumulator and AUX data

6.2.3.3 Eval()

```
EvalResult at.iaik.mpc_acc.AccumulatorDemo.Eval (
             List< BigInteger > X )  [private]
```

Accumulates a set of elements into an accumulator

Parameters

X	the set of elements

Returns

the accumulator and the AUX data

6.2.3.4 Gen()

```
void at.iaik.mpc_acc.AccumulatorDemo.Gen ( )  [private]
```

Generates the keys of the accumulator

6.2.3.5 getRandomScalar()

```
BigInteger at.iaik.mpc_acc.AccumulatorDemo.getRandomScalar ( )  [private]
```

Sample a random integer

Returns

a random integer

6.2.3.6 test()

```
void at.iaik.mpc_acc.AccumulatorDemo.test (
             List< BigInteger > X )
```

A simple member to test the accumulator

Parameters

X	the set to be accumulated

6.2.3.7 Verify()

```
Boolean at.iaik.mpc_acc.AccumulatorDemo.Verify (
             ECPoint acc,
             ECPoint wit,
             BigInteger x )  [private]
```

Verifies if an element is part of the accumulator

Parameters

acc	the accumulator
wit	the witness to the element
x	the element

Returns

true if the element is part of the accumulator, false otherwise

6.2.3.8 WitCreate()

```
ECPoint at.iaik.mpc_acc.AccumulatorDemo.WitCreate (
            ECPoint acc,
            Auxillery aux,
            BigInteger x )  [private]
```

Creates a Witness to an element in the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element in the accumulator

Returns

the witness to the element

6.2.3.9 WitUpdate()

```
ECPoint at.iaik.mpc_acc.AccumulatorDemo.WitUpdate (
            ECPoint wit,
            Auxillery aux,
            BigInteger x )  [private]
```

Updates a witness after element addition/removal

Parameters

wit	the previous witness of the element
aux	the AUX data
x	the element

Returns

the updated witness

6.2.4 Member Data Documentation

6.2.4.1 CURVE1

```
final EllipticCurve at.iaik.mpc_acc.AccumulatorDemo.CURVE1  [private]
```

6.2.4.2 CURVE2

```
final EllipticCurve at.iaik.mpc_acc.AccumulatorDemo.CURVE2  [private]
```

6.2.4.3 G1

```
final ECPoint at.iaik.mpc_acc.AccumulatorDemo.G1  [private]
```

6.2.4.4 G2

```
final ECPoint at.iaik.mpc_acc.AccumulatorDemo.G2  [private]
```

6.2.4.5 PAIRING

```
final Pairing at.iaik.mpc_acc.AccumulatorDemo.PAIRING  [private]
```

6.2.4.6 pk2

```
ECPoint at.iaik.mpc_acc.AccumulatorDemo.pk2  [private]
```

6.2.4.7 Q

```
final BigInteger at.iaik.mpc_acc.AccumulatorDemo.Q  [private]
```

6.2.4.8 random

```
SecureRandom at.iaik.mpc_acc.AccumulatorDemo.random  [private]
```

6.2.4.9 size

```
int at.iaik.mpc_acc.AccumulatorDemo.size  [private]
```

6.2.4.10 sk

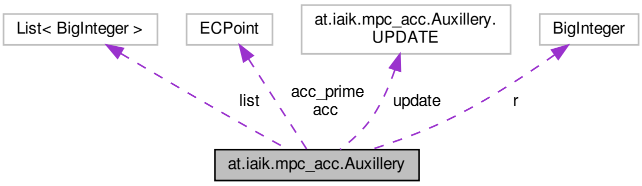
```
BigInteger at.iaik.mpc_acc.AccumulatorDemo.sk  [private]
```

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

- AccumulatorDemo.java

6.3 at.iaik.mpc_acc.Auxillery Class Reference

Collaboration diagram for at.iaik.mpc_acc.Auxillery:



Classes

- enum **UPDATE**

Public Member Functions

- Auxillery (List< BigInteger > list, UPDATE update)
- Auxillery (List< BigInteger > list, UPDATE update, ECPoint acc, ECPoint acc_prime)
- Auxillery (List< BigInteger > list, BigInteger r)
- List< BigInteger > getList ()
- UPDATE getUpdate ()
- void clearUpdate ()
- BigInteger getR ()
- ECPoint getAcc ()
- ECPoint getAccPrime ()

Private Attributes

- List< BigInteger > list
- BigInteger r
- UPDATE update
- ECPoint acc
- ECPoint acc_prime

6.3.1 Detailed Description

A simple class containing the AUX data

Author

Roman Walch

6.3.2 Constructor & Destructor Documentation

6.3.2.1 Auxillery() [1/3]

```
at.iaik.mpc_acc.Auxillery.Auxillery (
             List< BigInteger > list,
             UPDATE update )
```

Initialize the class

Parameters

list	the list of elements
update	indicator if a add/delete operation was conducted

6.3.2.2 Auxillery() [2/3]

```
at.iaik.mpc_acc.Auxillery.Auxillery (
             List< BigInteger > list,
             UPDATE update,
             ECPoint acc,
             ECPoint acc_prime )
```

Initialize the class

Parameters

list	the list of elements
update	indicator if a add/delete operation was conducted
acc	the previous accumulator
acc_prime	the accumulator

6.3.2.3 Auxillery() [3/3]

```
at.iaik.mpc_acc.Auxillery.Auxillery (
             List< BigInteger > list,
             BigInteger r )
```

Initialize the class

Parameters

list	the list of elements
r	the random element

6.3.3 Member Function Documentation

6.3.3.1 clearUpdate()

```
void at.iaik.mpc_acc.Auxillery.clearUpdate ( )
```

Clears the update indicator

6.3.3.2 getAcc()

```
ECPoint at.iaik.mpc_acc.Auxillery.getAcc ( )
```

Getter for the previous accumulator

6.3.3.3 getAccPrime()

```
ECPoint at.iaik.mpc_acc.Auxillery.getAccPrime ( )
```

Getter for the accumulator

6.3.3.4 getList()

```
List<BigInteger> at.iaik.mpc_acc.Auxillery.getList ( )
```

Getter for the list of elements

6.3.3.5 getR()

```
BigInteger at.iaik.mpc_acc.Auxillery.getR ( )
```

Getter for the random element

6.3.3.6 getUpdate()

```
UPDATE at.iaik.mpc_acc.Auxillery.getUpdate ( )
```

Getter for the update indicator

6.3.4 Member Data Documentation

6.3.4.1 acc

```
ECPoint at.iaik.mpc_acc.Auxillery.acc  [private]
```

6.3.4.2 acc_prime

```
ECPoint at.iaik.mpc_acc.Auxillery.acc_prime  [private]
```

6.3.4.3 list

```
List<BigInteger> at.iaik.mpc_acc.Auxillery.list  [private]
```

6.3.4.4 r

```
BigInteger at.iaik.mpc_acc.Auxillery.r  [private]
```

6.3.4.5 update

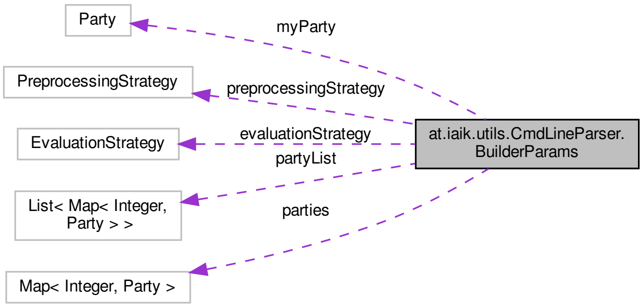
```
UPDATE at.iaik.mpc_acc.Auxillery.update  [private]
```

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

- Auxillery.java

6.4 at.iaik.utils.CmdLineParser.BuilderParams Class Reference

Collaboration diagram for at.iaik.utils.CmdLineParser.BuilderParams:



Public Member Functions

- BuilderParams (boolean logging, boolean multiThreaded)
- void setId (int id)
- void setEl (int el)
- void setParties (List< Map< Integer, Party >> partyList, Party party)
- void setMaxBitLength (int maxBitLength)
- void setPreprocessingStrategy (PreprocessingStrategy strategy)
- void setEvaluationStrategy (EvaluationStrategy strategy)

Public Attributes

- boolean logging
- int id
- Party myParty
- Map< Integer, Party > parties
- List< Map< Integer, Party > > partyList
- int maxBitLength
- PreprocessingStrategy preprocessingStrategy
- EvaluationStrategy evaluationStrategy
- boolean multiThreaded
- int el

6.4.1 Detailed Description

A class providing storage for all the parameters the applications need to be properly initialized. This includes parameters used in the framework as well as parameters which are use case specific.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 BuilderParams()

```
at.iaik.utils.CmdLineParser.BuilderParams.BuilderParams (
             boolean logging,
             boolean multiThreaded )
```

6.4.3 Member Function Documentation

6.4.3.1 setEl()

```
void at.iaik.utils.CmdLineParser.BuilderParams.setEl (
             int el )
```

6.4.3.2 setEvaluationStrategy()

```
void at.iaik.utils.CmdLineParser.BuilderParams.setEvaluationStrategy (
             EvaluationStrategy strategy )
```

6.4.3.3 setId()

```
void at.iaik.utils.CmdLineParser.BuilderParams.setId (
             int id )
```

6.4.3.4 setMaxBitLength()

```
void at.iaik.utils.CmdLineParser.BuilderParams.setMaxBitLength (
             int maxBitLength )
```

6.4.3.5 setParties()

```
void at.iaik.utils.CmdLineParser.BuilderParams.setParties (
             List< Map< Integer, Party >> partyList,
             Party party )
```

6.4.3.6 setPreprocessingStrategy()

```
void at.iaik.utils.CmdLineParser.BuilderParams.setPreprocessingStrategy (
             PreprocessingStrategy strategy )
```

6.4.4 Member Data Documentation

6.4.4.1 el

```
int at.iaik.utils.CmdLineParser.BuilderParams.el
```

6.4.4.2 evaluationStrategy

```
EvaluationStrategy at.iaik.utils.CmdLineParser.BuilderParams.evaluationStrategy
```

6.4.4.3 id

```
int at.iaik.utils.CmdLineParser.BuilderParams.id
```

6.4.4.4 logging

```
boolean at.iaik.utils.CmdLineParser.BuilderParams.logging
```

6.4.4.5 maxBitLength

```
int at.iaik.utils.CmdLineParser.BuilderParams.maxBitLength
```

6.4.4.6 multiThreaded

```
boolean at.iaik.utils.CmdLineParser.BuilderParams.multiThreaded
```

6.4.4.7 myParty

```
Party at.iaik.utils.CmdLineParser.BuilderParams.myParty
```

6.4.4.8 parties

```
Map<Integer, Party> at.iaik.utils.CmdLineParser.BuilderParams.parties
```

6.4.4.9 partyList

```
List<Map<Integer, Party> > at.iaik.utils.CmdLineParser.BuilderParams.partyList
```

6.4.4.10 preprocessingStrategy

```
PreprocessingStrategy at.iaik.utils.CmdLineParser.BuilderParams.preprocessingStrategy
```

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

- CmdLineParser.java

6.5 at.iaik.utils.CmdLineParser Class Reference

Classes

- class BuilderParams

Static Public Member Functions

- static Party checkParty (String partyOption) throws ParseException
- static BuilderParams GetCmdLineParams (String[ ] args) throws ParseException

Static Public Attributes

- static final String IDMSG = "The id of this player. Must be a unique positive integer."
- static final String SETMSG = "The number of elements in the generator."
- static final String PARTYMSG = "Connection data for a party. Use -p multiple times to specify many players. You must always at least include yourself. Must be on the form [id]:[hostname]:[port]. id is a unique positive integer for the player, host and port is where to find the player"
- static final String PRESTRATMSG = "Used to set the preprocessing Strategy of SPDZ"
- static final String LOGGINGMSG = "Informs FRESCO that performance logging should be triggered"
- static final String IDERRMSG = "ID must be positive"
- static final String PARTYERRMSG = "Party ids must be unique"
- static final String SETERRMSG = "number of elements must be > 1"
- static int newID = 0

Static Private Member Functions

- static List< Map< Integer, Party > > createPartyMap (Map< Integer, Party > parties, int myID)

6.5.1 Detailed Description

Utility class to gather all the builder parameters necessary for the applications from the command line.

Author

Fabian Schmid

6.5.2 Member Function Documentation

6.5.2.1 checkParty()

```
static Party at.iaik.utils.CmdLineParser.checkParty (
            String partyOption ) throws ParseException  [static]
```

Checks on a basic level if the party input after -p is in the correct form

Parameters

partyOption	The option provided after this -p argument

Returns

Returns the party parsed from the partyOption

Exceptions

ParseException	If the partyOption is not compliant with the expected format

6.5.2.2 createPartyMap()

```
static List<Map<Integer, Party> > at.iaik.utils.CmdLineParser.createPartyMap (
            Map< Integer, Party > parties,
            int myID )  [static], [private]
```

In case the application is executed using multi threading, this function separates the parties into different party maps. in these different maps, the ids are newly set, since in the framework the ids have to contain 1 and have to ascend from one on. If this is to be changed, one has to replace the socket network of the framework with an independent implementation.

Parameters

parties	the current party map which is to be separated
myID	the id this user currently possesses

Returns

a list of new party maps, one for each subgroup of clients

NOTE: This function has to be exchanged/modified if one wants to change the logic of separation (e.g. separation by date) Now separation is by id: (1,2,3), (1,4,5) ... 1 is infineon and has to be present in every map

6.5.2.3 GetCmdLineParams()

```
static BuilderParams at.iaik.utils.CmdLineParser.GetCmdLineParams (
             String [] args ) throws ParseException  [static]
```

Main functionality to read the input arguments and parse the Builder Parameters. Every option is required but the multithreaded and logging flag. the correct format of each option can be read below or int the examples given in the makefile

Parameters

args	the command line arguments provided when starting up the program

Returns

Returns a BuilderParams object, providing everything the builders need.

Exceptions

ParseException	throws this exception if some value is not as expected

6.5.3 Member Data Documentation

6.5.3.1 IDERRMSG

```
final String at.iaik.utils.CmdLineParser.IDERRMSG = "ID must be positive"  [static]
```

6.5.3.2 IDMSG

```
final String at.iaik.utils.CmdLineParser.IDMSG = "The id of this player.  Must be a unique
positive integer."  [static]
```

6.5.3.3 LOGGINGMSG

```
final String at.iaik.utils.CmdLineParser.LOGGINGMSG = "Informs FRESCO that performance logging
should be triggered"  [static]
```

6.5.3.4 newID

```
int at.iaik.utils.CmdLineParser.newID = 0  [static]
```

6.5.3.5 PARTYERRMSG

```
final String at.iaik.utils.CmdLineParser.PARTYERRMSG = "Party ids must be unique"  [static]
```

6.5.3.6 PARTYMSG

```
final String at.iaik.utils.CmdLineParser.PARTYMSG = "Connection data for a party.  Use -p
multiple times to specify many players.  You must always at least include yourself.  Must be
on the form [id]:[hostname]:[port].  id is a unique positive integer for the player, host and
port is where to find the player"  [static]
```

6.5.3.7 PRESTRATMSG

```
final String at.iaik.utils.CmdLineParser.PRESTRATMSG = "Used to set the preprocessing Strategy
of SPDZ"  [static]
```

6.5.3.8 SETERRMSG

```
final String at.iaik.utils.CmdLineParser.SETERRMSG = "number of elements must be > 1"  [static]
```

6.5.3.9 SETMSG

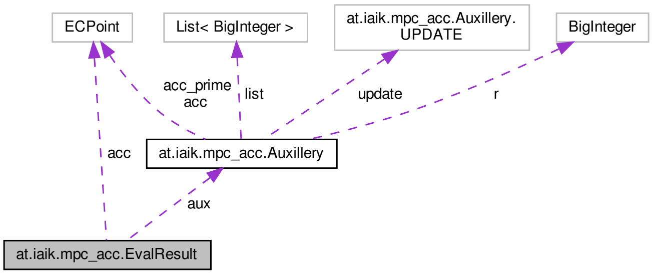
```
final String at.iaik.utils.CmdLineParser.SETMSG = "The number of elements in the generator."
[static]
```

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

- CmdLineParser.java

6.6 at.iaik.mpc_acc.EvalResult Class Reference

Collaboration diagram for at.iaik.mpc_acc.EvalResult:


ECPoint
List< BigInteger >
at.iaik.mpc_acc.Auxillery.
UPDATE
BigInteger
acc_prime
acc
list
update
r
acc
at.iaik.mpc_acc.Auxillery
aux
at.iaik.mpc_acc.EvalResult

Public Member Functions

- EvalResult (ECPoint acc, List< BigInteger > list)
- EvalResult (ECPoint acc, List< BigInteger > list, Auxillery.UPDATE update)
- EvalResult (ECPoint acc, List< BigInteger > list, Auxillery.UPDATE update, ECPoint acc_old)
- EvalResult (ECPoint acc, List< BigInteger > list, BigInteger r)
- ECPoint getAcc ()
- Auxillery getAuxillery ()

Private Attributes

- ECPoint acc
- Auxillery aux

6.6.1 Detailed Description

A simple class containing the evaluation result

Author

Roman Walch

6.6.2 Constructor & Destructor Documentation

6.6.2.1 EvalResult() [1/4]

```
at.iaik.mpc_acc.EvalResult.EvalResult (
            ECPoint acc,
            List< BigInteger > list )
```

Initialize the class

Parameters

acc	the accumulator
list	the list of elements in the accumulator

6.6.2.2 EvalResult() [2/4]

```
at.iaik.mpc_acc.EvalResult.EvalResult (
            ECPoint acc,
            List< BigInteger > list,
            Auxillery.UPDATE update )
```

Initialize the class

Parameters

acc	the accumulator
list	the list of elements in the accumulator
update	specifies if a add/delete operation was conducted

6.6.2.3 EvalResult() [3/4]

```
at.iaik.mpc_acc.EvalResult.EvalResult (
            ECPoint acc,
            List< BigInteger > list,
            Auxillery.UPDATE update,
            ECPoint acc_old )
```

Initialize the class

Parameters

acc	the accumulator
list	the list of elements in the accumulator
update	specifies if a add/delete operation was conducted
acc_old	the previous accumulator

6.6.2.4 EvalResult() [4/4]

```
at.iaik.mpc_acc.EvalResult.EvalResult (
            ECPoint acc,
            List< BigInteger > list,
            BigInteger r )
```

Initialize the class

Parameters

acc	the accumulator
list	the list of elements in the accumulator
r	the random element

6.6.3 Member Function Documentation

6.6.3.1 getAcc()

```
ECPoint at.iaik.mpc_acc.EvalResult.getAcc ( )
```

Getter for the accumulator

6.6.3.2 getAuxillery()

```
Auxillery at.iaik.mpc_acc.EvalResult.getAuxillery ( )
```

Getter for the AUX data

6.6.4 Member Data Documentation

6.6.4.1 acc

```
ECPoint at.iaik.mpc_acc.EvalResult.acc [private]
```

6.6.4.2 aux

```
Auxillery at.iaik.mpc_acc.EvalResult.aux [private]
```

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

- EvalResult.java

6.7 at.iaik.mpc_acc.Main Class Reference

Static Public Member Functions

- static void main (String[ ] args) throws ParseException

Static Private Member Functions

- static List< BigInteger > genList (BigInteger mod, int elements)

Static Private Attributes

- static final int SIZE = 400

6.7.1 Detailed Description

A simple demo computing the MPC accumulator.

Author

Roman Walch

6.7.2 Member Function Documentation

6.7.2.1 genList()

```
static List<BigInteger> at.iaik.mpc_acc.Main.genList (
             BigInteger mod,
             int elements )  [static], [private]
```

A simple member to generate a set to be accumulated

Parameters

mod	the used modulus
elements	the number of elements

Returns

the list of elements

6.7.2.2 main()

```
static void at.iaik.mpc_acc.Main.main (
             String [] args ) throws ParseException  [static]
```

The main class to test the MPC accumulator implementation

Exceptions

ParseException	

6.7.3 Member Data Documentation

6.7.3.1 SIZE

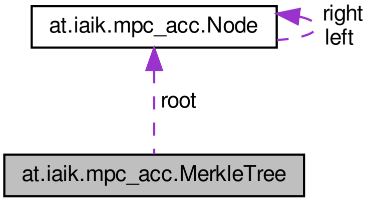
```
final int at.iaik.mpc_acc.Main.SIZE = 400  [static], [private]
```

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

- Main.java

6.8 at.iaik.mpc_acc.MerkleTree Class Reference

Collaboration diagram for at.iaik.mpc_acc.MerkleTree:



Classes

- enum Position
- class ProofNode

Public Member Functions

- MerkleTree (List< byte[ ]> hashes)
- MerkleTree (byte[ ] root_hash)
- List< ProofNode > proof (byte[ ] digest)
- Boolean verify (byte[ ] element, List< ProofNode > proof)
- byte [ ] getRootHash ()

Static Public Member Functions

- static void test (List< BigInteger > X)

Private Member Functions

- Boolean find (Node node, Stack< Pair< Position, Node >> nodes, byte[ ] digest)

Static Private Member Functions

- static int round_up_to_power_of_2 (int value)

Private Attributes

- Node root

  the root node of the merkle tree

6.8.1 Detailed Description

A simple class for Merkle Tree accumulators

Author

Roman Walch

6.8.2 Constructor & Destructor Documentation

6.8.2.1 MerkleTree() [1/2]

```
at.iaik.mpc_acc.MerkleTree.MerkleTree (
             List< byte[]> hashes )
```

Initialize Merkle tree from a set of digests.

Given the vector of digest, builds a Merkle tree where the digests are placed in the the leaf nodes.

Parameters

hashes	set of digests, which will be hashed into the leaf nodes

6.8.2.2 MerkleTree() [2/2]

```
at.iaik.mpc_acc.MerkleTree.MerkleTree (
             byte [] root_hash )
```

Initialize Merkle tree from a root hash.

In this configuration, the Merkle tree can only be used for verification.

Parameters

root_hash	the root hash

6.8.3 Member Function Documentation

6.8.3.1 find()

```
Boolean at.iaik.mpc_acc.MerkleTree.find (
             Node node,
             Stack< Pair< Position, Node >> nodes,
             byte [] digest )  [private]
```

6.8.3.2 getRootHash()

```
byte [] at.iaik.mpc_acc.MerkleTree.getRootHash ( )
```

Return root hash of the Merkle tree

Returns

root hash

6.8.3.3 proof()

```
List<ProofNode> at.iaik.mpc_acc.MerkleTree.proof (
             byte [] digest )
```

Create a member ship proof for the given digest.

A proof consists of a sequence of proof_node instances, where each proof_node declares if the proven value is the left or right input to the hash function and contains the digest of the sibling.

Parameters

digest	digest to proof

Returns

sequence of proof nodes or an empty sequence of the value is not contained in the tree

6.8.3.4 round_up_to_power_of_2()

```
static int at.iaik.mpc_acc.MerkleTree.round_up_to_power_of_2 (
             int value )  [static], [private]
```

6.8.3.5 test()

```
static void at.iaik.mpc_acc.MerkleTree.test (
             List< BigInteger > X )  [static]
```

A simple member to test the merkle tree

Parameters

X	the set to be accumulated

6.8.3.6 verify()

```
Boolean at.iaik.mpc_acc.MerkleTree.verify (
             byte [] element,
             List< ProofNode > proof )
```

Verify the membership of a given value and its proof against the root hash.

Parameters

element	value to be tested
proof	proof for the given value

Returns

true if the value is contained in the tree, i.e. the proof matches the root hash

6.8.4 Member Data Documentation

6.8.4.1 root

```
Node at.iaik.mpc_acc.MerkleTree.root  [private]
```

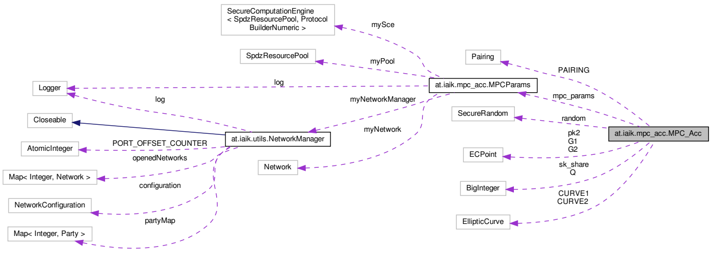
the root node of the merkle tree

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

- MerkleTree.java

6.9 at.iaik.mpc_acc.MPC_Acc Class Reference

Collaboration diagram for at.iaik.mpc_acc.MPC_Acc:



Public Member Functions

- BigInteger getRandomScalar ()
- MPC_Acc (int size, CmdLineParser.BuilderParams params) throws ParseException
- void gen ()
- EvalResult eval (List< BigInteger > X)
- Witness witCreate (ECPoint acc, Auxillery aux, BigInteger x)
- Boolean verify (ECPoint acc, Witness wit, BigInteger x)
- EvalResult add (ECPoint acc, Auxillery aux, BigInteger x)
- EvalResult delete (ECPoint acc, Auxillery aux, BigInteger x)
- Witness witUpdate (Witness wit, Auxillery aux, BigInteger x)
- void close ()
- void printAndResetComm (String info)
- void prepareBatches (int mul, int random)
- BigInteger getQ ()

Private Attributes

- final Pairing PAIRING
- final EllipticCurve CURVE1
- final EllipticCurve CURVE2
- final BigInteger Q
- final ECPoint G1
- final ECPoint G2
- int size
- ECPoint pk2
- MPCParams mpc_params
- BigInteger sk_share
- SecureRandom random

6.9.1 Detailed Description

A simple demo for the MPC accumulator computations

Author

Roman Walch

6.9.2 Constructor & Destructor Documentation

6.9.2.1 MPC_Acc()

```
at.iaik.mpc_acc.MPC_Acc.MPC_Acc (
             int size,
             CmdLineParser.BuilderParams params ) throws ParseException
```

Constructor

Parameters

size	bitlength of the used pairing curves (400)
params	command line params to specify the computation parameters

6.9.3 Member Function Documentation

6.9.3.1 add()

```
EvalResult at.iaik.mpc_acc.MPC_Acc.add (
             ECPoint acc,
             Auxillery aux,
             BigInteger x )
```

Adds an element to the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element

Returns

the new accumulator and AUX data

6.9.3.2 close()

```
void at.iaik.mpc_acc.MPC_Acc.close ( )
```

Closes the network and secure computation engine

6.9.3.3 delete()

```
EvalResult at.iaik.mpc_acc.MPC_Acc.delete (
            ECPoint acc,
            Auxillery aux,
            BigInteger x )
```

Removes an element from the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element

Returns

the new accumulator and AUX data

6.9.3.4 eval()

```
EvalResult at.iaik.mpc_acc.MPC_Acc.eval (
            List< BigInteger > X )
```

Accumulates a set of elements into an accumulator

Parameters

X	the set of elements

Returns

the accumulator and the AUX data

6.9.3.5 gen()

```
void at.iaik.mpc_acc.MPC_Acc.gen ( )
```

Generates the keys of the accumulator

6.9.3.6 getQ()

```
BigInteger at.iaik.mpc_acc.MPC_Acc.getQ ( )
```

Getter for the modulus

6.9.3.7 getRandomScalar()

```
BigInteger at.iaik.mpc_acc.MPC_Acc.getRandomScalar ( )
```

Sample a random integer

Returns

a random integer

6.9.3.8 prepareBatches()

```
void at.iaik.mpc_acc.MPC_Acc.prepareBatches (
            int mul,
            int random )
```

Precomputes shared random elements and beaver triples during the offline phase

Parameters

mul	the number of beaver triples to be produced
random	the number of shared random elements to be produced

6.9.3.9 printAndResetComm()

```
void at.iaik.mpc_acc.MPC_Acc.printAndResetComm (
            String info )
```

Prints the currently communicated bytes and reset the counter

Parameters

info	Output prefix

6.9.3.10 verify()

```
Boolean at.iaik.mpc_acc.MPC_Acc.verify (
            ECPoint acc,
            Witness wit,
            BigInteger x )
```

Verifies if an element is part of the accumulator

Parameters

acc	the accumulator
wit	the witness to the element
x	the element

Returns

true if the element is part of the accumulator, false otherwise

6.9.3.11 witCreate()

```
Witness at.iaik.mpc_acc.MPC_Acc.witCreate (
            ECPoint acc,
            Auxillery aux,
            BigInteger x )
```

Creates a Witness to an element in the accumulator

Parameters

acc	the accumulator
aux	the AUX data
x	the element in the accumulator

Returns

the witness to the element

6.9.3.12 witUpdate()

```
Witness at.iaik.mpc_acc.MPC_Acc.witUpdate (
            Witness wit,
            Auxillery aux,
            BigInteger x )
```

Updates a witness after element addition/removal

Parameters

wit	the previous witness of the element
aux	the AUX data
x	the element

Returns

the updated witness

6.9.4 Member Data Documentation

6.9.4.1 CURVE1

```
final EllipticCurve at.iaik.mpc_acc.MPC_Acc.CURVE1  [private]
```

6.9.4.2 CURVE2

```
final EllipticCurve at.iaik.mpc_acc.MPC_Acc.CURVE2  [private]
```

6.9.4.3 G1

```
final ECPoint at.iaik.mpc_acc.MPC_Acc.G1  [private]
```

6.9.4.4 G2

```
final ECPoint at.iaik.mpc_acc.MPC_Acc.G2  [private]
```

6.9.4.5 mpc_params

```
MPCParams at.iaik.mpc_acc.MPC_Acc.mpc_params  [private]
```

6.9.4.6 PAIRING

```
final Pairing at.iaik.mpc_acc.MPC_Acc.PAIRING  [private]
```

6.9.4.7 pk2

```
ECPoint at.iaik.mpc_acc.MPC_Acc.pk2  [private]
```

6.9.4.8 Q

```
final BigInteger at.iaik.mpc_acc.MPC_Acc.Q  [private]
```

6.9.4.9 random

```
SecureRandom at.iaik.mpc_acc.MPC_Acc.random  [private]
```

6.9.4.10 size

```
int at.iaik.mpc_acc.MPC_Acc.size [private]
```

6.9.4.11 sk_share

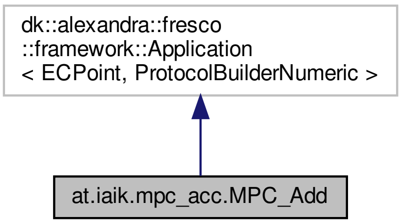
```
BigInteger at.iaik.mpc_acc.MPC_Acc.sk_share [private]
```

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

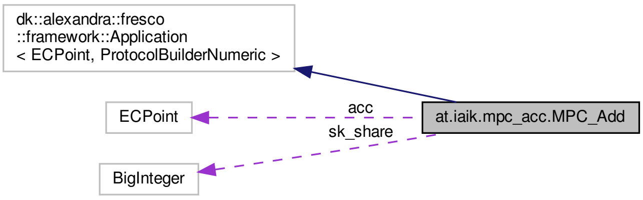
- MPC_Acc.java

6.10 at.iaik.mpc_acc.MPC_Add Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPC_Add:



Collaboration diagram for at.iaik.mpc_acc.MPC_Add:



Public Member Functions

- MPC_Add (BigInteger sk_share, ECPoint acc)
- DRes< ECPoint > buildComputation (ProtocolBuilderNumeric producer)

Private Attributes

- BigInteger sk_share
- ECPoint acc

6.10.1 Detailed Description

Implements the functionality to add an element to the accumulator

Author

Roman Walch

6.10.2 Constructor & Destructor Documentation

6.10.2.1 MPC_Add()

```
at.iaik.mpc_acc.MPC_Add.MPC_Add (
             BigInteger sk_share,
             ECPoint acc )
```

Construct a add computation object for element addition

Parameters

sk_share	the shared secret key
x	the element to be added
acc	the accumulator

6.10.3 Member Function Documentation

6.10.3.1 buildComputation()

```
DRes<ECPoint> at.iaik.mpc_acc.MPC_Add.buildComputation (
             ProtocolBuilderNumeric producer )
```

Performs the MPC computation to add an element to the accumulator

Returns

the updated accumulator

6.10.4 Member Data Documentation

6.10.4.1 acc

```
ECPoint at.iaik.mpc_acc.MPC_Add.acc  [private]
```

6.10.4.2 sk_share

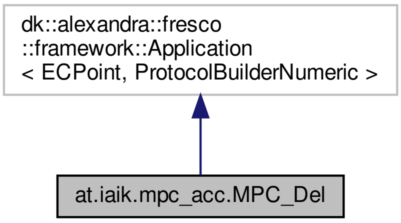
```
BigInteger at.iaik.mpc_acc.MPC_Add.sk_share  [private]
```

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

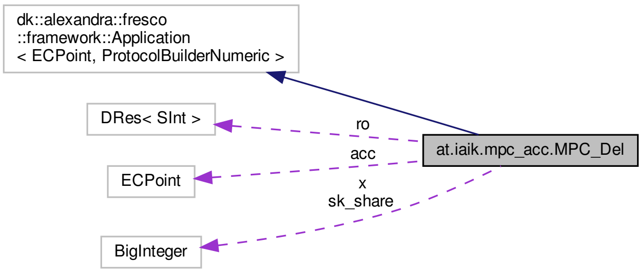
- MPC_Add.java

6.11 at.iaik.mpc_acc.MPC_Del Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPC_Del:



Collaboration diagram for at.iaik.mpc_acc.MPC_Del:



Public Member Functions

- MPC_Del (BigInteger sk_share, BigInteger x, ECPoint acc)
- DRes< ECPoint > buildComputation (ProtocolBuilderNumeric producer)

Private Attributes

- BigInteger sk_share
- BigInteger x
- ECPoint acc
- DRes< SInt > ro

6.11.1 Detailed Description

Implements the functionality to delete an element from the accumulator

Author

Roman Walch

6.11.2 Constructor & Destructor Documentation

6.11.2.1 MPC_Del()

```
at.iaik.mpc_acc.MPC_Del.MPC_Del (
              BigInteger sk_share,
              BigInteger x,
              ECPoint acc )
```

Construct a delete computation object for element removal

Parameters

sk_share	the shared secret key
x	the element to be removed
acc	the accumulator

6.11.3 Member Function Documentation

6.11.3.1 buildComputation()

```
DRes<ECPoint> at.iaik.mpc_acc.MPC_Del.buildComputation (
              ProtocolBuilderNumeric producer )
```

Performs the MPC computation to remove an element from the accumulator

Returns

the updated accumulator

6.11.4 Member Data Documentation

6.11.4.1 acc

```
ECPoint at.iaik.mpc_acc.MPC_Del.acc  [private]
```

6.11.4.2 ro

```
DRes<SInt> at.iaik.mpc_acc.MPC_Del.ro  [private]
```

6.11.4.3 sk_share

```
BigInteger at.iaik.mpc_acc.MPC_Del.sk_share  [private]
```

6.11.4.4 x

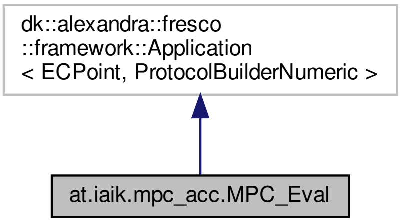
```
BigInteger at.iaik.mpc_acc.MPC_Del.x  [private]
```

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

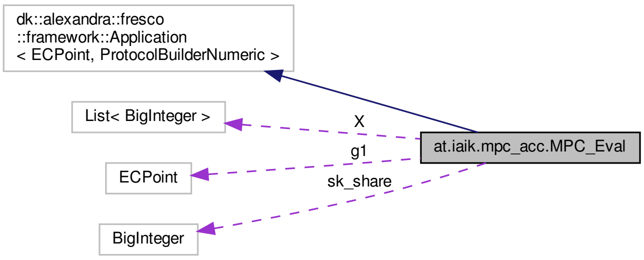
- MPC_Del.java

6.12 at.iaik.mpc_acc.MPC_Eval Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPC_Eval:


dk::alexandra::fresco ::framework::Application < ECPoint, ProtocolBuilderNumeric >
at.iaik.mpc_acc.MPC_Eval

Collaboration diagram for at.iaik.mpc_acc.MPC_Eval:


dk::alexandra::fresco ::framework::Application < ECPoint, ProtocolBuilderNumeric >
List< BigInteger >
ECPoint
BigInteger
X
g1
sk_share
at.iaik.mpc_acc.MPC_Eval

Public Member Functions

- MPC_Eval (BigInteger sk_share, ECPoint g1, List< BigInteger > X)
- DRes< ECPoint > buildComputation (ProtocolBuilderNumeric producer)

Private Attributes

- BigInteger sk_share
- ECPoint g1
- List< BigInteger > X

6.12.1 Detailed Description

Implements the functionality to create the accumulator from a set of values.

Author

Roman Walch

6.12.2 Constructor & Destructor Documentation

6.12.2.1 MPC_Eval()

```
at.iaik.mpc_acc.MPC_Eval.MPC_Eval (
            BigInteger sk_share,
            ECPoint g1,
            List< BigInteger > X )
```

Construct a eval computation object for creating the accumulator

Parameters

sk_share	the shared secret key
g1	the generator of the first pairing group
X	the list of elements to be accumulated

6.12.3 Member Function Documentation

6.12.3.1 buildComputation()

```
DRes<ECPoint> at.iaik.mpc_acc.MPC_Eval.buildComputation (
            ProtocolBuilderNumeric producer )
```

Performs the MPC computation to create the accumulator from the set of elements

Returns

the accumulator

6.12.4 Member Data Documentation

6.12.4.1 g1

```
ECPoint at.iaik.mpc_acc.MPC_Eval.g1  [private]
```

6.12.4.2 sk_share

```
BigInteger at.iaik.mpc_acc.MPC_Eval.sk_share  [private]
```

6.12.4.3 X

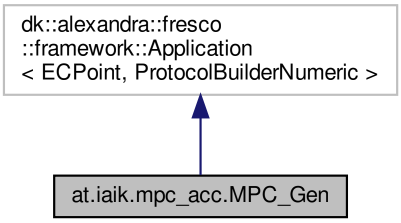
```
List<BigInteger> at.iaik.mpc_acc.MPC_Eval.X  [private]
```

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

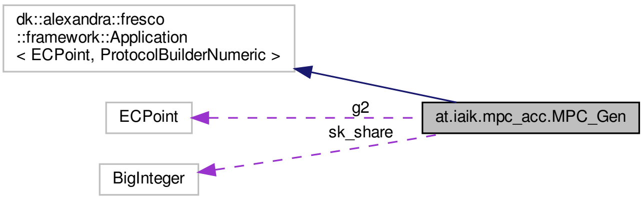
- MPC_Eval.java

6.13 at.iaik.mpc_acc.MPC_Gen Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPC_Gen:



Collaboration diagram for at.iaik.mpc_acc.MPC_Gen:



Public Member Functions

- MPC_Gen (BigInteger sk_share, ECPoint g2)
- DRes< ECPoint > buildComputation (ProtocolBuilderNumeric producer)

Private Attributes

- BigInteger sk_share
- ECPoint g2

6.13.1 Detailed Description

Implements the functionality to generate the pulbic key from the secret key shares

Author

Roman Walch

6.13.2 Constructor & Destructor Documentation

6.13.2.1 MPC_Gen()

```
at.iaik.mpc_acc.MPC_Gen.MPC_Gen (
             BigInteger sk_share,
             ECPoint g2 )
```

Construct a public key generation computation object

Parameters

sk_share	the shared secret key
g2	the generator of the second pairing group

6.13.3 Member Function Documentation

6.13.3.1 buildComputation()

```
DRes<ECPoint> at.iaik.mpc_acc.MPC_Gen.buildComputation (
             ProtocolBuilderNumeric producer )
```

Perform the MPC computation to create the public key from the shared secret key

Returns

the public key

6.13.4 Member Data Documentation

6.13.4.1 g2

```
ECPoint at.iaik.mpc_acc.MPC_Gen.g2  [private]
```

6.13.4.2 sk_share

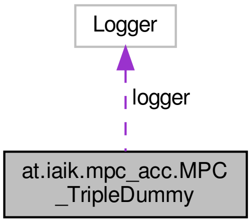
```
BigInteger at.iaik.mpc_acc.MPC_Gen.sk_share  [private]
```

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

- MPC_Gen.java

6.14 at.iaik.mpc_acc.MPC_TripleDummy Class Reference

Collaboration diagram for at.iaik.mpc_acc.MPC_TripleDummy:



Static Public Member Functions

- static void produceTriples (SpdzMascotDataSupplier supplier, int elements)
- static void produceRandomFieldElements (SpdzMascotDataSupplier supplier, int elements)

Static Private Attributes

- static final int LIMIT = 1024
- static Logger logger = LoggerFactory.getLogger(MPC_TripleDummy.class)

6.14.1 Detailed Description

Dummy to create the batches of triples and random shares in the "offline" phase

Author

Roman Walch

6.14.2 Member Function Documentation

6.14.2.1 produceRandomFieldElements()

```
static void at.iaik.mpc_acc.MPC_TripleDummy.produceRandomFieldElements (
            SpdzMascotDataSupplier supplier,
            int elements )  [static]
```

Precomputes shared random elements in the offline phase

Parameters

supplier	the data supplier
elements	the number of shared ranodm elements to be produced

6.14.2.2 produceTriples()

```
static void at.iaik.mpc_acc.MPC_TripleDummy.produceTriples (
             SpdzMascotDataSupplier supplier,
             int elements )  [static]
```

Precomputes shared triples in the offline phase

Parameters

supplier	the data supplier
elements	the number of triples to be produced

6.14.3 Member Data Documentation

6.14.3.1 LIMIT

```
final int at.iaik.mpc_acc.MPC_TripleDummy.LIMIT = 1024  [static], [private]
```

6.14.3.2 logger

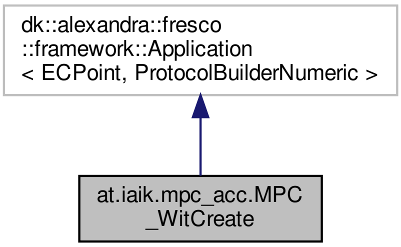
```
Logger at.iaik.mpc_acc.MPC_TripleDummy.logger = LoggerFactory.getLogger(MPC_TripleDummy.class)
[static], [private]
```

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

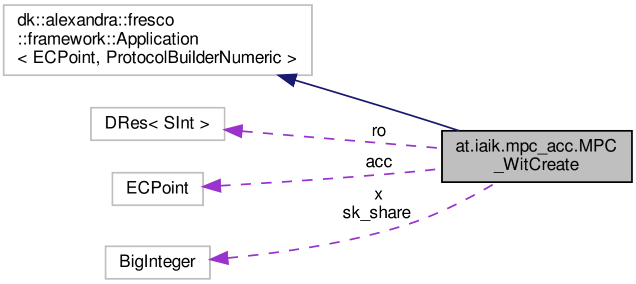
- MPC_TripleDummy.java

6.15 at.iaik.mpc_acc.MPC_WitCreate Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPC_WitCreate:


dk::alexandra::fresco ::framework::Application < ECPoint, ProtocolBuilderNumeric >
at.iaik.mpc_acc.MPC _WitCreate

Collaboration diagram for at.iaik.mpc_acc.MPC_WitCreate:


dk::alexandra::fresco ::framework::Application < ECPoint, ProtocolBuilderNumeric >
DRes< SInt >
ro
ECPoint
acc
x
sk_share
BigInteger
at.iaik.mpc_acc.MPC _WitCreate

Public Member Functions

- MPC_WitCreate (BigInteger sk_share, BigInteger x, ECPoint acc)
- DRes< ECPoint > buildComputation (ProtocolBuilderNumeric producer)

Private Attributes

- BigInteger sk_share
- BigInteger x
- ECPoint acc
- DRes< SInt > ro

6.15.1 Detailed Description

Creates the witness to a member of the accumulator

Author

Roman Walch

6.15.2 Constructor & Destructor Documentation

6.15.2.1 MPC_WitCreate()

```
at.iaik.mpc_acc.MPC_WitCreate.MPC_WitCreate (
             BigInteger sk_share,
             BigInteger x,
             ECPoint acc )
```

Construct a witness computation object

Parameters

sk_share	the shared secret key
x	the element
wit	the accumulator

6.15.3 Member Function Documentation

6.15.3.1 buildComputation()

```
DRes<ECPoint> at.iaik.mpc_acc.MPC_WitCreate.buildComputation (
             ProtocolBuilderNumeric producer )
```

Performs the MPC computation to create a witness of a member of the accumulator

Returns

the witness of the element

6.15.4 Member Data Documentation

6.15.4.1 acc

```
ECPoint at.iaik.mpc_acc.MPC_WitCreate.acc  [private]
```

6.15.4.2 ro

```
DRes<SInt> at.iaik.mpc_acc.MPC_WitCreate.ro [private]
```

6.15.4.3 sk_share

```
BigInteger at.iaik.mpc_acc.MPC_WitCreate.sk_share [private]
```

6.15.4.4 x

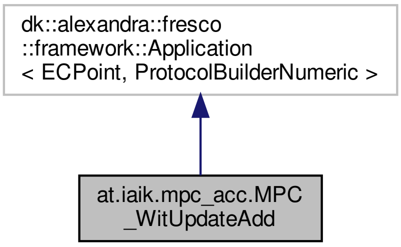
```
BigInteger at.iaik.mpc_acc.MPC_WitCreate.x [private]
```

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

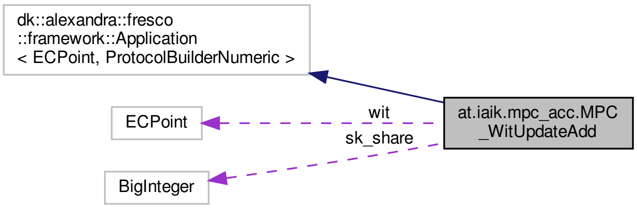
- MPC_WitCreate.java

6.16 at.iaik.mpc_acc.MPC_WitUpdateAdd Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPC_WitUpdateAdd:



Collaboration diagram for at.iaik.mpc_acc.MPC_WitUpdateAdd:



Public Member Functions

- MPC_WitUpdateAdd (BigInteger sk_share, ECPoint wit)
- DRes< ECPoint > buildComputation (ProtocolBuilderNumeric producer)

Private Attributes

- BigInteger sk_share
- ECPoint wit

6.16.1 Detailed Description

Implements the functionality to update a witness after an element is added to the accumulator

Author

Roman Walch

6.16.2 Constructor & Destructor Documentation

6.16.2.1 MPC_WitUpdateAdd()

```
at.iaik.mpc_acc.MPC_WitUpdateAdd.MPC_WitUpdateAdd (
             BigInteger sk_share,
             ECPoint wit )
```

Construct a witness update computation object for element addition

Parameters

sk_share	the shared secret key
x	the added element
wit	the witness to be updated

6.16.3 Member Function Documentation

6.16.3.1 buildComputation()

```
DRes<ECPoint> at.iaik.mpc_acc.MPC_WitUpdateAdd.buildComputation (
             ProtocolBuilderNumeric producer )
```

Performs the MPC computation to update a witness after an element is added to the accumulator

Returns

the updated witness

6.16.4 Member Data Documentation

6.16.4.1 sk_share

`BigInteger at.iaik.mpc_acc.MPC_WitUpdateAdd.sk_share [private]`

6.16.4.2 wit

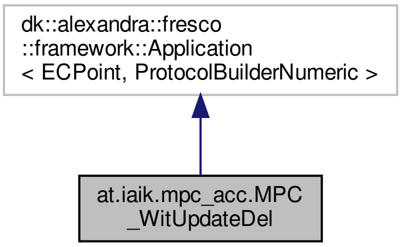
`ECPoint at.iaik.mpc_acc.MPC_WitUpdateAdd.wit [private]`

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

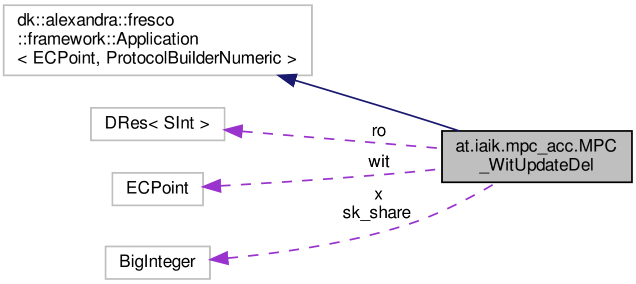
- MPC_WitUpdateAdd.java

6.17 at.iaik.mpc_acc.MPC_WitUpdateDel Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPC_WitUpdateDel:



Collaboration diagram for at.iaik.mpc_acc.MPC_WitUpdateDel:



Public Member Functions

- MPC_WitUpdateDel (BigInteger sk_share, BigInteger x, ECPoint wit)
- DRes< ECPoint > buildComputation (ProtocolBuilderNumeric producer)

Private Attributes

- BigInteger sk_share
- BigInteger x
- ECPoint wit
- DRes< SInt > ro

6.17.1 Detailed Description

Implements the functionality to update a witness after an element is removed from the accumulator

Author

Roman Walch

6.17.2 Constructor & Destructor Documentation

6.17.2.1 MPC_WitUpdateDel()

```
at.iaik.mpc_acc.MPC_WitUpdateDel.MPC_WitUpdateDel (
             BigInteger sk_share,
             BigInteger x,
             ECPoint wit )
```

Construct a witness update computation object for element removal

Parameters

sk_share	the shared secret key
x	the removed element
wit	the witness to be updated

6.17.3 Member Function Documentation

6.17.3.1 buildComputation()

```
DRes<ECPoint> at.iaik.mpc_acc.MPC_WitUpdateDel.buildComputation (
             ProtocolBuilderNumeric producer )
```

Performs the MPC computation to update a witness after an element is removed from the accumulator

Returns

the updated witness

6.17.4 Member Data Documentation

6.17.4.1 ro

```
DRes<SInt> at.iaik.mpc_acc.MPC_WitUpdateDel.ro  [private]
```

6.17.4.2 sk_share

```
BigInteger at.iaik.mpc_acc.MPC_WitUpdateDel.sk_share  [private]
```

6.17.4.3 wit

```
ECPoint at.iaik.mpc_acc.MPC_WitUpdateDel.wit  [private]
```

6.17.4.4 x

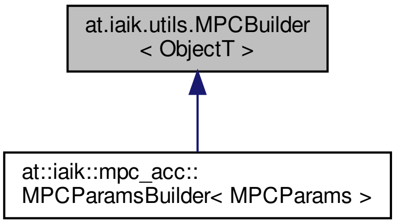
```
BigInteger at.iaik.mpc_acc.MPC_WitUpdateDel.x  [private]
```

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

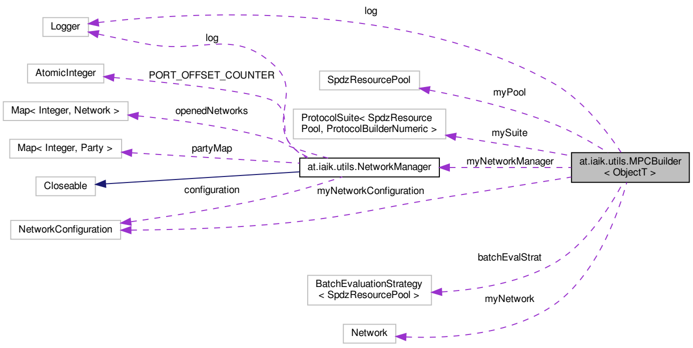
- MPC_WitUpdateDel.java

6.18 at.iaik.utils.MPCBuilder< ObjectT > Class Template Reference

Inheritance diagram for at.iaik.utils.MPCBuilder< ObjectT >:



Collaboration diagram for at.iaik.utils.MPCBuilder< ObjectT >:



Public Member Functions

- MPCBuilder (boolean logging)
- MPCBuilder< ObjectT > withID (int id)
- MPCBuilder< ObjectT > withResourcePool (PreprocessingStrategy strategy, BigInteger modulus) throws ParseException
- MPCBuilder< ObjectT > withSpdzLength (int maxBitLength)
- MPCBuilder< ObjectT > withBatchEvalStrat (EvaluationStrategy strat)
- MPCBuilder< ObjectT > withNetwork (Map< Integer, Party > parties, Party myParty) throws ParseException
- abstract ObjectT build ()

Static Protected Member Functions

- static Drbg getDrbg (int myId, int prgSeedLength)
- static SpdzSInt [ ] computeSInts (DRes< List< DRes< SInt >>> pipe)
- static DRes< List< DRes< SInt > > > createPipe (int myId, int noOfPlayers, int pipeLength, Network pipeNetwork, SpdzMascotDataSupplier tripleSupplier, int maxBitLength)
- static void evaluate (ProtocolBuilderNumeric spdzBuilder, SpdzResourcePool tripleResourcePool, Network network, SpdzProtocolSuite spdzProtocolSuite)
- static Map< Integer, RotList > getSeedOts (int myId, List< Integer > partyIds, int prgSeedLength, Drbg drbg, Network network)

Protected Attributes

- int numberOfParties
- int myID
- SpdzResourcePool myPool
- ProtocolSuite< SpdzResourcePool, ProtocolBuilderNumeric > mySuite
- NetworkConfiguration myNetworkConfiguration
- BatchEvaluationStrategy< SpdzResourcePool > batchEvalStrat
- Network myNetwork
- int maxBitLength
- NetworkManager myNetworkManager
- Logger log = LoggerFactory.getLogger(MPCBuilder.class)
- boolean logging

6.18.1 Detailed Description

The abstract builder class to instantiate the application objects. This builder takes care of the boiler plate code which is required to setup the framework. It also handles the use case specific input, these two parts could be separated at some point..

Parameters

<ObjectT>	The Object type the Builder extension wants to implement (either client or host)

Author

Fabian Schmid

6.18.2 Constructor & Destructor Documentation

6.18.2.1 MPCBuilder()

```
at.iaik.utils.MPCBuilder< ObjectT >.MPCBuilder (
            boolean logging )
```

Creating the builder following the builder pattern.

Parameters

logging	whether logging is activated during the computation

6.18.3 Member Function Documentation

6.18.3.1 build()

```
abstract ObjectT at.iaik.utils.MPCBuilder< ObjectT >.build ( )  [abstract]
```

This function has to be implemented in each child

Returns

The Object of the class which is built here

6.18.3.2 computeSInts()

```
static SpdzSInt [] at.iaik.utils.MPCBuilder< ObjectT >.computeSInts (
            DRes< List< DRes< SInt >>> pipe )  [static], [protected]
```

Converting the List of SInts that the are the pipe into an array of SpdzSInts

Parameters

pipe	the pipe to be converted

Returns

the array of SpdzSInt

6.18.3.3 createPipe()

```
static DRes<List<DRes<SInt> > > at.iaik.utils.MPCBuilder< ObjectT >.createPipe (
            int myId,
            int noOfPlayers,
            int pipeLength,
            Network pipeNetwork,
            SpdzMascotDataSupplier tripleSupplier,
            int maxBitLength ) [static], [protected]
```

Creates a protocol for the exponentiation pipe.

Parameters

myId	my id
noOfPlayers	the number of players in the network
pipeLength	the required length of the new pipe
pipeNetwork	the network to be used to create the network in an MPC environment
tripleSupplier	A simple triple supplier used to do the pipe creation using the MASCOT protocol
maxBitLength	the maximum bit length of variables in this application

Returns

The newly created pipe

6.18.3.4 evaluate()

```
static void at.iaik.utils.MPCBuilder< ObjectT >.evaluate (
            ProtocolBuilderNumeric spdzBuilder,
            SpdzResourcePool tripleResourcePool,
            Network network,
            SpdzProtocolSuite spdzProtocolSuite ) [static], [protected]
```

Evaluating the pipe creation protocol, so that the defered pipe can be used

Parameters

spdzBuilder	the protocolbuilder used to do MPC operations
tripleResourcePool	The resource pool used to execute this MPC protocol
network	The network used for communication
spdzProtocolSuite	the protocolSuite instance used

6.18.3.5 getDrbg()

```
static Drbg at.iaik.utils.MPCBuilder< ObjectT >.getDrbg (
            int myId,
            int prgSeedLength )  [static], [protected]
```

Auxiliary function when initiating the MASCOT protocol in the resource pool function

Parameters

myId	my id
prgSeedLength	the seed length for the deterministic random bit generator

Returns

a new Drbg instance

6.18.3.6 getSeedOts()

```
static Map<Integer, RotList> at.iaik.utils.MPCBuilder< ObjectT >.getSeedOts (
            int myId,
            List< Integer > partyIds,
            int prgSeedLength,
            Drbg drbg,
            Network network )  [static], [protected]
```

An auxiliary function used for the oblivious transfer necessary to initiate the MASCOT protocol.

Parameters

myId	my id
partyIds	a list of all ids in the network
prgSeedLength	the length of the random seed
drbg	the actual random bit generator instance
network	the network used to do the oblivious transfer

Returns

The map of SeedOts

6.18.3.7 withBatchEvalStrat()

```
MPCBuilder<ObjectT> at.iaik.utils.MPCBuilder< ObjectT >.withBatchEvalStrat (
            EvaluationStrategy strat )
```

Set the BatchEvaluationStrategy accordingly, get the loggingDecorator if logging is activated

Parameters

strat	the evaluation strategy to be used

Returns

this

6.18.3.8 withID()

```
MPCBuilder<ObjectT> at.iaik.utils.MPCBuilder< ObjectT >.withID (
            int id )
```

Setting the id of the application object

Parameters

id	the id to be set

Returns

this

6.18.3.9 withNetwork()

```
MPCBuilder<ObjectT> at.iaik.utils.MPCBuilder< ObjectT >.withNetwork (
            Map< Integer, Party > parties,
            Party myParty ) throws ParseException
```

Setting up a network manager and creating the first network to communicate with the other parties

Parameters

parties	the parties to connect with
myParty	this party

Returns

this

Exceptions

ParseException	is thrown, if myParty is not contained in the map of parties

6.18.3.10 withResourcePool()

```
MPCBuilder<ObjectT> at.iaik.utils.MPCBuilder< ObjectT >.withResourcePool (
             PreprocessingStrategy strategy,
             BigInteger modulus ) throws ParseException
```

Initializing the resourcePool Object required by the framework. MASCOT has to be used as a preprocessingStrategy to achieve active security

Parameters

strategy	the PreprocessingStrategy used
modBitLength	the bitLength of the modulus (128 bit is recommended)

Returns

this

Exceptions

ParseException	thrown when preprocessingStrategy is unknown

6.18.3.11 withSpdzLength()

```
MPCBuilder<ObjectT> at.iaik.utils.MPCBuilder< ObjectT >.withSpdzLength (
             int maxBitLength )
```

Instantiating the SpdzProtocolSuite required for the computation

Parameters

maxBitLength	the maximum number of bits for each shared variable

Returns

this.

6.18.4 Member Data Documentation

6.18.4.1 batchEvalStrat

```
BatchEvaluationStrategy<SpdzResourcePool> at.iaik.utils.MPCBuilder< ObjectT >.batchEvalStrat
[protected]
```

6.18.4.2 log

```
Logger at.iaik.utils.MPCBuilder< ObjectT >.log = LoggerFactory.getLogger(MPCBuilder.class)
[protected]
```

6.18.4.3 logging

```
boolean at.iaik.utils.MPCBuilder< ObjectT >.logging  [protected]
```

6.18.4.4 maxBitLength

```
int at.iaik.utils.MPCBuilder< ObjectT >.maxBitLength  [protected]
```

6.18.4.5 myID

```
int at.iaik.utils.MPCBuilder< ObjectT >.myID  [protected]
```

6.18.4.6 myNetwork

```
Network at.iaik.utils.MPCBuilder< ObjectT >.myNetwork  [protected]
```

6.18.4.7 myNetworkConfiguration

```
NetworkConfiguration at.iaik.utils.MPCBuilder< ObjectT >.myNetworkConfiguration  [protected]
```

6.18.4.8 myNetworkManager

```
NetworkManager at.iaik.utils.MPCBuilder< ObjectT >.myNetworkManager  [protected]
```

6.18.4.9 myPool

```
SpdzResourcePool at.iaik.utils.MPCBuilder< ObjectT >.myPool  [protected]
```

6.18.4.10 mySuite

```
ProtocolSuite<SpdzResourcePool, ProtocolBuilderNumeric> at.iaik.utils.MPCBuilder< ObjectT
>.mySuite [protected]
```

6.18.4.11 numberOfParties

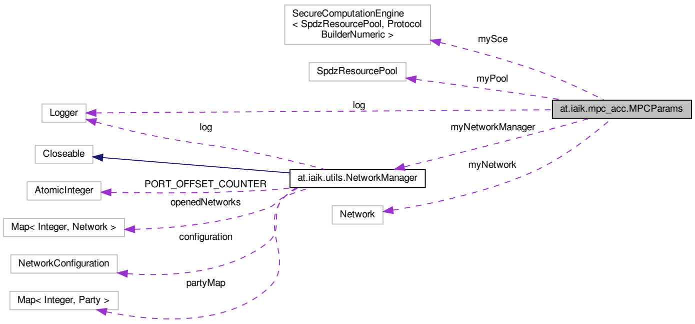
```
int at.iaik.utils.MPCBuilder< ObjectT >.numberOfParties [protected]
```

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

- MPCBuilder.java

6.19 at.iaik.mpc_acc.MPCParams Class Reference

Collaboration diagram for at.iaik.mpc_acc.MPCParams:



Public Member Functions

- NetworkManager getMyNetworkManager ()
- boolean isLogging ()
- Network getMyNetwork ()
- SpdzResourcePool getMyPool ()
- SecureComputationEngine< SpdzResourcePool, ProtocolBuilderNumeric > getMySce ()
- void closeNetwork ()
- void shutdownSce ()
- void shutdown ()
- void log (String string)

6.19.1 Detailed Description

MPCParams The build computation function adds the MPC functionality to the protocol builder

Author

Fabian Schmid

6.19.2 Member Function Documentation

6.19.2.1 closeNetwork()

```
void at.iaik.mpc_acc.MPCParams.closeNetwork ( )
```

Closes the network

6.19.2.2 getMyNetwork()

```
Network at.iaik.mpc_acc.MPCParams.getMyNetwork ( )
```

Getter for the network

6.19.2.3 getMyNetworkManager()

```
NetworkManager at.iaik.mpc_acc.MPCParams.getMyNetworkManager ( )
```

Getter for the network manager

6.19.2.4 getMyPool()

```
SpdzResourcePool at.iaik.mpc_acc.MPCParams.getMyPool ( )
```

Getter for the resource pool

6.19.2.5 getMySce()

```
SecureComputationEngine<SpdzResourcePool, ProtocolBuilderNumeric> at.iaik.mpc_acc.MPCParams.↩
getMySce ( )
```

Getter for the secure computation engine

6.19.2.6 isLogging()

```
boolean at.iaik.mpc_acc.MPCParams.isLogging ( )
```

6.19.2.7 log()

```
void at.iaik.mpc_acc.MPCParams.log (
             String string )
```

6.19.2.8 shutdown()

```
void at.iaik.mpc_acc.MPCParams.shutdown ( )
```

Closes the network and the secure compuatation engine.

6.19.2.9 shutdownSce()

```
void at.iaik.mpc_acc.MPCParams.shutdownSce ( )
```

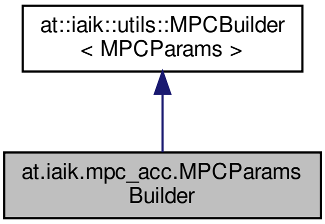
Closes the secure compuatation engine.

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

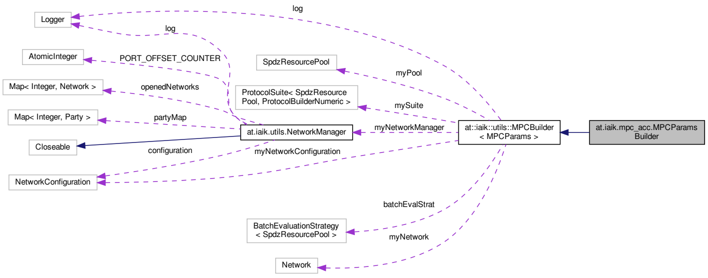
- MPCParams.java

6.20 at.iaik.mpc_acc.MPCParamsBuilder Class Reference

Inheritance diagram for at.iaik.mpc_acc.MPCParamsBuilder:



Collaboration diagram for at.iaik.mpc_acc.MPCParamsBuilder:



Public Member Functions

- MPCParamsBuilder (boolean logging)
- MPCParams build ()

Additional Inherited Members

6.20.1 Detailed Description

MPCParamsBuilder, extends the generic Builder and facilitates creating an instance of the client application: MPCParams

Author

Fabian Schmid

6.20.2 Constructor & Destructor Documentation

6.20.2.1 MPCParamsBuilder()

```
at.iaik.mpc_acc.MPCParamsBuilder.MPCParamsBuilder (
              boolean logging )
```

6.20.3 Member Function Documentation

6.20.3.1 build()

```
MPCParams at.iaik.mpc_acc.MPCParamsBuilder.build ( )
```

The MPCParams object is created and its members are set according to the members set in the parent builder.

Returns

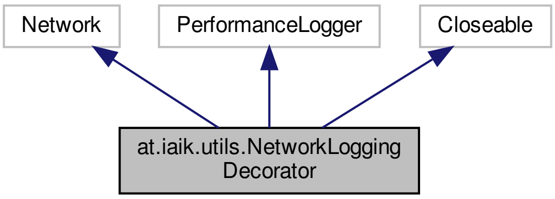
the fully initiated MPCParams object.

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

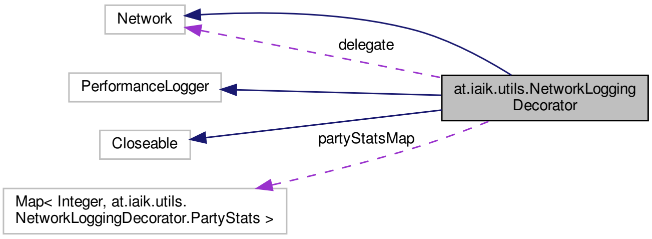
- MPCParamsBuilder.java

6.21 at.iaik.utils.NetworkLoggingDecorator Class Reference

Inheritance diagram for at.iaik.utils.NetworkLoggingDecorator:



Collaboration diagram for at.iaik.utils.NetworkLoggingDecorator:


Network
delegate
PerformanceLogger
at.iaik.utils.NetworkLogging Decorator
Closeable
partyStatsMap
Map< Integer, at.iaik.utils. NetworkLoggingDecorator.PartyStats >

Classes

- class PartyStats

Public Member Functions

- NetworkLoggingDecorator (Network network)
- byte [ ] receive (int partyId)
- int getNoOfParties ()
- void send (int partyId, byte[ ] data)
- void reset ()
- void close () throws IOException
- Map< String, Long > getLoggedValues ()

Static Public Attributes

- static final String NETWORK_PARTY_BYTES = "Amount of bytes received pr. party"
- static final String NETWORK_TOTAL_BYTES = "Total amount of bytes received"
- static final String NETWORK_TOTAL_BATCHES = "Total amount of batches received"

Private Attributes

- Network delegate
- Map< Integer, PartyStats > partyStatsMap = new HashMap<>()

6.21.1 Detailed Description

A decorator for the network to extract the logged data to the network manager

Author

Fabian Schmid

6.21.2 Constructor & Destructor Documentation

6.21.2.1 NetworkLoggingDecorator()

```
at.iaik.utils.NetworkLoggingDecorator.NetworkLoggingDecorator (
             Network network )
```

creates the decorator for the given network

Parameters

network	the delegate network to be decorated

6.21.3 Member Function Documentation

6.21.3.1 close()

```
void at.iaik.utils.NetworkLoggingDecorator.close ( ) throws IOException
```

6.21.3.2 getLoggedValues()

```
Map<String, Long> at.iaik.utils.NetworkLoggingDecorator.getLoggedValues ( )
```

get the logged values

Returns

Return the entire map of party stats - used in the network Manager

6.21.3.3 getNoOfParties()

```
int at.iaik.utils.NetworkLoggingDecorator.getNoOfParties ( )
```

6.21.3.4 receive()

```
byte [] at.iaik.utils.NetworkLoggingDecorator.receive (
             int partyId )
```

Upon receiving from a party, the received bytes are stored in a map

Parameters

party↩Id	the party from which to receive

Returns

the received bytes

6.21.3.5 reset()

```
void at.iaik.utils.NetworkLoggingDecorator.reset ( )
```

6.21.3.6 send()

```
void at.iaik.utils.NetworkLoggingDecorator.send (
             int partyId,
             byte [] data )
```

6.21.4 Member Data Documentation

6.21.4.1 delegate

```
Network at.iaik.utils.NetworkLoggingDecorator.delegate  [private]
```

6.21.4.2 NETWORK_PARTY_BYTES

```
final String at.iaik.utils.NetworkLoggingDecorator.NETWORK_PARTY_BYTES = "Amount of bytes
received pr.  party"  [static]
```

6.21.4.3 NETWORK_TOTAL_BATCHES

```
final String at.iaik.utils.NetworkLoggingDecorator.NETWORK_TOTAL_BATCHES = "Total amount of
batches received"  [static]
```

6.21.4.4 NETWORK_TOTAL_BYTES

```
final String at.iaik.utils.NetworkLoggingDecorator.NETWORK_TOTAL_BYTES = "Total amount of
bytes received"  [static]
```

6.21.4.5 partyStatsMap

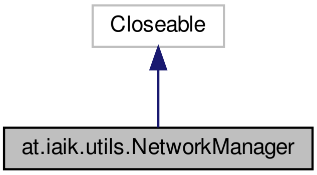
```
Map<Integer, PartyStats> at.iaik.utils.NetworkLoggingDecorator.partyStatsMap = new Hash←-
Map<>()  [private]
```

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

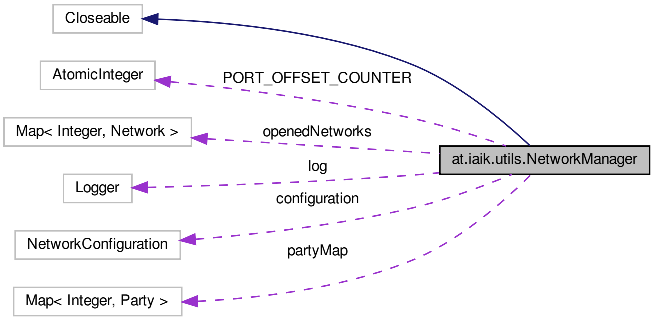
- NetworkLoggingDecorator.java

6.22 at.iaik.utils.NetworkManager Class Reference

Inheritance diagram for at.iaik.utils.NetworkManager:



Collaboration diagram for at.iaik.utils.NetworkManager:



Public Member Functions

- NetworkManager (NetworkConfiguration configuration, boolean logging, Map< Integer, Party > parties)
- Network createExtraNetwork ()
- Map< Integer, Party > getParties ()
- Map< String, Long > getLoggedValues ()
- void reset ()
- void close ()

Static Public Member Functions

- static boolean equalParties (Party p1, Party p2)
- static Map< Integer, Party > getPartyMap (List< Map< Integer, Party >> partyList, Party myParty)

Private Member Functions

- void log (String string)
- NetworkConfiguration UpdateConfiguration ()
- void close (Closeable closeable)

Private Attributes

- final AtomicInteger PORT_OFFSET_COUNTER = new AtomicInteger(0)
- final int PORT_INCREMENT = 50
- final Map< Integer, Network > openedNetworks
- final Map< Integer, Party > partyMap
- final NetworkConfiguration configuration
- int portOffset
- final boolean logging

Static Private Attributes

- static Logger log = LoggerFactory.getLogger(NetworkManager.class)

6.22.1 Detailed Description

The NetworkManager enables multiple networks with the same participants to be managed.

Author

Fabian Schmid

6.22.2 Constructor & Destructor Documentation

6.22.2.1 NetworkManager()

```
at.iaik.utils.NetworkManager.NetworkManager (
             NetworkConfiguration configuration,
             boolean logging,
             Map< Integer, Party > parties )
```

Create a new NetworkManager

Parameters

configuration	the configuration
logging	whether this application uses logging
parties	the parties with which the networks are created

6.22.3 Member Function Documentation

6.22.3.1 close() [1/2]

```
void at.iaik.utils.NetworkManager.close ( )
```

closes the networkManager and all the networks

6.22.3.2 close() [2/2]

```
void at.iaik.utils.NetworkManager.close (
             Closeable closeable )  [private]
```

closes a specific network

Parameters

closeable	the network to be closed

6.22.3.3 createExtraNetwork()

```
Network at.iaik.utils.NetworkManager.createExtraNetwork ( )
```

Create another network with the same parties but different ports, just to not interfere with the other protocols using the same network.

Returns

the newly created network

6.22.3.4 equalParties()

```
static boolean at.iaik.utils.NetworkManager.equalParties (
            Party p1,
            Party p2 )  [static]
```

Compare two parties with each other

Parameters

p1	party 1
p2	party 2

Returns

is p1 == p2? (true/false)

6.22.3.5 getLoggedValues()

```
Map<String, Long> at.iaik.utils.NetworkManager.getLoggedValues ( )
```

If logging is activated, returns all the logged values from all networks this is used to gather the amount of network traffic received from all the parties

Returns

The string map of all the loggings

6.22.3.6 getParties()

```
Map<Integer, Party> at.iaik.utils.NetworkManager.getParties ( )
```

6.22.3.7 getPartyMap()

```
static Map<Integer, Party> at.iaik.utils.NetworkManager.getPartyMap (
             List< Map< Integer, Party >> partyList,
             Party myParty )  [static]
```

in a multithreaded setting there are several parties with the same id... Returns that party map, which contains my party instance

Parameters

partyList	the list of all party maps
myParty	the instance of my party object

Returns

the party map which this party belongs to

6.22.3.8 log()

```
void at.iaik.utils.NetworkManager.log (
             String string )  [private]
```

6.22.3.9 reset()

```
void at.iaik.utils.NetworkManager.reset ( )
```

6.22.3.10 UpdateConfiguration()

```
NetworkConfiguration at.iaik.utils.NetworkManager.UpdateConfiguration ( )  [private]
```

create new Network configuration by incrementing the ports

Returns

the new network configuration

6.22.4 Member Data Documentation

6.22.4.1 configuration

```
final NetworkConfiguration at.iaik.utils.NetworkManager.configuration  [private]
```

6.22.4.2 log

```
Logger at.iaik.utils.NetworkManager.log = LoggerFactory.getLogger(NetworkManager.class)  [static],
[private]
```

6.22.4.3 logging

```
final boolean at.iaik.utils.NetworkManager.logging  [private]
```

6.22.4.4 openedNetworks

```
final Map<Integer, Network> at.iaik.utils.NetworkManager.openedNetworks  [private]
```

6.22.4.5 partyMap

```
final Map<Integer, Party> at.iaik.utils.NetworkManager.partyMap  [private]
```

6.22.4.6 PORT_INCREMENT

```
final int at.iaik.utils.NetworkManager.PORT_INCREMENT = 50  [private]
```

6.22.4.7 PORT_OFFSET_COUNTER

```
final AtomicInteger at.iaik.utils.NetworkManager.PORT_OFFSET_COUNTER = new AtomicInteger(0)
[private]
```

6.22.4.8 portOffset

```
int at.iaik.utils.NetworkManager.portOffset  [private]
```

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

- NetworkManager.java

6.23 at.iaik.mpc_acc.Node Class Reference

Collaboration diagram for at.iaik.mpc_acc.Node:

at.iaik.mpc_acc.Node
right
left

Public Member Functions

- Node (byte[ ] digest)
- Node (Node left, Node right)
- void compute_digest ()

Static Public Member Functions

- static byte [ ] compute_digest (byte[ ] left, byte[ ] right)

Public Attributes

- byte [ ] digest
- Node left
- Node right

6.23.1 Detailed Description

A class containing the node of a Merkle Tree accumulator

Author

Roman Walch

6.23.2 Constructor & Destructor Documentation

6.23.2.1 Node() [1/2]

```
at.iaik.mpc_acc.Node.Node (
             byte [ ] digest )
```

Initialize a node without child-nodes

Parameters

digest	the digest of the node

6.23.2.2 Node() [2/2]

```
at.iaik.mpc_acc.Node.Node (
            Node left,
            Node right )
```

Initialize a node with two child nodes

Parameters

left	the left child
right	the right child

6.23.3 Member Function Documentation

6.23.3.1 compute_digest() [1/2]

```
void at.iaik.mpc_acc.Node.compute_digest ( )
```

Computes the digest of the node.

6.23.3.2 compute_digest() [2/2]

```
static byte [] at.iaik.mpc_acc.Node.compute_digest (
            byte [] left,
            byte [] right )  [static]
```

computes the digest of the node.

Parameters

left	the left child node digest
right	the right child node digest

Returns

the digest of the node

6.23.4 Member Data Documentation

6.23.4.1 digest

```
byte [] at.iaik.mpc_acc.Node.digest
```

6.23.4.2 left

```
Node at.iaik.mpc_acc.Node.left
```

6.23.4.3 right

```
Node at.iaik.mpc_acc.Node.right
```

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

- Node.java

6.24 at.iaik.utils.NetworkLoggingDecorator.PartyStats Class Reference

Public Member Functions

- void recordTransmission (int noBytes)

Private Attributes

- long count
- long noBytes

6.24.1 Detailed Description

The data structure to store information about received transmissions

6.24.2 Member Function Documentation

6.24.2.1 recordTransmission()

```
void at.iaik.utils.NetworkLoggingDecorator.PartyStats.recordTransmission (
              int noBytes )
```

Upon receiving information, record transmission is called

Parameters

noBytes	the number of bytes received

6.24.3 Member Data Documentation

6.24.3.1 count

```
long at.iaik.utils.NetworkLoggingDecorator.PartyStats.count  [private]
```

6.24.3.2 noBytes

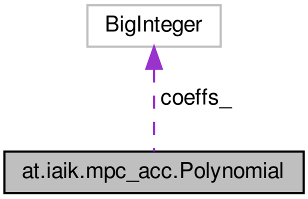
```
long at.iaik.utils.NetworkLoggingDecorator.PartyStats.noBytes  [private]
```

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

- NetworkLoggingDecorator.java

6.25 at.iaik.mpc_acc.Polynomial Class Reference

Collaboration diagram for at.iaik.mpc_acc.Polynomial:



Public Member Functions

- Polynomial (int degree, BigInteger order)
- int powX (int currentDegree)
- String toString ()

Static Public Member Functions

- static Polynomial expand (BigInteger[ ] roots, BigInteger order)

Private Attributes

- final BigInteger [ ] coeffs_
- final int degree_

6.25.1 Detailed Description

This class represents a monic, reducible polynomial over Z_p

Author

david
chanser

6.25.2 Constructor & Destructor Documentation

6.25.2.1 Polynomial()

```
at.iaik.mpc_acc.Polynomial.Polynomial (
            int degree,
            BigInteger order )
```

c'tor

Parameters

degree	the degree of the polynomial
order	the order to be applied to the coefficients

6.25.3 Member Function Documentation

6.25.3.1 expand()

```
static Polynomial at.iaik.mpc_acc.Polynomial.expand (
            BigInteger [ ] roots,
            BigInteger order ) [static]
```

Expands a polynomial of the form ${i=0}^{n} (X + A_i)$

Parameters

roots	The list containing the root A_i
order	modulus

Returns

The expanded polynomial

6.25.3.2 powX()

```
int at.iaik.mpc_acc.Polynomial.powX (
             int currentDegree )
```

Increases the degree_ of the polynomial by 1. This method is immutable.

Parameters

currentDegree	the current degree_

Returns

the resulting polynomial

6.25.3.3 toString()

```
String at.iaik.mpc_acc.Polynomial.toString ( )
```

6.25.4 Member Data Documentation

6.25.4.1 coeffs_

```
final BigInteger [ ] at.iaik.mpc_acc.Polynomial.coeffs_  [private]
```

6.25.4.2 degree_

```
final int at.iaik.mpc_acc.Polynomial.degree_  [private]
```

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

- Polynomial.java

6.26 at.iaik.mpc_acc.MerkleTree.Position Enum Reference

Public Attributes

- left
- right

6.26.1 Detailed Description

Enum for the position of the node as part of the proof.

6.26.2 Member Data Documentation

6.26.2.1 left

```
at.iaik.mpc_acc.MerkleTree.Position.left
```

6.26.2.2 right

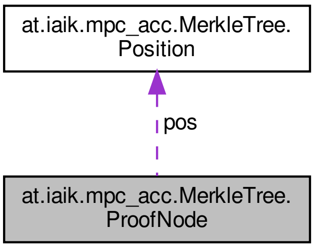
```
at.iaik.mpc_acc.MerkleTree.Position.right
```

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

- MerkleTree.java

6.27 at.iaik.mpc_acc.MerkleTree.ProofNode Class Reference

Collaboration diagram for at.iaik.mpc_acc.MerkleTree.ProofNode:



Public Member Functions

- ProofNode (byte[ ] digest, Position pos)

Public Attributes

- byte [ ] digest
- Position pos

6.27.1 Constructor & Destructor Documentation

6.27.1.1 ProofNode()

```
at.iaik.mpc_acc.MerkleTree.ProofNode.ProofNode (
            byte [ ] digest,
            Position pos )
```

6.27.2 Member Data Documentation

6.27.2.1 digest

```
byte [ ] at.iaik.mpc_acc.MerkleTree.ProofNode.digest
```

6.27.2.2 pos

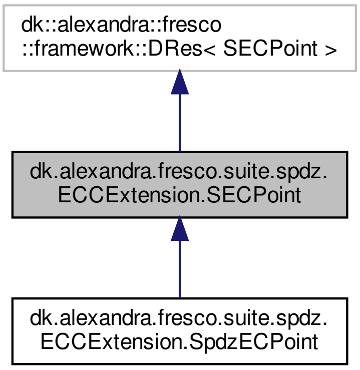
```
Position at.iaik.mpc_acc.MerkleTree.ProofNode.pos
```

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

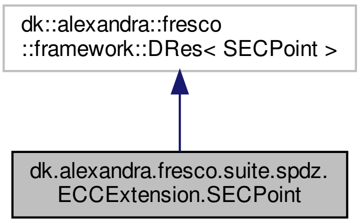
- MerkleTree.java

6.28 dk.alexandra.fresco.suite.spdz.ECCExtension.SECPoint Interface Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SECPoint:


dk::alexandra::fresco ::framework::DRes< SECPoint >
dk.alexandra.fresco.suite.spdz. ECCExtension.SECPoint
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzECPoint

Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SECPoint:


dk::alexandra::fresco ::framework::DRes< SECPoint >
dk.alexandra.fresco.suite.spdz. ECCExtension.SECPoint

6.28.1 Detailed Description

A interface class to make the Secure Elliptic Curve Point usable by FRESCO's applications

Author

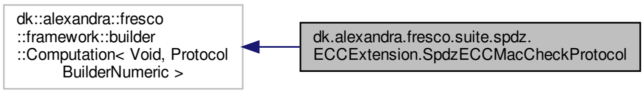
Roman Walch

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

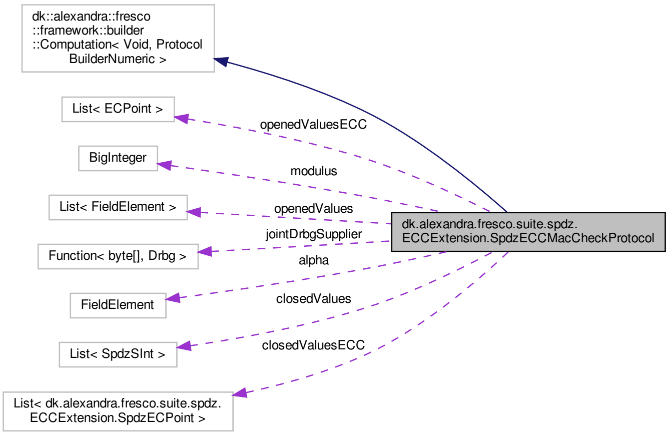
- SECPoint.java

6.29 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol Class Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol:


dk::alexandra::fresco ::framework::builder ::Computation< Void, Protocol BuilderNumeric >
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzECCMacCheckProtocol

Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol:


dk::alexandra::fresco ::framework::builder ::Computation< Void, Protocol BuilderNumeric >
List< ECPoint >
openedValuesECC
BigInteger
modulus
List< FieldElement >
openedValues
jointDrbgSupplier
Function< byte[], Drbg >
alpha
FieldElement
closedValues
List< SpdzSInt >
closedValuesECC
List< dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzECPoint >
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzECCMacCheckProtocol

Public Member Functions

- SpdzECCMacCheckProtocol (final Pair< List< SpdzSInt >, List< FieldElement >> toCheck, final Pair< List< SpdzECPoint >, List< ECPoint >> toCheckECC, final BigInteger modulus, final Function< byte[ ], Drbg > jointDrbgSupplier, final FieldElement alpha, final int drbgSeedBitLength)
- DRes< Void > buildComputation (ProtocolBuilderNumeric builder)

Private Member Functions

- FieldElement [ ] sampleRandomCoefficients (int numCoefficients, FieldDefinition fieldDefinition, Drbg jointDrbg)

Private Attributes

- final BigInteger modulus
- final List< SpdzSInt > closedValues
- final List< FieldElement > openedValues
- final List< SpdzECPoint > closedValuesECC
- final List< ECPoint > openedValuesECC
- final FieldElement alpha
- final Function< byte[ ], Drbg > jointDrbgSupplier
- final int drbgByteLength

6.29.1 Detailed Description

Protocol which handles the MAC check internal to SPDZ. If this protocol reaches the end, no malicious activity was detected and the storage is reset. This class is an Extension to the SpdzMacCheckProtocol class according to https://eprint.iacr.org/2019/768

Author

Roman Walch

6.29.2 Constructor & Destructor Documentation

6.29.2.1 SpdzECCMacCheckProtocol()

```
dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.SpdzECCMacCheckProtocol (
             final Pair< List< SpdzSInt >, List< FieldElement >> toCheck,
             final Pair< List< SpdzECPoint >, List< ECPoint >> toCheckECC,
             final BigInteger modulus,
             final Function< byte[], Drbg > jointDrbgSupplier,
             final FieldElement alpha,
             final int drbgSeedBitLength )
```

A Constructor for ECCMacCheckProtocol. This protocol handles the MAC check internal to SPDZ for ECC Points. If this protocol reaches the end, no malicious activity was detected and the storage is reset.

Parameters

toCheck	opened values and corresponding macs to check
modulus	the global modulus used
jointDrbgSupplier	supplier of DRBG to be used for joint randomness
alpha	this party's key share
drbgSeedBitLength	seed length for local DRBG

6.29.3 Member Function Documentation

6.29.3.1 buildComputation()

```
DRes<Void> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.buildComputation
(
             ProtocolBuilderNumeric builder )
```

Adds the ECC MacCheck to the execution queue.

Parameters

builder	The builder used to build the computation

6.29.3.2 sampleRandomCoefficients()

```
FieldElement [] dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.sample↩
RandomCoefficients (
             int numCoefficients,
             FieldDefinition fieldDefinition,
             Drbg jointDrbg )  [private]
```

This member samples the random coefficients used during the MAC check

6.29.4 Member Data Documentation

6.29.4.1 alpha

```
final FieldElement dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.alpha
[private]
```

this party's key share

6.29.4.2 closedValues

```
final List<SpdzSInt> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.↩
closedValues  [private]
```

List of closed Integers

6.29.4.3 closedValuesECC

```
final List<SpdzECPoint> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.↩
closedValuesECC  [private]
```

List of closed ECC Points

6.29.4.4 drbgByteLength

```
final int dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.drbgByteLength
[private]
```

seed length for local DRBG

6.29.4.5 jointDrbgSupplier

```
final Function<byte[], Drbg> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheck↩
Protocol.jointDrbgSupplier  [private]
```

supplier of DRBG to be used for joint randomness

6.29.4.6 modulus

```
final BigInteger dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.modulus
[private]
```

The ECC modulus

6.29.4.7 openedValues

```
final List<FieldElement> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.↩
openedValues  [private]
```

List of opened Integers

6.29.4.8 openedValuesECC

```
final List<ECPoint> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol.↩
openedValuesECC  [private]
```

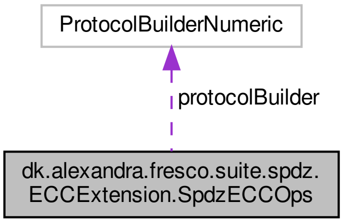
List of opened ECC Points

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

- SpdzECCMacCheckProtocol.java

6.30 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps Class Reference

Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps:



Public Member Functions

- SpdzECCOps (ProtocolBuilderNumeric protocolBuilder)
- DRes< SInt > knownScalar (BigInteger k)
- DRes< SECPoint > knownMultiply (ECPoint p, BigInteger k)
- DRes< SECPoint > multiply (ECPoint p, DRes< SInt > k)
- DRes< ECPoint > open (DRes< SECPoint > secretshare)

Private Attributes

- final ProtocolBuilderNumeric protocolBuilder

6.30.1 Detailed Description

A class containing wrappers to append elliptic curve computations to FRESCO's applications

Author

Roman Walch

6.30.2 Constructor & Destructor Documentation

6.30.2.1 SpdzECCOps()

```
dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps.SpdzECCOps (
            ProtocolBuilderNumeric protocolBuilder )
```

The constructor of the class. It sets the used builder

Parameters

protocolBuilder	the used protocol builder

6.30.3 Member Function Documentation

6.30.3.1 knownMultiply()

```
DRes<SECPoint> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps.knownMultiply (
            ECPoint p,
            BigInteger k )
```

Creates a valid shared integer from a previously shared value (which had no associated MAC) and multiplies it to a public ECC point

Parameters

p	The public ECC point
k	the previous shared value

Returns

the resulting shared ECC point

6.30.3.2 knownScalar()

```
DRes<SInt> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps.knownScalar (
            BigInteger k )
```

Creates a valid shared integer from a previously shared value (which had no associated MAC)

Parameters

k	the previous shared value

Returns

the valid shared integer

6.30.3.3 multiply()

```
DRes<SECPoint> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps.multiply (
            ECPoint p,
            DRes< SInt > k )
```

Multiplies a shared integer to an public ECC point

Parameters

p	The public ECC point
k	the shared integer

Returns

the resulting shared ECC point

6.30.3.4 open()

```
DRes<ECPoint> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps.open (
             DRes< SECPoint > secretshare )
```

Opens a shared ECC point

Parameters

secretshare	the shared ECC point

Returns

the public opened ECC point

6.30.4 Member Data Documentation

6.30.4.1 protocolBuilder

```
final ProtocolBuilderNumeric dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps.protocol↩
Builder  [private]
```

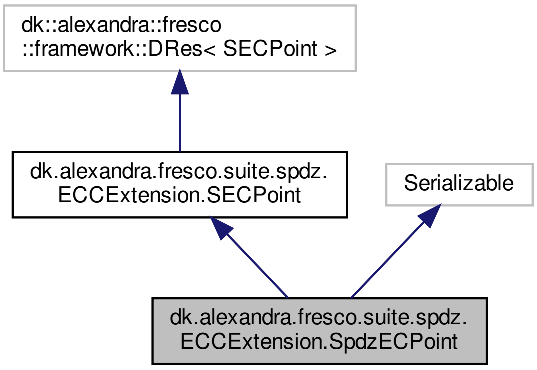
the used protocol builder

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

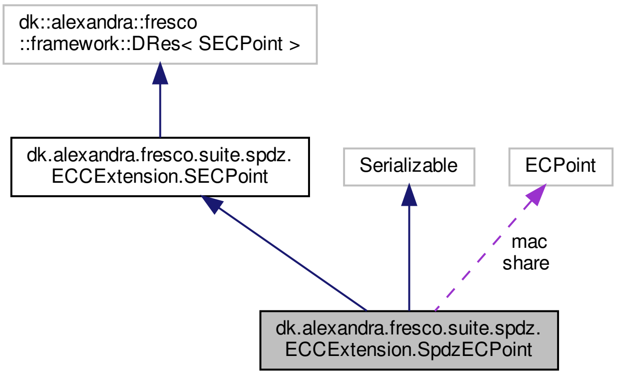
- SpdzECCOps.java

6.31 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint Class Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint:


dk::alexandra::fresco
::framework::DRes< SECPoint >
dk.alexandra.fresco.suite.spdz.
ECCExtension.SECPoint
Serializable
dk.alexandra.fresco.suite.spdz.
ECCExtension.SpdzECPoint

Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint:



Public Member Functions

- SpdzECPoint (ECPoint share, ECPoint mac)
- String toString ()
- SECPoint out ()
- ECPoint getShare ()
- ECPoint getMac ()
- byte [ ] serializeShare ()
- EllipticCurve getCurve ()

Static Public Member Functions

- static SpdzECPoint multiplyPoint (ECPoint p, BigInteger share, BigInteger mac)

Private Attributes

- final ECPoint share
- final ECPoint mac

Static Private Attributes

- static final long serialVersionUID = 5882876872861854360L

6.31.1 Detailed Description

A Spdz class for shared ECC points containing the MAC and share.

Author

Roman Walch

6.31.2 Constructor & Destructor Documentation

6.31.2.1 SpdzECPoint()

```
dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.SpdzECPoint (
            ECPoint share,
            ECPoint mac )
```

Constructor to initialize the shared ECC point.

Parameters

share	the ECC share
mac	the MAC

6.31.3 Member Function Documentation

6.31.3.1 getCurve()

```
EllipticCurve dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.getCurve ( )
```

Returns the curve of the shared ECC point

Returns

the used elliptic curve

6.31.3.2 getMac()

```
ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.getMac ( )
```

Getter for the MAC

Returns

the MAC

6.31.3.3 getShare()

```
ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.getShare ( )
```

Getter for the share

Returns

the share

6.31.3.4 multiplyPoint()

```
static SpdzECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.multiplyPoint (
             ECPoint p,
             BigInteger share,
             BigInteger mac )  [static]
```

Multiplies a shared integer to an public ECC point

Parameters

p	the public ECC point
share	the share of the shared integer
mac	the MAC of the shared integer

Returns

the resulting shared ECC point

6.31.3.5 out()

```
SECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.out ( )
```

Helper class to output the actual shared ECC point

6.31.3.6 serializeShare()

```
byte [] dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.serializeShare ( )
```

Serializes the shared ECC point

Returns

the serialization

6.31.3.7 toString()

```
String dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.toString ( )
```

Covnerts the shared ECC point to a string

6.31.4 Member Data Documentation

6.31.4.1 mac

```
final ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.mac  [private]
```

The corresponding MAC

6.31.4.2 serialVersionUID

```
final long dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.serialVersionUID = 588287687286185
[static], [private]
```

6.31.4.3 share

```
final ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint.share  [private]
```

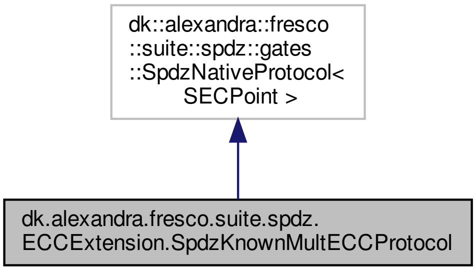
The share of the point

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

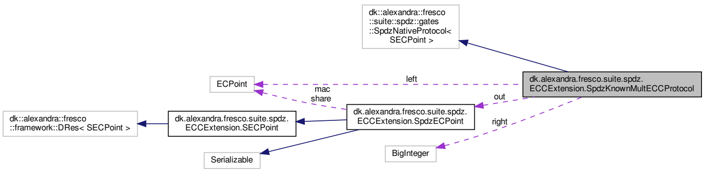
- SpdzECPoint.java

6.32 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol Class Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol:



Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol:



Public Member Functions

- SpdzKnownMultECCProtocol (ECPoint p, BigInteger k)
- EvaluationStatus evaluate (int round, SpdzResourcePool spdzResourcePool, Network network)
- SpdzECPoint out ()

Private Attributes

- ECPoint left
- BigInteger right
- SpdzECPoint out

6.32.1 Detailed Description

A Spdz protocol to create a valid shared integer from a previously shared value (which had no associated MAC) and multiplies it to a public ECC point

Author

Roman Walch

6.32.2 Constructor & Destructor Documentation

6.32.2.1 SpdzKnownMultECCProtocol()

```
dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol.SpdzKnownMultECCProtocol
(
             ECPoint p,
             BigInteger k )
```

The constructor to initialize the computation.

Parameters

p	the public ECC point
k	the previous shared value

6.32.3 Member Function Documentation

6.32.3.1 evaluate()

```
EvaluationStatus dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol.evaluate
(
             int round,
             SpdzResourcePool spdzResourcePool,
             Network network )
```

This members shares the previously shared value and multiplies the result to a public ECC point. FRESCO calls this member when executing the SpdzKnownMultECC computation.

6.32.3.2 out()

```
SpdzECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol.out ( )
```

Helper class to output the resulting shared ECC point

6.32.4 Member Data Documentation

6.32.4.1 left

```
ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol.left  [private]
```

the ECC point.

6.32.4.2 out

```
SpdzECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol.out  [private]
```

the resulting shared ECC point

6.32.4.3 right

```
BigInteger dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol.right  [private]
```

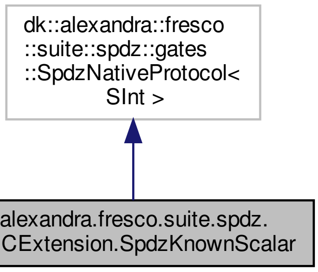
the previously shared Integer

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

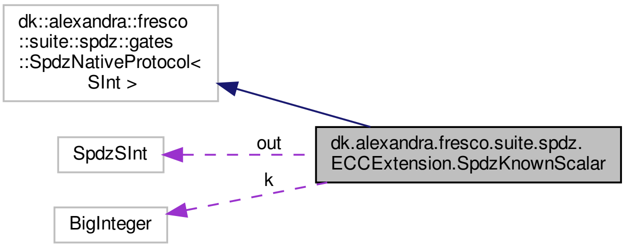
- SpdzKnownMultECCProtocol.java

6.33 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar Class Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar:


dk::alexandra::fresco ::suite::spdz::gates ::SpdzNativeProtocol< SInt >
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzKnownScalar

Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar:


dk::alexandra::fresco ::suite::spdz::gates ::SpdzNativeProtocol< SInt >
SpdzSInt
out
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzKnownScalar
k
BigInteger

Public Member Functions

- SpdzKnownScalar (BigInteger k)
- EvaluationStatus evaluate (int round, SpdzResourcePool spdzResourcePool, Network network)
- SInt out ()

Private Attributes

- BigInteger k
- SpdzSInt out

6.33.1 Detailed Description

A Spdz protocol to create a valid shared integer from a previously shared value (which had no associated MAC)

Author

Roman Walch

6.33.2 Constructor & Destructor Documentation

6.33.2.1 SpdzKnownScalar()

```
dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar.SpdzKnownScalar (
             BigInteger k )
```

The constructor to initialize the computation.

Parameters

k	the previous shared value

6.33.3 Member Function Documentation

6.33.3.1 evaluate()

```
EvaluationStatus dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar.evaluate (
             int round,
             SpdzResourcePool spdzResourcePool,
             Network network )
```

This members shares the previously shared value. FRESCO calls this member when executing the SpdzKnownScalar computation.

6.33.3.2 out()

```
SInt dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar.out ( )
```

Helper class to output the resulting shared Integer

6.33.4 Member Data Documentation

6.33.4.1 k

```
BigInteger dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar.k  [private]
```

the previous shared value

6.33.4.2 out

```
SpdzSInt dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar.out  [private]
```

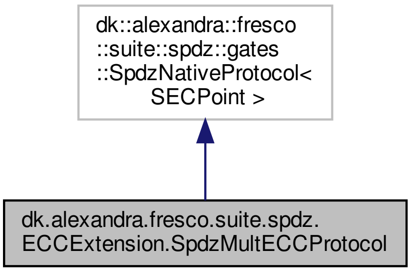
the resulting shared ECC point

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

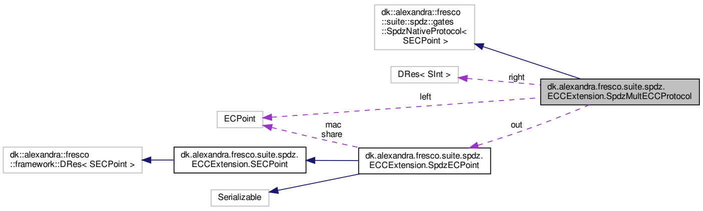
- SpdzKnownScalar.java

6.34 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol Class Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol:


dk::alexandra::fresco ::suite::spdz::gates ::SpdzNativeProtocol< SECPoint >
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzMultECCProtocol

Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol:


dk::alexandra::fresco ::suite::spdz::gates ::SpdzNativeProtocol< SECPoint >
DRes< SInt >
right
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzMultECCProtocol
left
ECPoint
mac
share
out
dk::alexandra::fresco ::framework::DRes< SECPoint >
dk.alexandra.fresco.suite.spdz. ECCExtension.SECPoint
dk.alexandra.fresco.suite.spdz. ECCExtension.SpdzECPoint
Serializable

Public Member Functions

- SpdzMultECCProtocol (ECPoint p, DRes< SInt > k)
- EvaluationStatus evaluate (int round, SpdzResourcePool spdzResourcePool, Network network)
- SpdzECPoint out ()

Private Attributes

- ECPoint left
- DRes< SInt > right
- SpdzECPoint out

6.34.1 Detailed Description

A Spdz protocol to multiply a shared integer to an public ECC point.

Author

Roman Walch

6.34.2 Constructor & Destructor Documentation

6.34.2.1 SpdzMultECCProtocol()

```
dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol.SpdzMultECCProtocol (
             ECPoint p,
             DRes< SInt > k )
```

The constructor to initialize the computation.

Parameters

p	the public ECC point
k	the shared integer

6.34.3 Member Function Documentation

6.34.3.1 evaluate()

```
EvaluationStatus dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol.evaluate (
             int round,
             SpdzResourcePool spdzResourcePool,
             Network network )
```

This member multiplies a shared integer to an public ECC point. FRESCO calls this member when executing the SpdzMultECC computation.

6.34.3.2 out()

`SpdzECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol.out ( )`

Helper class to output the resulting shared ECC point

6.34.4 Member Data Documentation

6.34.4.1 left

`ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol.left [private]`

the public ECC point

6.34.4.2 out

`SpdzECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol.out [private]`

the resulting shared ECC point

6.34.4.3 right

`DRes<SInt> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol.right [private]`

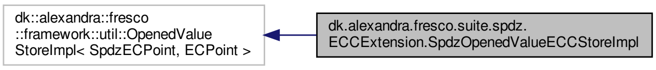
the shared Integer

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

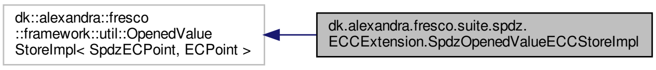
- SpdzMultECCProtocol.java

6.35 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOpenedValueECCStoreImpl Class Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOpenedValueECCStoreImpl:


dk::alexandra::fresco
::framework::util::OpenedValue
StoreImpl< SpdzECPoint, ECPoint >
dk.alexandra.fresco.suite.spdz.
ECCExtension.SpdzOpenedValueECCStoreImpl

Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOpenedValueECCStoreImpl:


dk::alexandra::fresco
::framework::util::OpenedValue
StoreImpl< SpdzECPoint, ECPoint >
dk.alexandra.fresco.suite.spdz.
ECCExtension.SpdzOpenedValueECCStoreImpl

6.35.1 Detailed Description

Spdz-specific instantiation of OpenedValueStore to store the Opened ECC Values.

Author

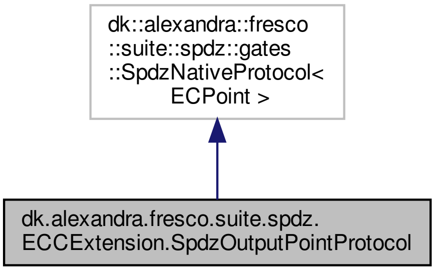
Roman Walch

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

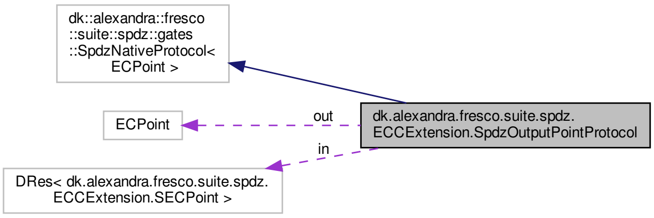
- SpdzOpenedValueECCStoreImpl.java

6.36 dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol Class Reference

Inheritance diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol:



Collaboration diagram for dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol:



Public Member Functions

- SpdzOutputPointProtocol (DRes< SECPoint > in)
- EvaluationStatus evaluate (int round, SpdzResourcePool spdzResourcePool, Network network)
- ECPoint out ()

Private Attributes

- DRes< SECPoint > in
- ECPoint out

6.36.1 Detailed Description

A Spdz protocol to open shared ECC points.

Author

Roman Walch

6.36.2 Constructor & Destructor Documentation

6.36.2.1 SpdzOutputPointProtocol()

```
dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol.SpdzOutputPointProtocol (
            DRes< SECPoint > in )
```

The constructor to initialize the computation.

Parameters

in	the shared ECC point

6.36.3 Member Function Documentation

6.36.3.1 evaluate()

```
EvaluationStatus dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol.evaluate
(
            int round,
            SpdzResourcePool spdzResourcePool,
            Network network )
```

This members opens the the shared ECC point. FRESCO calls this member when executing the SpdzOutpuPoint computation.

6.36.3.2 out()

```
ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol.out ( )
```

Helper class to output the resulting opened ECC point

6.36.4 Member Data Documentation

6.36.4.1 in

```
DRes<SECPoint> dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol.in  [private]
```

the shared ECC point

6.36.4.2 out

```
ECPoint dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol.out  [private]
```

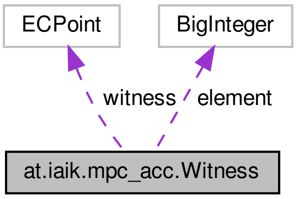
the opened public ECC point

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

- SpdzOutputPointProtocol.java

6.37 at.iaik.mpc_acc.Witness Class Reference

Collaboration diagram for at.iaik.mpc_acc.Witness:



Public Member Functions

- Witness (ECPoint witness, BigInteger element)
- ECPoint getWitness ()
- BigInteger getElement ()

Private Attributes

- ECPoint witness
- BigInteger element

6.37.1 Detailed Description

A simple class containing the Witness of the Accumulator

Author

Roman Walch

6.37.2 Constructor & Destructor Documentation

6.37.2.1 Witness()

```
at.iaik.mpc_acc.Witness.Witness (
             ECPoint witness,
             BigInteger element )
```

A constructor to initialize the Witness of an element

Parameters

witness	the witness
element	the corresponding element

6.37.3 Member Function Documentation

6.37.3.1 getElement()

```
BigInteger at.iaik.mpc_acc.Witness.getElement ( )
```

Getter for the corresponding element

Returns

the corresponding element

6.37.3.2 getWitness()

```
ECPoint at.iaik.mpc_acc.Witness.getWitness ( )
```

Getter for the witness

Returns

the witness

6.37.4 Member Data Documentation

6.37.4.1 element

```
BigInteger at.iaik.mpc_acc.Witness.element [private]
```

6.37.4.2 witness

```
ECPoint at.iaik.mpc_acc.Witness.witness [private]
```

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

- Witness.java

7 File Documentation

7.1 Accumulator.java File Reference

Classes

- class at.iaik.mpc_acc.Accumulator

Packages

- package at.iaik.mpc_acc

7.2 AccumulatorDemo.java File Reference

Classes

- class at.iaik.mpc_acc.AccumulatorDemo

Packages

- package at.iaik.mpc_acc

7.3 Auxillery.java File Reference

Classes

- class at.iaik.mpc_acc.Auxillery
- enum **at.iaik.mpc_acc.Auxillery.UPDATE**

Packages

- package at.iaik.mpc_acc

7.4 CmdLineParser.java File Reference

Classes

- class at.iaik.utils.CmdLineParser
- class at.iaik.utils.CmdLineParser.BuilderParams

Packages

- package at.iaik.utils

7.5 EvalResult.java File Reference

Classes

- class at.iaik.mpc_acc.EvalResult

Packages

- package at.iaik.mpc_acc

7.6 Main.java File Reference

Classes

- class at.iaik.mpc_acc.Main

Packages

- package at.iaik.mpc_acc

7.7 MerkleTree.java File Reference

Classes

- class at.iaik.mpc_acc.MerkleTree
- class at.iaik.mpc_acc.MerkleTree.ProofNode
- enum at.iaik.mpc_acc.MerkleTree.Position

Packages

- package at.iaik.mpc_acc

7.8 MPC_Acc.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_Acc

Packages

- package at.iaik.mpc_acc

7.9 MPC_Add.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_Add

Packages

- package at.iaik.mpc_acc

7.10 MPC_Del.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_Del

Packages

- package at.iaik.mpc_acc

7.11 MPC_Eval.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_Eval

Packages

- package at.iaik.mpc_acc

7.12 MPC_Gen.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_Gen

Packages

- package at.iaik.mpc_acc

7.13 MPC_TripleDummy.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_TripleDummy

Packages

- package at.iaik.mpc_acc

7.14 MPC_WitCreate.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_WitCreate

Packages

- package at.iaik.mpc_acc

7.15 MPC_WitUpdateAdd.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_WitUpdateAdd

Packages

- package at.iaik.mpc_acc

7.16 MPC_WitUpdateDel.java File Reference

Classes

- class at.iaik.mpc_acc.MPC_WitUpdateDel

Packages

- package at.iaik.mpc_acc

7.17 MPCBuilder.java File Reference

Classes

- class at.iaik.utils.MPCBuilder< ObjectT >

Packages

- package at.iaik.utils

7.18 MPCParams.java File Reference

Classes

- class at.iaik.mpc_acc.MPCParams

Packages

- package at.iaik.mpc_acc

7.19 MPCParamsBuilder.java File Reference

Classes

- class at.iaik.mpc_acc.MPCParamsBuilder

Packages

- package at.iaik.mpc_acc

7.20 NetworkLoggingDecorator.java File Reference

Classes

- class at.iaik.utils.NetworkLoggingDecorator
- class at.iaik.utils.NetworkLoggingDecorator.PartyStats

Packages

- package at.iaik.utils

7.21 NetworkManager.java File Reference

Classes

- class at.iaik.utils.NetworkManager

Packages

- package at.iaik.utils

7.22 Node.java File Reference

Classes

- class at.iaik.mpc_acc.Node

Packages

- package at.iaik.mpc_acc

7.23 Polynomial.java File Reference

Classes

- class at.iaik.mpc_acc.Polynomial

Packages

- package at.iaik.mpc_acc

7.24 SECPoint.java File Reference

Classes

- interface dk.alexandra.fresco.suite.spdz.ECCExtension.SECPoint

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.25 SpdzECCMacCheckProtocol.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCMacCheckProtocol

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.26 SpdzECCOps.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECCOps

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.27 SpdzECPoint.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzECPoint

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.28 SpdzKnownMultECCProtocol.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownMultECCProtocol

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.29 SpdzKnownScalar.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzKnownScalar

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.30 SpdzMultECCProtocol.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzMultECCProtocol

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.31 SpdzOpenedValueECCStoreImpl.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOpenedValueECCStoreImpl

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.32 SpdzOutputPointProtocol.java File Reference

Classes

- class dk.alexandra.fresco.suite.spdz.ECCExtension.SpdzOutputPointProtocol

Packages

- package dk.alexandra.fresco.suite.spdz.ECCExtension

7.33 Witness.java File Reference

Classes

- class at.iaik.mpc_acc.Witness

Packages

- package at.iaik.mpc_acc

Index