

SA104X

Generated by Doxygen 1.8.3.1

Mon Mar 25 2013 11:20:34



# Contents

|          |                                                  |          |
|----------|--------------------------------------------------|----------|
| <b>1</b> | <b>sa104x-kexjobb</b>                            | <b>1</b> |
| <b>2</b> | <b>Hierarchical Index</b>                        | <b>3</b> |
| 2.1      | Class Hierarchy . . . . .                        | 3        |
| <b>3</b> | <b>Class Index</b>                               | <b>5</b> |
| 3.1      | Class List . . . . .                             | 5        |
| <b>4</b> | <b>File Index</b>                                | <b>7</b> |
| 4.1      | File List . . . . .                              | 7        |
| <b>5</b> | <b>Class Documentation</b>                       | <b>9</b> |
| 5.1      | BaseLeaf Class Reference . . . . .               | 9        |
| 5.1.1    | Constructor & Destructor Documentation . . . . . | 9        |
| 5.1.1.1  | BaseLeaf . . . . .                               | 9        |
| 5.1.1.2  | ~BaseLeaf . . . . .                              | 9        |
| 5.2      | BaseNode Class Reference . . . . .               | 9        |
| 5.2.1    | Detailed Description . . . . .                   | 10       |
| 5.2.2    | Member Enumeration Documentation . . . . .       | 10       |
| 5.2.2.1  | NodeType . . . . .                               | 10       |
| 5.2.3    | Constructor & Destructor Documentation . . . . . | 10       |
| 5.2.3.1  | BaseNode . . . . .                               | 10       |
| 5.2.3.2  | ~BaseNode . . . . .                              | 10       |
| 5.2.4    | Member Function Documentation . . . . .          | 10       |
| 5.2.4.1  | concatData . . . . .                             | 10       |
| 5.2.4.2  | copy . . . . .                                   | 11       |
| 5.2.4.3  | getLength . . . . .                              | 11       |
| 5.2.4.4  | getType . . . . .                                | 11       |
| 5.2.4.5  | ReadNodeHeader . . . . .                         | 11       |
| 5.2.4.6  | serialize . . . . .                              | 11       |
| 5.2.4.7  | toVector . . . . .                               | 11       |
| 5.3      | DataLeaf Class Reference . . . . .               | 11       |
| 5.3.1    | Constructor & Destructor Documentation . . . . . | 11       |

|          |                                        |    |
|----------|----------------------------------------|----|
| 5.3.1.1  | DataLeaf                               | 12 |
| 5.3.1.2  | DataLeaf                               | 12 |
| 5.3.1.3  | DataLeaf                               | 12 |
| 5.3.1.4  | DataLeaf                               | 12 |
| 5.3.1.5  | ~DataLeaf                              | 12 |
| 5.3.2    | Member Function Documentation          | 12 |
| 5.3.2.1  | getData                                | 12 |
| 5.3.2.2  | getData                                | 12 |
| 5.3.2.3  | getLength                              | 12 |
| 5.3.2.4  | operator=                              | 12 |
| 5.3.2.5  | toVector                               | 12 |
| 5.4      | IntLeaf Class Reference                | 12 |
| 5.4.1    | Constructor & Destructor Documentation | 13 |
| 5.4.1.1  | IntLeaf                                | 13 |
| 5.4.1.2  | IntLeaf                                | 13 |
| 5.4.1.3  | IntLeaf                                | 14 |
| 5.4.1.4  | IntLeaf                                | 14 |
| 5.4.1.5  | IntLeaf                                | 14 |
| 5.4.1.6  | IntLeaf                                | 14 |
| 5.4.1.7  | IntLeaf                                | 14 |
| 5.4.1.8  | IntLeaf                                | 14 |
| 5.4.1.9  | ~IntLeaf                               | 14 |
| 5.4.2    | Member Function Documentation          | 14 |
| 5.4.2.1  | add                                    | 14 |
| 5.4.2.2  | addMod                                 | 14 |
| 5.4.2.3  | addTo                                  | 14 |
| 5.4.2.4  | addToMod                               | 14 |
| 5.4.2.5  | constructPartFromFile                  | 14 |
| 5.4.2.6  | exp                                    | 14 |
| 5.4.2.7  | expMod                                 | 14 |
| 5.4.2.8  | expTo                                  | 14 |
| 5.4.2.9  | expToMod                               | 14 |
| 5.4.2.10 | getBigInt                              | 14 |
| 5.4.2.11 | getLength                              | 14 |
| 5.4.2.12 | inverse                                | 14 |
| 5.4.2.13 | mod                                    | 14 |
| 5.4.2.14 | modTo                                  | 14 |
| 5.4.2.15 | mult                                   | 14 |
| 5.4.2.16 | multMod                                | 14 |
| 5.4.2.17 | multTo                                 | 14 |

|          |                                        |    |
|----------|----------------------------------------|----|
| 5.4.2.18 | multToMod                              | 14 |
| 5.4.2.19 | operator!=                             | 14 |
| 5.4.2.20 | operator*                              | 15 |
| 5.4.2.21 | operator*==                            | 15 |
| 5.4.2.22 | operator+                              | 15 |
| 5.4.2.23 | operator+=                             | 15 |
| 5.4.2.24 | operator-                              | 15 |
| 5.4.2.25 | operator<                              | 15 |
| 5.4.2.26 | operator=                              | 15 |
| 5.4.2.27 | operator==                             | 15 |
| 5.4.2.28 | operator=                              | 15 |
| 5.4.2.29 | operator==                             | 15 |
| 5.4.2.30 | operator>                              | 15 |
| 5.4.2.31 | toString                               | 15 |
| 5.4.2.32 | toVector                               | 15 |
| 5.4.3    | Member Data Documentation              | 15 |
| 5.4.3.1  | ARRAYORDER                             | 15 |
| 5.4.3.2  | ENDIAN                                 | 15 |
| 5.4.3.3  | NAILS                                  | 15 |
| 5.5      | Node Class Reference                   | 15 |
| 5.5.1    | Constructor & Destructor Documentation | 17 |
| 5.5.1.1  | Node                                   | 17 |
| 5.5.1.2  | Node                                   | 17 |
| 5.5.1.3  | Node                                   | 17 |
| 5.5.1.4  | Node                                   | 17 |
| 5.5.1.5  | Node                                   | 17 |
| 5.5.1.6  | ~Node                                  | 17 |
| 5.5.2    | Member Function Documentation          | 17 |
| 5.5.2.1  | add                                    | 17 |
| 5.5.2.2  | add                                    | 17 |
| 5.5.2.3  | addChild                               | 17 |
| 5.5.2.4  | addMod                                 | 17 |
| 5.5.2.5  | addMod                                 | 17 |
| 5.5.2.6  | addTo                                  | 17 |
| 5.5.2.7  | addTo                                  | 17 |
| 5.5.2.8  | addToMod                               | 17 |
| 5.5.2.9  | addToMod                               | 17 |
| 5.5.2.10 | constructPartFromFile                  | 17 |
| 5.5.2.11 | exp                                    | 17 |
| 5.5.2.12 | exp                                    | 17 |

|          |                 |    |
|----------|-----------------|----|
| 5.5.2.13 | expMod          | 18 |
| 5.5.2.14 | expMod          | 18 |
| 5.5.2.15 | expMod          | 18 |
| 5.5.2.16 | expMult         | 18 |
| 5.5.2.17 | expMultMod      | 18 |
| 5.5.2.18 | expMultMod      | 18 |
| 5.5.2.19 | expTo           | 18 |
| 5.5.2.20 | expTo           | 18 |
| 5.5.2.21 | expToMod        | 18 |
| 5.5.2.22 | expToMod        | 18 |
| 5.5.2.23 | expToMod        | 18 |
| 5.5.2.24 | getChildren     | 18 |
| 5.5.2.25 | getIntLeafChild | 18 |
| 5.5.2.26 | getIntLeafChild | 18 |
| 5.5.2.27 | getLength       | 18 |
| 5.5.2.28 | getNodeChild    | 18 |
| 5.5.2.29 | getNodeChild    | 18 |
| 5.5.2.30 | mod             | 18 |
| 5.5.2.31 | modTo           | 18 |
| 5.5.2.32 | mult            | 18 |
| 5.5.2.33 | mult            | 18 |
| 5.5.2.34 | multMod         | 18 |
| 5.5.2.35 | multMod         | 18 |
| 5.5.2.36 | multTo          | 18 |
| 5.5.2.37 | multTo          | 18 |
| 5.5.2.38 | multToMod       | 18 |
| 5.5.2.39 | multToMod       | 18 |
| 5.5.2.40 | operator!=      | 19 |
| 5.5.2.41 | operator*       | 19 |
| 5.5.2.42 | operator*       | 19 |
| 5.5.2.43 | operator*=      | 19 |
| 5.5.2.44 | operator*=      | 19 |
| 5.5.2.45 | operator+       | 19 |
| 5.5.2.46 | operator+       | 19 |
| 5.5.2.47 | operator+=      | 19 |
| 5.5.2.48 | operator+=      | 19 |
| 5.5.2.49 | operator=       | 19 |
| 5.5.2.50 | operator==      | 19 |
| 5.5.2.51 | prod            | 19 |
| 5.5.2.52 | prodMod         | 19 |

|          |                                        |           |
|----------|----------------------------------------|-----------|
| 5.5.2.53 | sum                                    | 19        |
| 5.5.2.54 | sumMod                                 | 19        |
| 5.5.2.55 | toString                               | 19        |
| 5.5.2.56 | toVector                               | 19        |
| 5.6      | PRG Class Reference                    | 19        |
| 5.6.1    | Constructor & Destructor Documentation | 19        |
| 5.6.1.1  | PRG                                    | 20        |
| 5.6.1.2  | ~PRG                                   | 20        |
| 5.6.2    | Member Function Documentation          | 20        |
| 5.6.2.1  | next                                   | 20        |
| 5.7      | proofStruct Struct Reference           | 20        |
| 5.7.1    | Member Data Documentation              | 20        |
| 5.7.1.1  | Gq                                     | 20        |
| 5.7.1.2  | hash                                   | 20        |
| 5.7.1.3  | lambda                                 | 20        |
| 5.7.1.4  | N                                      | 20        |
| 5.7.1.5  | nE                                     | 20        |
| 5.7.1.6  | nHash                                  | 20        |
| 5.7.1.7  | nR                                     | 20        |
| 5.7.1.8  | nV                                     | 20        |
| 5.7.1.9  | pk                                     | 21        |
| 5.7.1.10 | rho                                    | 21        |
| 5.7.1.11 | Rw                                     | 21        |
| 5.7.1.12 | width                                  | 21        |
| 5.7.1.13 | x                                      | 21        |
| 5.7.1.14 | y                                      | 21        |
| 5.8      | RO Class Reference                     | 21        |
| 5.8.1    | Constructor & Destructor Documentation | 21        |
| 5.8.1.1  | RO                                     | 21        |
| 5.8.1.2  | ~RO                                    | 21        |
| 5.8.2    | Member Function Documentation          | 21        |
| 5.8.2.1  | operator()                             | 21        |
| <b>6</b> | <b>File Documentation</b>              | <b>23</b> |
| 6.1      | Arithmetic/BaseLeaf.cpp File Reference | 23        |
| 6.2      | Arithmetic/BaseLeaf.h File Reference   | 23        |
| 6.3      | Arithmetic/BaseNode.cpp File Reference | 23        |
| 6.4      | Arithmetic/BaseNode.h File Reference   | 23        |
| 6.5      | Arithmetic/DataLeaf.cpp File Reference | 24        |
| 6.5.1    | Macro Definition Documentation         | 24        |

|          |                                       |    |
|----------|---------------------------------------|----|
| 6.5.1.1  | ARRAYORDER                            | 24 |
| 6.6      | Arithmetic/DataLeaf.h File Reference  | 24 |
| 6.7      | Arithmetic/IntLeaf.cpp File Reference | 24 |
| 6.8      | Arithmetic/IntLeaf.h File Reference   | 24 |
| 6.9      | Arithmetic/Node.cpp File Reference    | 25 |
| 6.10     | Arithmetic/Node.h File Reference      | 25 |
| 6.11     | Arithmetic/types.h File Reference     | 25 |
| 6.11.1   | Typedef Documentation                 | 25 |
| 6.11.1.1 | bytevector                            | 25 |
| 6.12     | Crypto/EIGamal.cpp File Reference     | 25 |
| 6.12.1   | Function Documentation                | 26 |
| 6.12.1.1 | Enc                                   | 26 |
| 6.12.1.2 | PDec                                  | 26 |
| 6.12.1.3 | TDec                                  | 26 |
| 6.13     | Crypto/EIGamal.h File Reference       | 26 |
| 6.13.1   | Function Documentation                | 26 |
| 6.13.1.1 | Enc                                   | 26 |
| 6.13.1.2 | PDec                                  | 26 |
| 6.13.1.3 | TDec                                  | 26 |
| 6.14     | Crypto/H_SHA.cpp File Reference       | 26 |
| 6.14.1   | Function Documentation                | 26 |
| 6.14.1.1 | H_SHA                                 | 26 |
| 6.14.1.2 | H_SHA256                              | 26 |
| 6.14.1.3 | H_SHA384                              | 26 |
| 6.14.1.4 | H_SHA512                              | 27 |
| 6.15     | Crypto/H_SHA.h File Reference         | 27 |
| 6.15.1   | Function Documentation                | 27 |
| 6.15.1.1 | H_SHA                                 | 27 |
| 6.15.1.2 | H_SHA256                              | 27 |
| 6.15.1.3 | H_SHA384                              | 27 |
| 6.15.1.4 | H_SHA512                              | 27 |
| 6.16     | Crypto/PRG.cpp File Reference         | 27 |
| 6.17     | Crypto/PRG.h File Reference           | 27 |
| 6.18     | Crypto/RandomArray.cpp File Reference | 27 |
| 6.18.1   | Function Documentation                | 28 |
| 6.18.1.1 | RandomArray                           | 28 |
| 6.19     | Crypto/RandomArray.h File Reference   | 28 |
| 6.19.1   | Function Documentation                | 28 |
| 6.19.1.1 | RandomArray                           | 28 |
| 6.20     | Crypto/RO.cpp File Reference          | 28 |



|          |                                                       |    |
|----------|-------------------------------------------------------|----|
| 6.21     | Crypto/RO.h File Reference                            | 28 |
| 6.22     | README.md File Reference                              | 28 |
| 6.23     | Tests/ByteTreeTests.cpp File Reference                | 28 |
| 6.23.1   | Function Documentation                                | 29 |
| 6.23.1.1 | TEST                                                  | 29 |
| 6.23.1.2 | TEST                                                  | 29 |
| 6.23.1.3 | TEST                                                  | 29 |
| 6.24     | Tests/IntLeafArithmeticsTests.cpp File Reference      | 29 |
| 6.24.1   | Function Documentation                                | 29 |
| 6.24.1.1 | TEST                                                  | 29 |
| 6.24.1.2 | TEST                                                  | 29 |
| 6.24.1.3 | TEST                                                  | 29 |
| 6.24.1.4 | TEST                                                  | 29 |
| 6.25     | Tests/NodeArithmeticsTests.cpp File Reference         | 29 |
| 6.25.1   | Function Documentation                                | 30 |
| 6.25.1.1 | TEST                                                  | 30 |
| 6.25.1.2 | TEST                                                  | 30 |
| 6.25.1.3 | TEST                                                  | 30 |
| 6.25.1.4 | TEST                                                  | 30 |
| 6.26     | Tests/NodeDataInitTests.cpp File Reference            | 30 |
| 6.26.1   | Function Documentation                                | 30 |
| 6.26.1.1 | TEST                                                  | 30 |
| 6.26.1.2 | TEST                                                  | 30 |
| 6.26.1.3 | TEST                                                  | 30 |
| 6.27     | Tests/NodeToStringTests.cpp File Reference            | 30 |
| 6.27.1   | Function Documentation                                | 30 |
| 6.27.1.1 | TEST                                                  | 30 |
| 6.27.1.2 | TEST                                                  | 30 |
| 6.27.1.3 | TEST                                                  | 30 |
| 6.28     | Tests/PRGTests.cpp File Reference                     | 31 |
| 6.28.1   | Function Documentation                                | 31 |
| 6.28.1.1 | TEST                                                  | 31 |
| 6.28.1.2 | TEST                                                  | 31 |
| 6.28.1.3 | TEST                                                  | 31 |
| 6.29     | Tests/TestRunner.cpp File Reference                   | 31 |
| 6.29.1   | Function Documentation                                | 31 |
| 6.29.1.1 | main                                                  | 31 |
| 6.30     | Verifier/DecryptionFactorsVerifier.cpp File Reference | 31 |
| 6.30.1   | Function Documentation                                | 32 |
| 6.30.1.1 | DecryptionFactorsVerifier                             | 32 |

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 6.31 Verifier/DecryptionFactorsVerifier.h File Reference . . . . . | 32 |
| 6.31.1 Function Documentation . . . . .                            | 32 |
| 6.31.1.1 DecryptionFactorsVerifier . . . . .                       | 32 |
| 6.32 Verifier/DecryptionVerifier.cpp File Reference . . . . .      | 32 |
| 6.32.1 Function Documentation . . . . .                            | 32 |
| 6.32.1.1 DecryptionVerifier . . . . .                              | 32 |
| 6.33 Verifier/DecryptionVerifier.h File Reference . . . . .        | 32 |
| 6.33.1 Function Documentation . . . . .                            | 33 |
| 6.33.1.1 DecryptionVerifier . . . . .                              | 33 |
| 6.34 Verifier/FileNames.h File Reference . . . . .                 | 33 |
| 6.34.1 Variable Documentation . . . . .                            | 33 |
| 6.34.1.1 CIPHERTEXTS_FILE . . . . .                                | 33 |
| 6.34.1.2 CIPHERTEXTS_FILE_PREFIX . . . . .                         | 33 |
| 6.34.1.3 FILE_SUFFIX . . . . .                                     | 33 |
| 6.34.1.4 FULL_PUBLIC_KEY_FILE . . . . .                            | 33 |
| 6.34.1.5 MAXCIPH_FILE . . . . .                                    | 33 |
| 6.34.1.6 PARTIAL_PUBLIC_KEY_FILE_PREFIX . . . . .                  | 33 |
| 6.34.1.7 PARTIAL_SECRET_KEY_FILE_PREFIX . . . . .                  | 33 |
| 6.34.1.8 PLAINTEXTS_FILE . . . . .                                 | 33 |
| 6.34.1.9 SHUFFLED_CIPHERTEXTS_FILE . . . . .                       | 33 |
| 6.35 Verifier/KeyVerifier.cpp File Reference . . . . .             | 33 |
| 6.35.1 Function Documentation . . . . .                            | 34 |
| 6.35.1.1 isPartialPublicKey . . . . .                              | 34 |
| 6.35.1.2 isPartialSecretKey . . . . .                              | 34 |
| 6.35.1.3 isPublicKey . . . . .                                     | 34 |
| 6.35.1.4 keyVerifier . . . . .                                     | 34 |
| 6.36 Verifier/KeyVerifier.h File Reference . . . . .               | 34 |
| 6.36.1 Function Documentation . . . . .                            | 34 |
| 6.36.1.1 isPartialPublicKey . . . . .                              | 34 |
| 6.36.1.2 isPartialSecretKey . . . . .                              | 34 |
| 6.36.1.3 isPublicKey . . . . .                                     | 34 |
| 6.36.1.4 keyVerifier . . . . .                                     | 34 |
| 6.37 Verifier/main.cpp File Reference . . . . .                    | 34 |
| 6.37.1 Function Documentation . . . . .                            | 35 |
| 6.37.1.1 main . . . . .                                            | 35 |
| 6.37.1.2 ParseProtinfoDirectory . . . . .                          | 35 |
| 6.37.1.3 SetMode . . . . .                                         | 35 |
| 6.38 Verifier/ProofOfShuffle.cpp File Reference . . . . .          | 35 |
| 6.38.1 Function Documentation . . . . .                            | 35 |
| 6.38.1.1 proofOfShuffle . . . . .                                  | 35 |

|                                                              |    |
|--------------------------------------------------------------|----|
| 6.39 Verifier/ProofOfShuffle.h File Reference . . . . .      | 35 |
| 6.39.1 Function Documentation . . . . .                      | 35 |
| 6.39.1.1 proofOfShuffle . . . . .                            | 35 |
| 6.40 Verifier/ShufflingVerifier.cpp File Reference . . . . . | 36 |
| 6.40.1 Function Documentation . . . . .                      | 36 |
| 6.40.1.1 isListOfCiphertexts . . . . .                       | 36 |
| 6.40.1.2 verifyShuffling . . . . .                           | 36 |
| 6.41 Verifier/ShufflingVerifier.h File Reference . . . . .   | 36 |
| 6.41.1 Function Documentation . . . . .                      | 36 |
| 6.41.1.1 isListOfCiphertexts . . . . .                       | 36 |
| 6.41.1.2 verifyShuffling . . . . .                           | 36 |
| 6.42 Verifier/Utilities.cpp File Reference . . . . .         | 36 |
| 6.42.1 Function Documentation . . . . .                      | 37 |
| 6.42.1.1 getGroupFromString . . . . .                        | 37 |
| 6.42.1.2 isElemOfCw . . . . .                                | 37 |
| 6.42.1.3 isElemOfGq . . . . .                                | 37 |
| 6.42.1.4 isElemOfMw . . . . .                                | 37 |
| 6.42.1.5 isElemOfZn . . . . .                                | 37 |
| 6.42.1.6 isPedersenCommitment . . . . .                      | 37 |
| 6.43 Verifier/Utilities.h File Reference . . . . .           | 37 |
| 6.43.1 Function Documentation . . . . .                      | 37 |
| 6.43.1.1 getGroupFromString . . . . .                        | 37 |
| 6.43.1.2 isElemOfCw . . . . .                                | 37 |
| 6.43.1.3 isElemOfGq . . . . .                                | 37 |
| 6.43.1.4 isElemOfMw . . . . .                                | 38 |
| 6.43.1.5 isElemOfZn . . . . .                                | 38 |
| 6.43.1.6 isPedersenCommitment . . . . .                      | 38 |
| 6.43.2 Variable Documentation . . . . .                      | 38 |
| 6.43.2.1 BOTTOM . . . . .                                    | 38 |
| 6.44 Verifier/Verifier.cpp File Reference . . . . .          | 38 |
| 6.44.1 Function Documentation . . . . .                      | 38 |
| 6.44.1.1 Verifier . . . . .                                  | 38 |
| 6.45 Verifier/Verifier.h File Reference . . . . .            | 38 |
| 6.45.1 Enumeration Type Documentation . . . . .              | 39 |
| 6.45.1.1 RunMode . . . . .                                   | 39 |
| 6.45.2 Function Documentation . . . . .                      | 39 |
| 6.45.2.1 Verifier . . . . .                                  | 39 |
| 6.45.3 Variable Documentation . . . . .                      | 39 |
| 6.45.3.1 CIPHERTEXT_FILE_PREFIX . . . . .                    | 39 |

**Index****39**

# Chapter 1

## sa104x-kexjobb

Implementation of a verifier for the Verificatum Mix-net

### Requirements

- OpenSSL
- GMP
- RapidXML



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

|                       |    |
|-----------------------|----|
| BaseNode . . . . .    | 9  |
| BaseLeaf . . . . .    | 9  |
| DataLeaf . . . . .    | 11 |
| IntLeaf . . . . .     | 12 |
| Node . . . . .        | 15 |
| PRG . . . . .         | 19 |
| proofStruct . . . . . | 20 |
| RO . . . . .          | 21 |





## Chapter 3

# Class Index

### 3.1 Class List

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

|                             |    |
|-----------------------------|----|
| <a href="#">BaseLeaf</a>    | 9  |
| <a href="#">BaseNode</a>    | 9  |
| <a href="#">DataLeaf</a>    | 11 |
| <a href="#">IntLeaf</a>     | 12 |
| <a href="#">Node</a>        | 15 |
| <a href="#">PRG</a>         | 19 |
| <a href="#">proofStruct</a> | 20 |
| <a href="#">RO</a>          | 21 |



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

|                                        |    |
|----------------------------------------|----|
| Arithmetic/BaseLeaf.cpp                | 23 |
| Arithmetic/BaseLeaf.h                  | 23 |
| Arithmetic/BaseNode.cpp                | 23 |
| Arithmetic/BaseNode.h                  | 23 |
| Arithmetic/DataLeaf.cpp                | 24 |
| Arithmetic/DataLeaf.h                  | 24 |
| Arithmetic/IntLeaf.cpp                 | 24 |
| Arithmetic/IntLeaf.h                   | 24 |
| Arithmetic/Node.cpp                    | 25 |
| Arithmetic/Node.h                      | 25 |
| Arithmetic/types.h                     | 25 |
| Crypto/ElGamal.cpp                     | 25 |
| Crypto/ElGamal.h                       | 26 |
| Crypto/H_SHA.cpp                       | 26 |
| Crypto/H_SHA.h                         | 27 |
| Crypto/PRG.cpp                         | 27 |
| Crypto/PRG.h                           | 27 |
| Crypto/RandomArray.cpp                 | 27 |
| Crypto/RandomArray.h                   | 28 |
| Crypto/RO.cpp                          | 28 |
| Crypto/RO.h                            | 28 |
| Tests/ByteTreeTests.cpp                | 28 |
| Tests/IntLeafArithmeticsTests.cpp      | 29 |
| Tests/NodeArithmeticsTests.cpp         | 29 |
| Tests/NodeDataInitTests.cpp            | 30 |
| Tests/NodeToStringTests.cpp            | 30 |
| Tests/PRGTests.cpp                     | 31 |
| Tests/TestRunner.cpp                   | 31 |
| Verifier/DecryptionFactorsVerifier.cpp | 31 |
| Verifier/DecryptionFactorsVerifier.h   | 32 |
| Verifier/DecryptionVerifier.cpp        | 32 |
| Verifier/DecryptionVerifier.h          | 32 |
| Verifier/FileNames.h                   | 33 |
| Verifier/KeyVerifier.cpp               | 33 |
| Verifier/KeyVerifier.h                 | 34 |
| Verifier/main.cpp                      | 34 |
| Verifier/ProofOfShuffle.cpp            | 35 |
| Verifier/ProofOfShuffle.h              | 35 |

|                                          |    |
|------------------------------------------|----|
| Verifier/ShufflingVerifier.cpp . . . . . | 36 |
| Verifier/ShufflingVerifier.h . . . . .   | 36 |
| Verifier/Utilities.cpp . . . . .         | 36 |
| Verifier/Utilities.h . . . . .           | 37 |
| Verifier/Verifier.cpp . . . . .          | 38 |
| Verifier/Verifier.h . . . . .            | 38 |

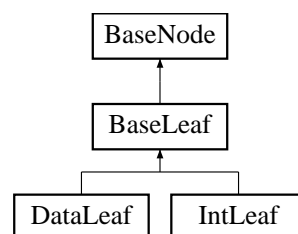
## Chapter 5

# Class Documentation

### 5.1 BaseLeaf Class Reference

```
#include <BaseLeaf.h>
```

Inheritance diagram for BaseLeaf:



#### Public Member Functions

- [BaseLeaf](#) ([BaseLeaf::NodeType](#) type)
- [~BaseLeaf](#) (void)

#### Additional Inherited Members

##### 5.1.1 Constructor & Destructor Documentation

5.1.1.1 [BaseLeaf::BaseLeaf](#) ( [BaseLeaf::NodeType](#) type ) [explicit]

5.1.1.2 [BaseLeaf::~~BaseLeaf](#) ( void )

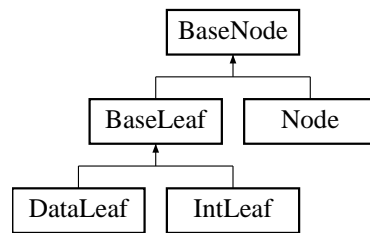
The documentation for this class was generated from the following files:

- Arithmetic/[BaseLeaf.h](#)
- Arithmetic/[BaseLeaf.cpp](#)

### 5.2 BaseNode Class Reference

```
#include <BaseNode.h>
```

Inheritance diagram for BaseNode:



## Public Types

- enum `NodeType` { `NODE` = 0, `INT_LEAF` = 1, `DATA_LEAF` = 2 }

## Public Member Functions

- `BaseNode` (`BaseNode::NodeType` type)
- `~BaseNode` (void)
- `BaseNode::NodeType` `getType` (void) const
- virtual `int32_t` `getLength` (void) const =0
- virtual `bytevector` `toVector` (void) const =0
- `bytevector` `serialize` () const
- `bytevector` `concatData` (const `BaseNode` \*const other) const

## Static Protected Member Functions

- static `BaseNode` \* `copy` (const `BaseNode` \*node)
- static void `ReadNodeHeader` (std::istream &file, char &type, uint32\_t &length)

### 5.2.1 Detailed Description

The basic node from which all nodes in a Byte Tree inherit.

### 5.2.2 Member Enumeration Documentation

#### 5.2.2.1 enum `BaseNode::NodeType`

Enumerator

**`NODE`** A node which contains other nodes.

**`INT_LEAF`** An `IntLeaf` which contains a number.

**`DATA_LEAF`** A `DataLeaf` which contains a string or vector of bytes.

### 5.2.3 Constructor & Destructor Documentation

#### 5.2.3.1 `BaseNode::BaseNode` ( `BaseNode::NodeType` type )

#### 5.2.3.2 `BaseNode::~~BaseNode` ( void )

### 5.2.4 Member Function Documentation

#### 5.2.4.1 `bytevector` `BaseNode::concatData` ( const `BaseNode` \*const other ) const

5.2.4.2 **BaseNode \* BaseNode::copy ( const BaseNode \* node )** [static], [protected]

5.2.4.3 **virtual int32\_t BaseNode::getLength ( void ) const** [pure virtual]

Implemented in [Node](#), [IntLeaf](#), and [DataLeaf](#).

5.2.4.4 **BaseNode::NodeType BaseNode::getType ( void ) const**

5.2.4.5 **void BaseNode::ReadNodeHeader ( std::istream & file, char & type, uint32\_t & length )** [static], [protected]

5.2.4.6 **bytevector BaseNode::serialize ( ) const**

5.2.4.7 **virtual bytevector BaseNode::toVector ( void ) const** [pure virtual]

Implemented in [IntLeaf](#), [Node](#), and [DataLeaf](#).

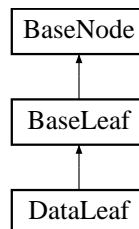
The documentation for this class was generated from the following files:

- [Arithmetic/BaseNode.h](#)
- [Arithmetic/BaseNode.cpp](#)

## 5.3 DataLeaf Class Reference

```
#include <DataLeaf.h>
```

Inheritance diagram for DataLeaf:



### Public Member Functions

- [DataLeaf](#) (void)
- [DataLeaf](#) (int32\_t size)
- [DataLeaf](#) (std::istream &file)
- [DataLeaf](#) (std::string str)
- [~DataLeaf](#) (void)
- [bytevector](#) & [getData](#) (void)
- const [bytevector](#) & [getData](#) (void) const
- virtual [bytevector](#) [toVector](#) (void) const
- virtual int32\_t [getLength](#) (void) const
- [DataLeaf](#) & [operator=](#) (const [DataLeaf](#) &leaf)

### Additional Inherited Members

#### 5.3.1 Constructor & Destructor Documentation

5.3.1.1 `DataLeaf::DataLeaf ( void )`

5.3.1.2 `DataLeaf::DataLeaf ( int32_t size ) [explicit]`

5.3.1.3 `DataLeaf::DataLeaf ( std::istream & file )`

5.3.1.4 `DataLeaf::DataLeaf ( std::string str )`

5.3.1.5 `DataLeaf::~DataLeaf ( void )`

## 5.3.2 Member Function Documentation

5.3.2.1 `bytevector & DataLeaf::getData ( void )`

5.3.2.2 `const bytevector & DataLeaf::getData ( void ) const`

5.3.2.3 `int32_t DataLeaf::getLength ( void ) const [virtual]`

Implements [BaseNode](#).

5.3.2.4 `DataLeaf & DataLeaf::operator= ( const DataLeaf & leaf )`

5.3.2.5 `bytevector DataLeaf::toVector ( void ) const [virtual]`

Implements [BaseNode](#).

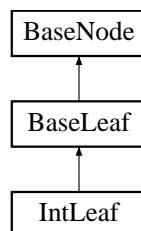
The documentation for this class was generated from the following files:

- Arithmetic/[DataLeaf.h](#)
- Arithmetic/[DataLeaf.cpp](#)

## 5.4 IntLeaf Class Reference

```
#include <IntLeaf.h>
```

Inheritance diagram for IntLeaf:



### Public Member Functions

- [IntLeaf](#) (void)
- [IntLeaf](#) (const [IntLeaf](#) &leaf)
- [IntLeaf](#) (const mpz\_class &bigint)
- [IntLeaf](#) (long int input)
- [IntLeaf](#) (long int input, long int length)
- [IntLeaf](#) (std::string input)
- [IntLeaf](#) ([bytevector](#) bytevec)



- [IntLeaf](#) (std::istream &file)
- [~IntLeaf](#) (void)
- [IntLeaf & operator=](#) (const [IntLeaf](#) &leaf)
- [IntLeaf & operator=](#) (long int input)
- [IntLeaf & operator=](#) (std::string input)
- [IntLeaf & modTo](#) (const [IntLeaf](#) &leaf)
- [IntLeaf mod](#) (const [IntLeaf](#) &leaf) const
- [IntLeaf & addTo](#) (const [IntLeaf](#) &leaf)
- [IntLeaf add](#) (const [IntLeaf](#) &leaf) const
- [IntLeaf & addToMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod)
- [IntLeaf addMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod) const
- [IntLeaf & operator+=](#) (const [IntLeaf](#) &leaf)
- [IntLeaf operator+](#) (const [IntLeaf](#) &leaf) const
- [IntLeaf & multTo](#) (const [IntLeaf](#) &leaf)
- [IntLeaf mult](#) (const [IntLeaf](#) &leaf) const
- [IntLeaf & multToMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod)
- [IntLeaf multMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod) const
- [IntLeaf & operator\\*=](#) (const [IntLeaf](#) &leaf)
- [IntLeaf operator\\*](#) (const [IntLeaf](#) &leaf) const
- [IntLeaf & expTo](#) (unsigned long exponent)
- [IntLeaf exp](#) (unsigned long exponent) const
- [IntLeaf & expToMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod)
- [IntLeaf expMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod) const
- bool [operator==](#) (const [IntLeaf](#) &leaf) const
- bool [operator!=](#) (const [IntLeaf](#) &leaf) const
- bool [operator<](#) (const [IntLeaf](#) &leaf) const
- bool [operator>](#) (const [IntLeaf](#) &leaf) const
- [IntLeaf operator-](#) (void) const
- [IntLeaf inverse](#) (const [IntLeaf](#) &mod) const
- mpz\_class [getBigInt](#) (void) const
- virtual [bytevector toVector](#) (void) const
- virtual int32\_t [getLength](#) (void) const
- std::string [toString](#) (void) const

### Static Public Member Functions

- static [BaseNode](#) \* [constructPartFromFile](#) (std::istream &file, uint32\_t length)

### Static Public Attributes

- static const int [ARRAYORDER](#) = 1
- static const int [ENDIAN](#) = 0
- static const int [NAILS](#) = 0

### Additional Inherited Members

#### 5.4.1 Constructor & Destructor Documentation

5.4.1.1 [IntLeaf::IntLeaf \( void \)](#)

5.4.1.2 [IntLeaf::IntLeaf \( const IntLeaf & leaf \)](#)

5.4.1.3 `IntLeaf::IntLeaf ( const mpz_class & bigint )` `[explicit]`

5.4.1.4 `IntLeaf::IntLeaf ( long int input )`

5.4.1.5 `IntLeaf::IntLeaf ( long int input, long int length )`

5.4.1.6 `IntLeaf::IntLeaf ( std::string input )` `[explicit]`

5.4.1.7 `IntLeaf::IntLeaf ( bytevector bytevec )` `[explicit]`

5.4.1.8 `IntLeaf::IntLeaf ( std::istream & file )` `[explicit]`

5.4.1.9 `IntLeaf::~IntLeaf ( void )`

## 5.4.2 Member Function Documentation

5.4.2.1 `IntLeaf IntLeaf::add ( const IntLeaf & leaf ) const`

5.4.2.2 `IntLeaf IntLeaf::addMod ( const IntLeaf & leaf, const IntLeaf & mod ) const`

5.4.2.3 `IntLeaf & IntLeaf::addTo ( const IntLeaf & leaf )`

5.4.2.4 `IntLeaf & IntLeaf::addToMod ( const IntLeaf & leaf, const IntLeaf & mod )`

5.4.2.5 `BaseNode * IntLeaf::constructPartFromFile ( std::istream & file, uint32_t length )` `[static]`

5.4.2.6 `IntLeaf IntLeaf::exp ( unsigned long exponent ) const`

5.4.2.7 `IntLeaf IntLeaf::expMod ( const IntLeaf & leaf, const IntLeaf & mod ) const`

5.4.2.8 `IntLeaf & IntLeaf::expTo ( unsigned long exponent )`

5.4.2.9 `IntLeaf & IntLeaf::expToMod ( const IntLeaf & leaf, const IntLeaf & mod )`

5.4.2.10 `mpz_class IntLeaf::getBigInt ( void ) const`

5.4.2.11 `int32_t IntLeaf::getLength ( void ) const` `[virtual]`

Implements [BaseNode](#).

5.4.2.12 `IntLeaf IntLeaf::inverse ( const IntLeaf & mod ) const`

5.4.2.13 `IntLeaf IntLeaf::mod ( const IntLeaf & leaf ) const`

5.4.2.14 `IntLeaf & IntLeaf::modTo ( const IntLeaf & leaf )`

5.4.2.15 `IntLeaf IntLeaf::mult ( const IntLeaf & leaf ) const`

5.4.2.16 `IntLeaf IntLeaf::multMod ( const IntLeaf & leaf, const IntLeaf & mod ) const`

5.4.2.17 `IntLeaf & IntLeaf::multTo ( const IntLeaf & leaf )`

5.4.2.18 `IntLeaf & IntLeaf::multToMod ( const IntLeaf & leaf, const IntLeaf & mod )`

5.4.2.19 `bool IntLeaf::operator!= ( const IntLeaf & leaf ) const`

5.4.2.20 `IntLeaf IntLeaf::operator* ( const IntLeaf & leaf ) const`

5.4.2.21 `IntLeaf & IntLeaf::operator*= ( const IntLeaf & leaf )`

5.4.2.22 `IntLeaf IntLeaf::operator+ ( const IntLeaf & leaf ) const`

5.4.2.23 `IntLeaf & IntLeaf::operator+= ( const IntLeaf & leaf )`

5.4.2.24 `IntLeaf IntLeaf::operator- ( void ) const`

5.4.2.25 `bool IntLeaf::operator< ( const IntLeaf & leaf ) const`

5.4.2.26 `IntLeaf & IntLeaf::operator= ( const IntLeaf & leaf )`

5.4.2.27 `IntLeaf & IntLeaf::operator= ( long int input )`

5.4.2.28 `IntLeaf & IntLeaf::operator= ( std::string input )`

5.4.2.29 `bool IntLeaf::operator== ( const IntLeaf & leaf ) const`

5.4.2.30 `bool IntLeaf::operator> ( const IntLeaf & leaf ) const`

5.4.2.31 `std::string IntLeaf::toString ( void ) const`

5.4.2.32 `bytevector IntLeaf::toVector ( void ) const` [virtual]

Implements [BaseNode](#).

### 5.4.3 Member Data Documentation

5.4.3.1 `const int IntLeaf::ARRAYORDER = 1` [static]

5.4.3.2 `const int IntLeaf::ENDIAN = 0` [static]

5.4.3.3 `const int IntLeaf::NAILS = 0` [static]

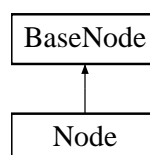
The documentation for this class was generated from the following files:

- Arithmetic/[IntLeaf.h](#)
- Arithmetic/[IntLeaf.cpp](#)

## 5.5 Node Class Reference

```
#include <Node.h>
```

Inheritance diagram for Node:



## Public Member Functions

- [Node](#) (void)
- [Node](#) (const [Node](#) &node)
- [Node](#) (const [bytevector](#) data)
- [Node](#) (const std::string filename)
- [Node](#) (std::istream &file)
- [~Node](#) (void)
- virtual [bytevector](#) [toVector](#) (void) const
- [Node](#) & [operator=](#) (const [Node](#) &node)
- [Node](#) & [modTo](#) (const [IntLeaf](#) &leaf)
- [Node](#) [mod](#) (const [IntLeaf](#) &leaf) const
- [Node](#) & [addTo](#) (const [IntLeaf](#) &leaf)
- [Node](#) [add](#) (const [IntLeaf](#) &leaf) const
- [Node](#) & [addTo](#) (const [Node](#) &node)
- [Node](#) [add](#) (const [Node](#) &node) const
- [Node](#) & [addToMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod)
- [Node](#) [addMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod) const
- [Node](#) & [addToMod](#) (const [Node](#) &node, const [IntLeaf](#) &mod)
- [Node](#) [addMod](#) (const [Node](#) &node, const [IntLeaf](#) &mod) const
- [Node](#) & [operator+=](#) (const [IntLeaf](#) &leaf)
- [Node](#) [operator+](#) (const [IntLeaf](#) &leaf) const
- [Node](#) & [operator+=](#) (const [Node](#) &node)
- [Node](#) [operator+](#) (const [Node](#) &node) const
- [Node](#) & [multTo](#) (const [IntLeaf](#) &leaf)
- [Node](#) [mult](#) (const [IntLeaf](#) &leaf) const
- [Node](#) & [multTo](#) (const [Node](#) &node)
- [Node](#) [mult](#) (const [Node](#) &node) const
- [Node](#) & [multToMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod)
- [Node](#) [multMod](#) (const [IntLeaf](#) &leaf, const [IntLeaf](#) &mod) const
- [Node](#) & [multToMod](#) (const [Node](#) &node, const [IntLeaf](#) &mod)
- [Node](#) [multMod](#) (const [Node](#) &node, const [IntLeaf](#) &mod) const
- [Node](#) & [operator\\*=](#) (const [IntLeaf](#) &leaf)
- [Node](#) [operator\\*](#) (const [IntLeaf](#) &leaf) const
- [Node](#) & [operator\\*=](#) (const [Node](#) &node)
- [Node](#) [operator\\*](#) (const [Node](#) &node) const
- bool [operator==](#) (const [Node](#) &node) const
- bool [operator!=](#) (const [Node](#) &node) const
- [IntLeaf](#) [sum](#) (void) const
- [IntLeaf](#) [sumMod](#) (const [IntLeaf](#) &mod) const
- [IntLeaf](#) [prod](#) (void) const
- [IntLeaf](#) [prodMod](#) (const [IntLeaf](#) &mod) const
- [Node](#) [exp](#) (unsigned long exponent) const
- [Node](#) [expMod](#) (unsigned long exponent, const [IntLeaf](#) &mod) const
- [Node](#) [expMod](#) (const [IntLeaf](#) &exponent, const [IntLeaf](#) &mod) const
- [Node](#) [exp](#) (const [Node](#) &exponents) const
- [Node](#) [expMod](#) (const [Node](#) &exponents, const [IntLeaf](#) &mod) const
- [Node](#) & [expTo](#) (unsigned long exponent)
- [Node](#) & [expToMod](#) (unsigned long exponent, const [IntLeaf](#) &mod)
- [Node](#) & [expToMod](#) (const [IntLeaf](#) &exponent, const [IntLeaf](#) &mod)
- [Node](#) & [expTo](#) (const [Node](#) &exponents)
- [Node](#) & [expToMod](#) (const [Node](#) &exponents, const [IntLeaf](#) &mod)
- [IntLeaf](#) [expMultMod](#) (const [Node](#) &node, const [IntLeaf](#) &mod) const
- [IntLeaf](#) [expMult](#) (unsigned long exponent) const
- [IntLeaf](#) [expMultMod](#) (unsigned long exponent, const [IntLeaf](#) &mod) const

- [Node](#) & [addChild](#) (const [BaseNode](#) &child)
- [Node](#) [getChildren](#) (int32\_t index) const
- [IntLeaf](#) & [getIntLeafChild](#) (int32\_t index)
- const [IntLeaf](#) & [getIntLeafChild](#) (int32\_t index) const
- [Node](#) & [getNodeChild](#) (int32\_t index)
- const [Node](#) & [getNodeChild](#) (int32\_t index) const
- std::string [toString](#) (void) const
- virtual int32\_t [getLength](#) (void) const

### Static Public Member Functions

- static [BaseNode](#) \* [constructPartFromFile](#) (std::istream &file, uint32\_t count)

### Additional Inherited Members

#### 5.5.1 Constructor & Destructor Documentation

- 5.5.1.1 [Node::Node](#) ( void )
- 5.5.1.2 [Node::Node](#) ( const [Node](#) & *node* )
- 5.5.1.3 [Node::Node](#) ( const [bytevector](#) *data* )
- 5.5.1.4 [Node::Node](#) ( const std::string *filename* ) [explicit]
- 5.5.1.5 [Node::Node](#) ( std::istream & *file* ) [explicit]
- 5.5.1.6 [Node::~~Node](#) ( void )

#### 5.5.2 Member Function Documentation

- 5.5.2.1 [Node](#) [Node::add](#) ( const [IntLeaf](#) & *leaf* ) const
- 5.5.2.2 [Node](#) [Node::add](#) ( const [Node](#) & *node* ) const
- 5.5.2.3 [Node](#) & [Node::addChild](#) ( const [BaseNode](#) & *child* )
- 5.5.2.4 [Node](#) [Node::addMod](#) ( const [IntLeaf](#) & *leaf*, const [IntLeaf](#) & *mod* ) const
- 5.5.2.5 [Node](#) [Node::addMod](#) ( const [Node](#) & *node*, const [IntLeaf](#) & *mod* ) const
- 5.5.2.6 [Node](#) & [Node::addTo](#) ( const [IntLeaf](#) & *leaf* )
- 5.5.2.7 [Node](#) & [Node::addTo](#) ( const [Node](#) & *node* )
- 5.5.2.8 [Node](#) & [Node::addToMod](#) ( const [IntLeaf](#) & *leaf*, const [IntLeaf](#) & *mod* )
- 5.5.2.9 [Node](#) & [Node::addToMod](#) ( const [Node](#) & *node*, const [IntLeaf](#) & *mod* )
- 5.5.2.10 [BaseNode](#) \* [Node::constructPartFromFile](#) ( std::istream & *file*, uint32\_t *count* ) [static]
- 5.5.2.11 [Node](#) [Node::exp](#) ( unsigned long *exponent* ) const
- 5.5.2.12 [Node](#) [Node::exp](#) ( const [Node](#) & *exponents* ) const

- 5.5.2.13 **Node** Node::expMod ( unsigned long *exponent*, const IntLeaf & *mod* ) const
- 5.5.2.14 **Node** Node::expMod ( const IntLeaf & *exponent*, const IntLeaf & *mod* ) const
- 5.5.2.15 **Node** Node::expMod ( const **Node** & *exponents*, const IntLeaf & *mod* ) const
- 5.5.2.16 **IntLeaf** Node::expMult ( unsigned long *exponent* ) const
- 5.5.2.17 **IntLeaf** Node::expMultMod ( const **Node** & *node*, const IntLeaf & *mod* ) const
- 5.5.2.18 **IntLeaf** Node::expMultMod ( unsigned long *exponent*, const IntLeaf & *mod* ) const
- 5.5.2.19 **Node** & Node::expTo ( unsigned long *exponent* )
- 5.5.2.20 **Node**& Node::expTo ( const **Node** & *exponents* )
- 5.5.2.21 **Node** & Node::expToMod ( unsigned long *exponent*, const IntLeaf & *mod* )
- 5.5.2.22 **Node** & Node::expToMod ( const IntLeaf & *exponent*, const IntLeaf & *mod* )
- 5.5.2.23 **Node**& Node::expToMod ( const **Node** & *exponents*, const IntLeaf & *mod* )
- 5.5.2.24 **Node** Node::getChildren ( int32\_t *index* ) const
- 5.5.2.25 **IntLeaf** & Node::getIntLeafChild ( int32\_t *index* )
- 5.5.2.26 const **IntLeaf** & Node::getIntLeafChild ( int32\_t *index* ) const
- 5.5.2.27 int32\_t Node::getLength ( void ) const [virtual]

Implements [BaseNode](#).

- 5.5.2.28 **Node** & Node::getNodeChild ( int32\_t *index* )
- 5.5.2.29 const **Node** & Node::getNodeChild ( int32\_t *index* ) const
- 5.5.2.30 **Node** Node::mod ( const IntLeaf & *leaf* ) const
- 5.5.2.31 **Node** & Node::modTo ( const IntLeaf & *leaf* )
- 5.5.2.32 **Node** Node::mult ( const IntLeaf & *leaf* ) const
- 5.5.2.33 **Node** Node::mult ( const **Node** & *node* ) const
- 5.5.2.34 **Node** Node::multMod ( const IntLeaf & *leaf*, const IntLeaf & *mod* ) const
- 5.5.2.35 **Node** Node::multMod ( const **Node** & *node*, const IntLeaf & *mod* ) const
- 5.5.2.36 **Node** & Node::multTo ( const IntLeaf & *leaf* )
- 5.5.2.37 **Node** & Node::multTo ( const **Node** & *node* )
- 5.5.2.38 **Node** & Node::multToMod ( const IntLeaf & *leaf*, const IntLeaf & *mod* )
- 5.5.2.39 **Node** & Node::multToMod ( const **Node** & *node*, const IntLeaf & *mod* )

- 5.5.2.40 `bool Node::operator!= ( const Node & node ) const`
- 5.5.2.41 `Node Node::operator* ( const IntLeaf & leaf ) const`
- 5.5.2.42 `Node Node::operator* ( const Node & node ) const`
- 5.5.2.43 `Node & Node::operator*= ( const IntLeaf & leaf )`
- 5.5.2.44 `Node & Node::operator*= ( const Node & node )`
- 5.5.2.45 `Node Node::operator+ ( const IntLeaf & leaf ) const`
- 5.5.2.46 `Node Node::operator+ ( const Node & node ) const`
- 5.5.2.47 `Node & Node::operator+= ( const IntLeaf & leaf )`
- 5.5.2.48 `Node& Node::operator+= ( const Node & node )`
- 5.5.2.49 `Node & Node::operator= ( const Node & node )`
- 5.5.2.50 `bool Node::operator== ( const Node & node ) const`
- 5.5.2.51 `IntLeaf Node::prod ( void ) const`
- 5.5.2.52 `IntLeaf Node::prodMod ( const IntLeaf & mod ) const`
- 5.5.2.53 `IntLeaf Node::sum ( void ) const`
- 5.5.2.54 `IntLeaf Node::sumMod ( const IntLeaf & mod ) const`
- 5.5.2.55 `std::string Node::toString ( void ) const`
- 5.5.2.56 `bytevector Node::toVector ( void ) const [virtual]`

Implements [BaseNode](#).

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

- [Arithmetic/Node.h](#)
- [Arithmetic/Node.cpp](#)

## 5.6 PRG Class Reference

```
#include <PRG.h>
```

### Public Member Functions

- [PRG](#) ([bytevector](#)(\*hash)([bytevector](#) data), [bytevector](#) seed, unsigned int outbits)
- [~PRG](#) (void)
- [IntLeaf next](#) ()

#### 5.6.1 Constructor & Destructor Documentation

5.6.1.1 PRG::PRG ( [bytevector](#)(\*)([bytevector](#) data) *hash*, [bytevector](#) *seed*, unsigned int *outbits* )

5.6.1.2 PRG::~~PRG ( void )

## 5.6.2 Member Function Documentation

5.6.2.1 [IntLeaf](#) PRG::next ( )

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

- [Crypto/PRG.h](#)
- [Crypto/PRG.cpp](#)

## 5.7 proofStruct Struct Reference

```
#include <Utilities.h>
```

### Public Attributes

- [IntLeaf](#) rho
- unsigned int N
- unsigned int [lambda](#)
- unsigned int [width](#)
- unsigned int [nE](#)
- unsigned int [nR](#)
- unsigned int [nV](#)
- unsigned int [nHash](#)
- [bytevector](#)(\* [hash](#) )(bytevector)
- [Node](#) Gq
- [Node](#) Rw
- [Node](#) pk
- [Node](#) y
- [Node](#) x

### 5.7.1 Member Data Documentation

5.7.1.1 [Node](#) proofStruct::Gq

5.7.1.2 [bytevector](#)(\* proofStruct::hash)(bytevector)

5.7.1.3 unsigned int proofStruct::lambda

5.7.1.4 unsigned int proofStruct::N

5.7.1.5 unsigned int proofStruct::nE

5.7.1.6 unsigned int proofStruct::nHash

5.7.1.7 unsigned int proofStruct::nR

5.7.1.8 unsigned int proofStruct::nV



5.7.1.9 Node proofStruct::pk

5.7.1.10 IntLeaf proofStruct::rho

5.7.1.11 Node proofStruct::Rw

5.7.1.12 unsigned int proofStruct::width

5.7.1.13 Node proofStruct::x

5.7.1.14 Node proofStruct::y

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

- Verifier/[Utilities.h](#)

## 5.8 RO Class Reference

```
#include <RO.h>
```

### Public Member Functions

- [RO](#) ([bytevector](#)(\*hash)([bytevector](#) data), int Nout)
- [~RO](#) (void)
- [IntLeaf operator\(\)](#) ([bytevector](#) data)

### 5.8.1 Constructor & Destructor Documentation

5.8.1.1 RO::RO ( [bytevector](#)(\*)([bytevector](#) data) *hash*, int *Nout* )

5.8.1.2 RO::~~RO ( void )

### 5.8.2 Member Function Documentation

5.8.2.1 IntLeaf RO::operator() ( [bytevector](#) *data* )

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

- Crypto/[RO.h](#)
- Crypto/[RO.cpp](#)



## Chapter 6

# File Documentation

### 6.1 Arithmetic/BaseLeaf.cpp File Reference

```
#include "BaseLeaf.h"
```

### 6.2 Arithmetic/BaseLeaf.h File Reference

```
#include "basenode.h"  
#include <vector>
```

#### Classes

- class [BaseLeaf](#)

### 6.3 Arithmetic/BaseNode.cpp File Reference

```
#include "BaseNode.h"  
#include "Node.h"  
#include "IntLeaf.h"  
#include "DataLeaf.h"
```

### 6.4 Arithmetic/BaseNode.h File Reference

```
#include <stdint.h>  
#include <istream>  
#include "types.h"
```

#### Classes

- class [BaseNode](#)

## 6.5 Arithmetic/DataLeaf.cpp File Reference

```
#include "DataLeaf.h"  
#include <stdexcept>
```

### Macros

- `#define ARRAYORDER -1 /* -1 for least significant first, 1 for most significant first */`

### 6.5.1 Macro Definition Documentation

6.5.1.1 `#define ARRAYORDER -1 /* -1 for least significant first, 1 for most significant first */`

## 6.6 Arithmetic/DataLeaf.h File Reference

```
#include "baseleaf.h"  
#include <vector>  
#include <istream>  
#include <string>
```

### Classes

- class [DataLeaf](#)

## 6.7 Arithmetic/IntLeaf.cpp File Reference

```
#include "IntLeaf.h"
```

## 6.8 Arithmetic/IntLeaf.h File Reference

```
#include "BaseLeaf.h"  
#include <gmp.h>  
#include <gmpxx.h>  
#include <string>  
#include <fstream>
```

### Classes

- class [IntLeaf](#)

## 6.9 Arithmetic/Node.cpp File Reference

```
#include "Node.h"
#include "DataLeaf.h"
#include <sstream>
#include <iomanip>
#include <stdlib.h>
#include <iterator>
#include <iostream>
```

## 6.10 Arithmetic/Node.h File Reference

```
#include "BaseNode.h"
#include "IntLeaf.h"
#include <vector>
#include <fstream>
#include <string>
```

### Classes

- class [Node](#)

## 6.11 Arithmetic/types.h File Reference

```
#include <vector>
```

### Typedefs

- typedef std::vector< unsigned char > [bytevector](#)

### 6.11.1 Typedef Documentation

6.11.1.1 typedef std::vector<unsigned char> [bytevector](#)

## 6.12 Crypto/ElGamal.cpp File Reference

```
#include "ElGamal.h"
```

### Functions

- [IntLeaf PDec](#) ([IntLeaf](#) x, [IntLeaf](#) u, [IntLeaf](#) mod)
- [IntLeaf TDec](#) ([IntLeaf](#) f, [IntLeaf](#) v, [IntLeaf](#) mod)
- [Node Enc](#) ([Node](#) pk, [IntLeaf](#) m, [IntLeaf](#) s, [IntLeaf](#) mod)

### 6.12.1 Function Documentation

6.12.1.1 **Node Enc** ( **Node** *pk*, **IntLeaf** *m*, **IntLeaf** *s*, **IntLeaf** *mod* )

6.12.1.2 **IntLeaf PDec** ( **IntLeaf** *x*, **IntLeaf** *u*, **IntLeaf** *mod* )

6.12.1.3 **IntLeaf TDec** ( **IntLeaf** *f*, **IntLeaf** *v*, **IntLeaf** *mod* )

## 6.13 Crypto/ElGamal.h File Reference

```
#include "Node.h"
#include "IntLeaf.h"
```

### Functions

- [IntLeaf PDec](#) ([IntLeaf](#) *x*, [IntLeaf](#) *c*, [IntLeaf](#) *mod*)
- [IntLeaf TDec](#) ([IntLeaf](#) *x*, [IntLeaf](#) *c*, [IntLeaf](#) *mod*)
- [Node Enc](#) ([Node](#) *pk*, [IntLeaf](#) *m*, [IntLeaf](#) *s*, [IntLeaf](#) *mod*)

### 6.13.1 Function Documentation

6.13.1.1 **Node Enc** ( **Node** *pk*, **IntLeaf** *m*, **IntLeaf** *s*, **IntLeaf** *mod* )

6.13.1.2 **IntLeaf PDec** ( **IntLeaf** *x*, **IntLeaf** *c*, **IntLeaf** *mod* )

6.13.1.3 **IntLeaf TDec** ( **IntLeaf** *x*, **IntLeaf** *c*, **IntLeaf** *mod* )

## 6.14 Crypto/H\_SHA.cpp File Reference

```
#include "H_SHA.h"
#include <algorithm>
#include <openssl/sha.h>
```

### Functions

- [bytevector H\\_SHA256](#) ([bytevector](#) *seed*)
- [bytevector H\\_SHA384](#) ([bytevector](#) *seed*)
- [bytevector H\\_SHA512](#) ([bytevector](#) *seed*)
- [bytevector H\\_SHA](#) ([unsigned char](#) \*(\*SHA)([const unsigned char](#) \**d*, [size\\_t](#) *n*, [unsigned char](#) \**md*), [bytevector](#) *seed*, [unsigned int](#) *digest\_length*)

### 6.14.1 Function Documentation

6.14.1.1 **bytevector H\_SHA** ( [unsigned char](#) \*(\*)([const unsigned char](#) \**d*, [size\\_t](#) *n*, [unsigned char](#) \**md*) *SHA*, [bytevector](#) *seed*, [unsigned int](#) *digest\_length* )

6.14.1.2 **bytevector H\_SHA256** ( [bytevector](#) *seed* )

6.14.1.3 **bytevector H\_SHA384** ( [bytevector](#) *seed* )

#### 6.14.1.4 `bytevector H_SHA512 ( bytevector seed )`

## 6.15 Crypto/H\_SHA.h File Reference

```
#include <vector>
#include "types.h"
```

### Functions

- [bytevector H\\_SHA](#) (unsigned char \*(\*SHA)(const unsigned char \*d, size\_t n, unsigned char \*md), [bytevector](#) seed, unsigned int digest\_length)
- [bytevector H\\_SHA256](#) ([bytevector](#) seed)
- [bytevector H\\_SHA384](#) ([bytevector](#) seed)
- [bytevector H\\_SHA512](#) ([bytevector](#) seed)

### 6.15.1 Function Documentation

**6.15.1.1 `bytevector H_SHA ( unsigned char *(*)(const unsigned char *d, size_t n, unsigned char *md) SHA, bytevector seed, unsigned int digest_length )`**

**6.15.1.2 `bytevector H_SHA256 ( bytevector seed )`**

**6.15.1.3 `bytevector H_SHA384 ( bytevector seed )`**

**6.15.1.4 `bytevector H_SHA512 ( bytevector seed )`**

## 6.16 Crypto/PRG.cpp File Reference

```
#include "PRG.h"
```

## 6.17 Crypto/PRG.h File Reference

```
#include "Node.h"
#include <string>
#include <vector>
#include <queue>
```

### Classes

- class [PRG](#)

## 6.18 Crypto/RandomArray.cpp File Reference

```
#include "RandomArray.h"
```

## Functions

- [Node RandomArray](#) ([Node](#) Gq, unsigned int Nprime, [bytevector](#)(\*hash)([bytevector](#) data), [bytevector](#) seed, unsigned int Nr)

### 6.18.1 Function Documentation

- 6.18.1.1 **Node RandomArray** ( [Node](#) Gq, unsigned int Nprime, [bytevector](#)(\*)([bytevector](#) data) hash, [bytevector](#) seed, unsigned int Nr )

## 6.19 Crypto/RandomArray.h File Reference

```
#include "PRG.h"
```

## Functions

- [Node RandomArray](#) ([Node](#) Gq, unsigned int Nprime, [bytevector](#)(\*hash)([bytevector](#) data), [bytevector](#) seed, unsigned int Nr)

### 6.19.1 Function Documentation

- 6.19.1.1 **Node RandomArray** ( [Node](#) Gq, unsigned int Nprime, [bytevector](#)(\*)([bytevector](#) data) hash, [bytevector](#) seed, unsigned int Nr )

## 6.20 Crypto/RO.cpp File Reference

```
#include "RO.h"
```

## 6.21 Crypto/RO.h File Reference

```
#include "Node.h"
#include <string>
#include <vector>
```

## Classes

- class [RO](#)

## 6.22 README.md File Reference

## 6.23 Tests/ByteTreeTests.cpp File Reference

```
#include <gtest/gtest.h>
#include "BaseNode.h"
#include "Node.h"
#include <stdexcept>
```



## Functions

- [TEST](#) (ByteTreeTests, NodeConstructor)
- [TEST](#) (ByteTreeTests, ConcatData)
- [TEST](#) (ByteTreeTests, Serialize)

### 6.23.1 Function Documentation

6.23.1.1 [TEST](#) ( ByteTreeTests , NodeConstructor )

6.23.1.2 [TEST](#) ( ByteTreeTests , ConcatData )

6.23.1.3 [TEST](#) ( ByteTreeTests , Serialize )

## 6.24 Tests/IntLeafArithmeticsTests.cpp File Reference

```
#include <gtest/gtest.h>
#include "IntLeaf.h"
```

## Functions

- [TEST](#) (IntLeafArithmeticTests, MultiplicationPositive)
- [TEST](#) (IntLeafArithmeticTests, AdditionPositive)
- [TEST](#) (IntLeafArithmeticTests, MultiplicationNegative)
- [TEST](#) (IntLeafArithmeticTests, AdditionNegative)

### 6.24.1 Function Documentation

6.24.1.1 [TEST](#) ( IntLeafArithmeticTests , MultiplicationPositive )

6.24.1.2 [TEST](#) ( IntLeafArithmeticTests , AdditionPositive )

6.24.1.3 [TEST](#) ( IntLeafArithmeticTests , MultiplicationNegative )

6.24.1.4 [TEST](#) ( IntLeafArithmeticTests , AdditionNegative )

## 6.25 Tests/NodeArithmeticsTests.cpp File Reference

```
#include <gtest/gtest.h>
#include "IntLeaf.h"
#include "Node.h"
```

## Functions

- [TEST](#) (NodeArithmeticsTests, AddChildren)
- [TEST](#) (NodeArithmeticsTests, VectorAddition)
- [TEST](#) (NodeArithmeticsTests, LinkedListNodeAddition)
- [TEST](#) (NodeArithmeticsTests, VectorMultiplication)

### 6.25.1 Function Documentation

6.25.1.1 `TEST ( NodeArithmeticsTests , AddChildren )`

6.25.1.2 `TEST ( NodeArithmeticsTests , VectorAddition )`

6.25.1.3 `TEST ( NodeArithmeticsTests , LinkedListNodeAddition )`

6.25.1.4 `TEST ( NodeArithmeticsTests , VectorMultiplication )`

## 6.26 Tests/NodeDataInitTests.cpp File Reference

```
#include <gtest/gtest.h>
#include "Node.h"
```

### Functions

- [TEST](#) (NodeDataInitTests, NodeFromFileTest)
- [TEST](#) (NodeDataInitTests, NodeFromFileNameTest)
- [TEST](#) (NodeDataInitTests, NodeFromVectorTest)

### 6.26.1 Function Documentation

6.26.1.1 `TEST ( NodeDataInitTests , NodeFromFileTest )`

6.26.1.2 `TEST ( NodeDataInitTests , NodeFromFileNameTest )`

6.26.1.3 `TEST ( NodeDataInitTests , NodeFromVectorTest )`

## 6.27 Tests/NodeToStringTests.cpp File Reference

```
#include <gtest/gtest.h>
#include "IntLeaf.h"
#include "Node.h"
```

### Functions

- [TEST](#) (NodeToStringTests, Vector)
- [TEST](#) (NodeToStringTests, PairVector)
- [TEST](#) (NodeToStringTests, PairSingle)

### 6.27.1 Function Documentation

6.27.1.1 `TEST ( NodeToStringTests , Vector )`

6.27.1.2 `TEST ( NodeToStringTests , PairVector )`

6.27.1.3 `TEST ( NodeToStringTests , PairSingle )`

## 6.28 Tests/PRGTests.cpp File Reference

```
#include <gtest/gtest.h>
#include "H_SHA.h"
#include "PRG.h"
#include <vector>
#include <stdexcept>
#include <algorithm>
#include <iostream>
```

### Functions

- [TEST](#) (PRGTests, TestVector256)
- [TEST](#) (PRGTests, TestVector384)
- [TEST](#) (PRGTests, TestVector512)

#### 6.28.1 Function Documentation

6.28.1.1 [TEST](#) ( PRGTests , TestVector256 )

6.28.1.2 [TEST](#) ( PRGTests , TestVector384 )

6.28.1.3 [TEST](#) ( PRGTests , TestVector512 )

## 6.29 Tests/TestRunner.cpp File Reference

```
#include <gtest/gtest.h>
```

### Functions

- [int main](#) (int argc, char \*\*argv)

#### 6.29.1 Function Documentation

6.29.1.1 [int main](#) ( int *argc*, char \*\* *argv* )

## 6.30 Verifier/DecryptionFactorsVerifier.cpp File Reference

```
#include "DecryptionFactorsVerifier.h"
#include "Node.h"
#include "RO.h"
#include "PRG.h"
#include "H_SHA.h"
#include "ElGamal.h"
#include "Utilities.h"
```

## Functions

- bool [DecryptionFactorsVerifier](#) (const int *j*, const [proofStruct](#) &*ps*, const [Node](#) &*f*, const [Node](#) &*tauDec*, const [Node](#) &*sigmaDec*, const [Node](#) &*w*)

### 6.30.1 Function Documentation

- 6.30.1.1 bool [DecryptionFactorsVerifier](#) ( const int *j*, const [proofStruct](#) & *ps*, const [Node](#) & *f*, const [Node](#) & *tauDec*, const [Node](#) & *sigmaDec*, const [Node](#) & *w* )

## 6.31 Verifier/DecryptionFactorsVerifier.h File Reference

```
#include "Utilities.h"
```

## Functions

- bool [DecryptionFactorsVerifier](#) (const int *j*, const [proofStruct](#) &*ps*, const [Node](#) &*f*, const [Node](#) &*tauDec*, const [Node](#) &*sigmaDec*, const [Node](#) &*w*)

### 6.31.1 Function Documentation

- 6.31.1.1 bool [DecryptionFactorsVerifier](#) ( const int *j*, const [proofStruct](#) & *ps*, const [Node](#) & *f*, const [Node](#) & *tauDec*, const [Node](#) & *sigmaDec*, const [Node](#) & *w* )

## 6.32 Verifier/DecryptionVerifier.cpp File Reference

```
#include "DecryptionVerifier.h"
#include "DecryptionFactorsVerifier.h"
#include "ProofOfShuffle.h"
#include "ElGamal.h"
#include "Node.h"
```

## Functions

- bool [DecryptionVerifier](#) (const [proofStruct](#) &*ps*, const [Node](#) *L*, const [Node](#) *m*)

### 6.32.1 Function Documentation

- 6.32.1.1 bool [DecryptionVerifier](#) ( const [proofStruct](#) & *ps*, const [Node](#) *L*, const [Node](#) *m* )

## 6.33 Verifier/DecryptionVerifier.h File Reference

```
#include "Node.h"
#include "Utilities.h"
```

## Functions

- bool [DecryptionVerifier](#) (const [proofStruct](#) &*ps*, const [Node](#) *L*, const [Node](#) *m*)

### 6.33.1 Function Documentation

6.33.1.1 `bool DecryptionVerifier ( const proofStruct & ps, const Node L, const Node m )`

## 6.34 Verifier/FileNames.h File Reference

```
#include <string>
```

### Variables

- `const string FULL_PUBLIC_KEY_FILE = "FullPublicKey.bt"`
- `const string CIPHERTEXTS_FILE = "Ciphertexts.bt"`
- `const string PLAINTEXTS_FILE = "Plaintexts.bt"`
- `const string SHUFFLED_CIPHERTEXTS_FILE = "ShuffledCiphertexts.bt"`
- `const string PARTIAL_PUBLIC_KEY_FILE_PREFIX = "PublicKey"`
- `const string PARTIAL_SECRET_KEY_FILE_PREFIX = "SecretKey"`
- `const string CIPHERTEXTS_FILE_PREFIX = "Ciphertexts"`
- `const string MAXCIPH_FILE = "maxciph"`
- `const string FILE_SUFFIX = ".bt"`

### 6.34.1 Variable Documentation

6.34.1.1 `const string CIPHERTEXTS_FILE = "Ciphertexts.bt"`

6.34.1.2 `const string CIPHERTEXTS_FILE_PREFIX = "Ciphertexts"`

6.34.1.3 `const string FILE_SUFFIX = ".bt"`

6.34.1.4 `const string FULL_PUBLIC_KEY_FILE = "FullPublicKey.bt"`

6.34.1.5 `const string MAXCIPH_FILE = "maxciph"`

6.34.1.6 `const string PARTIAL_PUBLIC_KEY_FILE_PREFIX = "PublicKey"`

6.34.1.7 `const string PARTIAL_SECRET_KEY_FILE_PREFIX = "SecretKey"`

6.34.1.8 `const string PLAINTEXTS_FILE = "Plaintexts.bt"`

6.34.1.9 `const string SHUFFLED_CIPHERTEXTS_FILE = "ShuffledCiphertexts.bt"`

## 6.35 Verifier/KeyVerifier.cpp File Reference

```
#include <vector>
#include <fstream>
#include "KeyVerifier.h"
#include "Node.h"
#include "IntLeaf.h"
#include "FileNames.h"
```

## Functions

- bool [keyVerifier](#) (int *lambda*, [proofStruct](#) &pfStr)
- bool [isPublicKey](#) (const [Node](#) &G, const [Node](#) &pk)
- bool [isPartialPublicKey](#) (const [Node](#) &G, const [IntLeaf](#) &ppk)
- bool [isPartialSecretKey](#) (const [Node](#) &G, const [IntLeaf](#) &psk)

### 6.35.1 Function Documentation

6.35.1.1 bool [isPartialPublicKey](#) ( const [Node](#) & *G*, const [IntLeaf](#) & *ppk* )

6.35.1.2 bool [isPartialSecretKey](#) ( const [Node](#) & *G*, const [IntLeaf](#) & *psk* )

6.35.1.3 bool [isPublicKey](#) ( const [Node](#) & *G*, const [Node](#) & *pk* )

6.35.1.4 bool [keyVerifier](#) ( int *lambda*, [proofStruct](#) & *pfStr* )

## 6.36 Verifier/KeyVerifier.h File Reference

```
#include "Node.h"
#include "Utilities.h"
#include <string>
```

## Functions

- bool [keyVerifier](#) (int *lambda*, [proofStruct](#) &pfStr)
- bool [isPublicKey](#) (const [Node](#) &G, const [Node](#) &pk)
- bool [isPartialPublicKey](#) (const [Node](#) &G, const [IntLeaf](#) &ppk)
- bool [isPartialSecretKey](#) (const [Node](#) &G, const [IntLeaf](#) &psk)

### 6.36.1 Function Documentation

6.36.1.1 bool [isPartialPublicKey](#) ( const [Node](#) & *G*, const [IntLeaf](#) & *ppk* )

6.36.1.2 bool [isPartialSecretKey](#) ( const [Node](#) & *G*, const [IntLeaf](#) & *psk* )

6.36.1.3 bool [isPublicKey](#) ( const [Node](#) & *G*, const [Node](#) & *pk* )

6.36.1.4 bool [keyVerifier](#) ( int *lambda*, [proofStruct](#) & *pfStr* )

## 6.37 Verifier/main.cpp File Reference

```
#include <iostream>
#include <string>
#include "Verifier.h"
```

## Functions

- int [SetMode](#) (const [RunMode](#) new\_mode, [RunMode](#) &mode)
- int [ParseProtinfoDirectory](#) (const int i, char \*const argv[], string &protInfo, string &directory)
- int [main](#) (int argc, char \*argv[])

### 6.37.1 Function Documentation

6.37.1.1 `int main ( int argc, char * argv[] )`

6.37.1.2 `int ParseProtinfoDirectory ( const int i, char *const argv[], string & protInfo, string & directory )`

6.37.1.3 `int SetMode ( const RunMode new_mode, RunMode & mode )`

## 6.38 Verifier/ProofOfShuffle.cpp File Reference

```
#include "ProofOfShuffle.h"
#include "PRG.h"
#include "RO.h"
#include "ElGamal.h"
#include "RandomArray.h"
#include "DataLeaf.h"
#include <cmath>
```

### Functions

- bool `proofOfShuffle` (proofStruct &pfStr, const Node &w, const Node &w\_prime, const Node &mu, const Node &tau\_pos, const Node &sigma\_pos)

### 6.38.1 Function Documentation

6.38.1.1 `bool proofOfShuffle ( proofStruct & pfStr, const Node & w, const Node & w_prime, const Node & mu, const Node & tau_pos, const Node & sigma_pos )`

## 6.39 Verifier/ProofOfShuffle.h File Reference

```
#include "Node.h"
#include "Utilities.h"
```

### Functions

- bool `proofOfShuffle` (proofStruct &pfStr, const Node &w, const Node &w\_prime, const Node &mu, const Node &tau\_pos, const Node &sigma\_pos)

### 6.39.1 Function Documentation

6.39.1.1 `bool proofOfShuffle ( proofStruct & pfStr, const Node & w, const Node & w_prime, const Node & mu, const Node & tau_pos, const Node & sigma_pos )`

## 6.40 Verifier/ShufflingVerifier.cpp File Reference

```
#include <fstream>
#include <string>
#include "Node.h"
#include "IntLeaf.h"
#include "ShufflingVerifier.h"
#include "ProofOfShuffle.h"
#include "Utilities.h"
#include "FileNames.h"
```

### Functions

- bool [verifyShuffling](#) ([proofStruct](#) &pfStr, int lambda, [Node](#) &L0, [Node](#) &Llambda, bool posc, bool ccpos)
- bool [isListOfCiphertexts](#) (const [proofStruct](#) &pfStr, [Node](#) &L)

### 6.40.1 Function Documentation

6.40.1.1 bool [isListOfCiphertexts](#) ( const [proofStruct](#) & *pfStr*, [Node](#) & *L* )

6.40.1.2 bool [verifyShuffling](#) ( [proofStruct](#) & *pfStr*, int *lambda*, [Node](#) & *L0*, [Node](#) & *Llambda*, bool *posc*, bool *ccpos* )

## 6.41 Verifier/ShufflingVerifier.h File Reference

```
#include <string>
#include "Node.h"
#include "Utilities.h"
```

### Functions

- bool [verifyShuffling](#) ([proofStruct](#) &pfStr, int lambda, [Node](#) &L0, [Node](#) &Llambda, bool posc, bool ccpos)
- bool [isListOfCiphertexts](#) (const [proofStruct](#) &pfStr, [Node](#) &L)

### 6.41.1 Function Documentation

6.41.1.1 bool [isListOfCiphertexts](#) ( const [proofStruct](#) & *pfStr*, [Node](#) & *L* )

6.41.1.2 bool [verifyShuffling](#) ( [proofStruct](#) & *pfStr*, int *lambda*, [Node](#) & *L0*, [Node](#) & *Llambda*, bool *posc*, bool *ccpos* )

## 6.42 Verifier/Utilities.cpp File Reference

```
#include "Utilities.h"
#include <vector>
```

### Functions

- bool [isElemOfGq](#) (const [Node](#) &group, const [IntLeaf](#) &elem)
- bool [isElemOfZn](#) (const [IntLeaf](#) &n, const [IntLeaf](#) &elem)



- bool `isElemOfMw` (const `proofStruct` &`pfStr`, const `Node` &`plaintext`)
- bool `isElemOfCw` (const `proofStruct` &`pfStr`, const `Node` &`ciphertext`)
- bool `isPedersenCommitment` (const `Node` &`group`, const `IntLeaf` &`elem`)
- void `getGroupFromString` (`proofStruct` &`pfStr`, std::string `str`)

### 6.42.1 Function Documentation

6.42.1.1 void `getGroupFromString` ( `proofStruct` & *pfStr*, std::string *str* )

6.42.1.2 bool `isElemOfCw` ( const `proofStruct` & *pfStr*, const `Node` & *ciphertext* )

6.42.1.3 bool `isElemOfGq` ( const `Node` & *group*, const `IntLeaf` & *elem* )

6.42.1.4 bool `isElemOfMw` ( const `proofStruct` & *pfStr*, const `Node` & *plaintext* )

6.42.1.5 bool `isElemOfZn` ( const `IntLeaf` & *n*, const `IntLeaf` & *elem* )

6.42.1.6 bool `isPedersenCommitment` ( const `Node` & *group*, const `IntLeaf` & *elem* )

## 6.43 Verifier/Utilities.h File Reference

```
#include <string>
#include "Node.h"
#include "IntLeaf.h"
```

### Classes

- struct `proofStruct`

### Functions

- bool `isElemOfGq` (const `Node` &`group`, const `IntLeaf` &`elem`)
- bool `isElemOfZn` (const `IntLeaf` &`n`, const `IntLeaf` &`elem`)
- bool `isElemOfCw` (const `proofStruct` &`pfStr`, const `Node` &`ciphertext`)
- bool `isElemOfMw` (const `proofStruct` &`pfStr`, const `Node` &`plaintext`)
- bool `isPedersenCommitment` (const `Node` &`group`, const `IntLeaf` &`elem`)
- void `getGroupFromString` (`proofStruct` &`pfStr`, std::string `str`)

### Variables

- const `IntLeaf` `BOTTOM` = `IntLeaf`(-1)

### 6.43.1 Function Documentation

6.43.1.1 void `getGroupFromString` ( `proofStruct` & *pfStr*, std::string *str* )

6.43.1.2 bool `isElemOfCw` ( const `proofStruct` & *pfStr*, const `Node` & *ciphertext* )

6.43.1.3 bool `isElemOfGq` ( const `Node` & *group*, const `IntLeaf` & *elem* )

6.43.1.4 `bool isElemOfMw ( const proofStruct & pfStr, const Node & plaintext )`

6.43.1.5 `bool isElemOfZn ( const IntLeaf & n, const IntLeaf & elem )`

6.43.1.6 `bool isPedersenCommitment ( const Node & group, const IntLeaf & elem )`

## 6.43.2 Variable Documentation

6.43.2.1 `const IntLeaf BOTTOM = IntLeaf(-1)`

## 6.44 Verifier/Verifier.cpp File Reference

```
#include "Verifier.h"
#include "Utilities.h"
#include "DecryptionVerifier.h"
#include "ShufflingVerifier.h"
#include "KeyVerifier.h"
#include "FileNames.h"
#include <string>
#include <fstream>
#include <vector>
#include <rapidxml/rapidxml.hpp>
#include "DataLeaf.h"
#include "Node.h"
#include "H_SHA.h"
```

### Functions

- `int Verifier (string protinfo, string directory, RunMode typeExp, string auxsidExp, int wExp, bool posc, bool ccpos, bool dec)`

### 6.44.1 Function Documentation

6.44.1.1 `int Verifier ( string protinfo, string directory, RunMode typeExp, string auxsidExp, int wExp, bool posc, bool ccpos, bool dec )`

## 6.45 Verifier/Verifier.h File Reference

```
#include <string>
```

### Enumerations

- `enum RunMode {  
NONE, HELP, COMPAT, MIX,  
SHUFFLE, DECRYPT }`

### Functions

- `int Verifier (string protinfo, string directory, RunMode typeExp, string auxsidExp, int wExp, bool posc, bool ccpos, bool dec)`

## Variables

- `const std::string CIPHERTEXT_FILE_PREFIX = "Ciphertexts"`

## 6.45.1 Enumeration Type Documentation

### 6.45.1.1 enum RunMode

Enumerator

***NONE***

***HELP***

***COMPAT***

***MIX***

***SHUFFLE***

***DECRYPT***

## 6.45.2 Function Documentation

6.45.2.1 `int Verifier ( string protinfo, string directory, RunMode typeExp, string auxsidExp, int wExp, bool posc, bool ccpos, bool dec )`

## 6.45.3 Variable Documentation

6.45.3.1 `const std::string CIPHERTEXT_FILE_PREFIX = "Ciphertexts"`

# Index

- ~BaseLeaf
  - BaseLeaf, 9
- ~BaseNode
  - BaseNode, 10
- ~DataLeaf
  - DataLeaf, 12
- ~IntLeaf
  - IntLeaf, 14
- ~Node
  - Node, 17
- ~PRG
  - PRG, 20
- ~RO
  - RO, 21
- ARRAYORDER
  - DataLeaf.cpp, 24
  - IntLeaf, 15
- add
  - IntLeaf, 14
  - Node, 17
- addChild
  - Node, 17
- addMod
  - IntLeaf, 14
  - Node, 17
- addTo
  - IntLeaf, 14
  - Node, 17
- addToMod
  - IntLeaf, 14
  - Node, 17
- Arithmetic/BaseLeaf.cpp, 23
- Arithmetic/BaseLeaf.h, 23
- Arithmetic/BaseNode.cpp, 23
- Arithmetic/BaseNode.h, 23
- Arithmetic/DataLeaf.cpp, 24
- Arithmetic/DataLeaf.h, 24
- Arithmetic/IntLeaf.cpp, 24
- Arithmetic/IntLeaf.h, 24
- Arithmetic/Node.cpp, 25
- Arithmetic/Node.h, 25
- Arithmetic/types.h, 25
- BOTTOM
  - Utilities.h, 38
- BaseNode
  - DATA\_LEAF, 10
  - INT\_LEAF, 10
  - NODE, 10
- BaseLeaf, 9
  - ~BaseLeaf, 9
  - BaseLeaf, 9
  - BaseLeaf, 9
- BaseNode, 9
  - ~BaseNode, 10
  - BaseNode, 10
  - BaseNode, 10
  - concatData, 10
  - copy, 10
  - getLength, 11
  - getType, 11
  - NodeType, 10
  - ReadNodeHeader, 11
  - serialize, 11
  - toVector, 11
- ByteTreeTests.cpp
  - TEST, 29
- bytevector
  - types.h, 25
- COMPAT
  - Verifier.h, 39
- CIPHERTEXTS\_FILE
  - FileNames.h, 33
- concatData
  - BaseNode, 10
- constructPartFromFile
  - IntLeaf, 14
  - Node, 17
- copy
  - BaseNode, 10
- Crypto/EIGamal.cpp, 25
- Crypto/EIGamal.h, 26
- Crypto/H\_SHA.cpp, 26
- Crypto/H\_SHA.h, 27
- Crypto/PRG.cpp, 27
- Crypto/PRG.h, 27
- Crypto/RO.cpp, 28
- Crypto/RO.h, 28
- Crypto/RandomArray.cpp, 27
- Crypto/RandomArray.h, 28
- DATA\_LEAF
  - BaseNode, 10
- DECRYPT
  - Verifier.h, 39
- DataLeaf, 11
  - ~DataLeaf, 12
  - DataLeaf, 11, 12

- DataLeaf, 11, 12
- getData, 12
- getLength, 12
- operator=, 12
- toVector, 12
- DataLeaf.cpp
  - ARRAYORDER, 24
- DecryptionFactorsVerifier
  - DecryptionFactorsVerifier.cpp, 32
  - DecryptionFactorsVerifier.h, 32
- DecryptionFactorsVerifier.cpp
  - DecryptionFactorsVerifier, 32
- DecryptionFactorsVerifier.h
  - DecryptionFactorsVerifier, 32
- DecryptionVerifier
  - DecryptionVerifier.cpp, 32
  - DecryptionVerifier.h, 33
- DecryptionVerifier.cpp
  - DecryptionVerifier, 32
- DecryptionVerifier.h
  - DecryptionVerifier, 33
- ENDIAN
  - IntLeaf, 15
- ElGamal.cpp
  - Enc, 26
  - PDec, 26
  - TDec, 26
- ElGamal.h
  - Enc, 26
  - PDec, 26
  - TDec, 26
- Enc
  - ElGamal.cpp, 26
  - ElGamal.h, 26
- exp
  - IntLeaf, 14
  - Node, 17
- expMod
  - IntLeaf, 14
  - Node, 17, 18
- expMult
  - Node, 18
- expMultMod
  - Node, 18
- expTo
  - IntLeaf, 14
  - Node, 18
- expToMod
  - IntLeaf, 14
  - Node, 18
- FILE\_SUFFIX
  - FileNames.h, 33
- FileNames.h
  - CIPHERTEXTS\_FILE, 33
  - FILE\_SUFFIX, 33
  - MAXCIPH\_FILE, 33
  - PLAINTEXTS\_FILE, 33
- getBigInt
  - IntLeaf, 14
- getChildren
  - Node, 18
- getData
  - DataLeaf, 12
- getGroupFromString
  - Utilities.cpp, 37
  - Utilities.h, 37
- getIntLeafChild
  - Node, 18
- getLength
  - BaseNode, 11
  - DataLeaf, 12
  - IntLeaf, 14
  - Node, 18
- getNodeChild
  - Node, 18
- getType
  - BaseNode, 11
- Gq
  - proofStruct, 20
- HELP
  - Verifier.h, 39
- H\_SHA
  - H\_SHA.cpp, 26
  - H\_SHA.h, 27
- H\_SHA.cpp
  - H\_SHA, 26
  - H\_SHA256, 26
  - H\_SHA384, 26
  - H\_SHA512, 26
- H\_SHA.h
  - H\_SHA, 27
  - H\_SHA256, 27
  - H\_SHA384, 27
  - H\_SHA512, 27
- H\_SHA256
  - H\_SHA.cpp, 26
  - H\_SHA.h, 27
- H\_SHA384
  - H\_SHA.cpp, 26
  - H\_SHA.h, 27
- H\_SHA512
  - H\_SHA.cpp, 26
  - H\_SHA.h, 27
- hash
  - proofStruct, 20
- INT\_LEAF
  - BaseNode, 10
- IntLeaf, 12
  - ~IntLeaf, 14
  - ARRAYORDER, 15
  - add, 14
  - addMod, 14
  - addTo, 14
  - addToMod, 14

- constructPartFromFile, 14
- ENDIAN, 15
- exp, 14
- expMod, 14
- expTo, 14
- expToMod, 14
- getBigInt, 14
- getLength, 14
- IntLeaf, 13, 14
- IntLeaf, 13, 14
- inverse, 14
- mod, 14
- modTo, 14
- mult, 14
- multMod, 14
- multTo, 14
- multToMod, 14
- NAILS, 15
- operator<, 15
- operator>, 15
- operator\*, 14
- operator\*=: 15
- operator+, 15
- operator+=, 15
- operator-, 15
- operator=, 15
- operator==, 15
- toString, 15
- toVector, 15
- IntLeafArithmeticsTests.cpp
  - TEST, 29
- inverse
  - IntLeaf, 14
- isElemOfCw
  - Utilities.cpp, 37
  - Utilities.h, 37
- isElemOfGq
  - Utilities.cpp, 37
  - Utilities.h, 37
- isElemOfMw
  - Utilities.cpp, 37
  - Utilities.h, 37
- isElemOfZn
  - Utilities.cpp, 37
  - Utilities.h, 38
- isListOfCiphertexts
  - ShufflingVerifier.cpp, 36
  - ShufflingVerifier.h, 36
- isPartialPublicKey
  - KeyVerifier.cpp, 34
  - KeyVerifier.h, 34
- isPartialSecretKey
  - KeyVerifier.cpp, 34
  - KeyVerifier.h, 34
- isPedersenCommitment
  - Utilities.cpp, 37
  - Utilities.h, 38
- isPublicKey
  - KeyVerifier.cpp, 34
  - KeyVerifier.h, 34
- keyVerifier
  - KeyVerifier.cpp, 34
  - KeyVerifier.h, 34
- KeyVerifier.cpp
  - isPartialPublicKey, 34
  - isPartialSecretKey, 34
  - isPublicKey, 34
  - keyVerifier, 34
- KeyVerifier.h
  - isPartialPublicKey, 34
  - isPartialSecretKey, 34
  - isPublicKey, 34
  - keyVerifier, 34
- lambda
  - proofStruct, 20
- MIX
  - Verifier.h, 39
- MAXCIPH\_FILE
  - FileNames.h, 33
- main
  - main.cpp, 35
  - TestRunner.cpp, 31
- main.cpp
  - main, 35
  - ParseProtinfoDirectory, 35
  - SetMode, 35
- mod
  - IntLeaf, 14
  - Node, 18
- modTo
  - IntLeaf, 14
  - Node, 18
- mult
  - IntLeaf, 14
  - Node, 18
- multMod
  - IntLeaf, 14
  - Node, 18
- multTo
  - IntLeaf, 14
  - Node, 18
- multToMod
  - IntLeaf, 14
  - Node, 18
- N
  - proofStruct, 20
- NODE
  - BaseNode, 10
- NONE
  - Verifier.h, 39
- NAILS
  - IntLeaf, 15
- nE

- proofStruct, 20
- nHash
  - proofStruct, 20
- nR
  - proofStruct, 20
- nV
  - proofStruct, 20
- next
  - PRG, 20
- Node, 15
  - ~Node, 17
  - add, 17
  - addChild, 17
  - addMod, 17
  - addTo, 17
  - addToMod, 17
  - constructPartFromFile, 17
  - exp, 17
  - expMod, 17, 18
  - expMult, 18
  - expMultMod, 18
  - expTo, 18
  - expToMod, 18
  - getChildren, 18
  - getIntLeafChild, 18
  - getLength, 18
  - getNodeChild, 18
  - mod, 18
  - modTo, 18
  - mult, 18
  - multMod, 18
  - multTo, 18
  - multToMod, 18
  - Node, 17
  - operator\*, 19
  - operator\*==, 19
  - operator+, 19
  - operator+=, 19
  - operator=, 19
  - operator==, 19
  - prod, 19
  - prodMod, 19
  - sum, 19
  - sumMod, 19
  - toString, 19
  - toVector, 19
- NodeArithmeticsTests.cpp
  - TEST, 30
- NodeDataInitTests.cpp
  - TEST, 30
- NodeToStringTests.cpp
  - TEST, 30
- NodeType
  - BaseNode, 10
- operator<
  - IntLeaf, 15
- operator>
  - IntLeaf, 15
- operator\*
  - IntLeaf, 14
  - Node, 19
- operator\*=
  - IntLeaf, 15
  - Node, 19
- operator()
  - RO, 21
- operator+
  - IntLeaf, 15
  - Node, 19
- operator+=
  - IntLeaf, 15
  - Node, 19
- operator-
  - IntLeaf, 15
- operator=
  - DataLeaf, 12
  - IntLeaf, 15
  - Node, 19
- operator==
  - IntLeaf, 15
  - Node, 19
- PDec
  - ElGamal.cpp, 26
  - ElGamal.h, 26
- PLAINTEXTS\_FILE
  - FileNames.h, 33
- PRG, 19
  - ~PRG, 20
  - next, 20
  - PRG, 19
  - PRG, 19
- PRGTests.cpp
  - TEST, 31
- ParseProtinfoDirectory
  - main.cpp, 35
- pk
  - proofStruct, 20
- prod
  - Node, 19
- prodMod
  - Node, 19
- proofOfShuffle
  - ProofOfShuffle.cpp, 35
  - ProofOfShuffle.h, 35
- ProofOfShuffle.cpp
  - proofOfShuffle, 35
- ProofOfShuffle.h
  - proofOfShuffle, 35
- proofStruct, 20
  - Gq, 20
  - hash, 20
  - lambda, 20
  - N, 20
  - nE, 20
  - nHash, 20
  - nR, 20

- nV, 20
- pk, 20
- rho, 21
- Rw, 21
- width, 21
- x, 21
- y, 21
- README.md, 28
- RO, 21
  - ~RO, 21
  - operator(), 21
  - RO, 21
  - RO, 21
- RandomArray
  - RandomArray.cpp, 28
  - RandomArray.h, 28
- RandomArray.cpp
  - RandomArray, 28
- RandomArray.h
  - RandomArray, 28
- ReadNodeHeader
  - BaseNode, 11
- rho
  - proofStruct, 21
- RunMode
  - Verifier.h, 39
- Rw
  - proofStruct, 21
- SHUFFLE
  - Verifier.h, 39
- serialize
  - BaseNode, 11
- SetMode
  - main.cpp, 35
- ShufflingVerifier.cpp
  - isListOfCiphertexts, 36
  - verifyShuffling, 36
- ShufflingVerifier.h
  - isListOfCiphertexts, 36
  - verifyShuffling, 36
- sum
  - Node, 19
- sumMod
  - Node, 19
- TDec
  - ElGamal.cpp, 26
  - ElGamal.h, 26
- TEST
  - ByteTreeTests.cpp, 29
  - IntLeafArithmeticsTests.cpp, 29
  - NodeArithmeticsTests.cpp, 30
  - NodeDataInitTests.cpp, 30
  - NodeToStringTests.cpp, 30
  - PRGTests.cpp, 31
- TestRunner.cpp
  - main, 31
- Tests/ByteTreeTests.cpp, 28
- Tests/IntLeafArithmeticsTests.cpp, 29
- Tests/NodeArithmeticsTests.cpp, 29
- Tests/NodeDataInitTests.cpp, 30
- Tests/NodeToStringTests.cpp, 30
- Tests/PRGTests.cpp, 31
- Tests/TestRunner.cpp, 31
- toString
  - IntLeaf, 15
  - Node, 19
- toVector
  - BaseNode, 11
  - DataLeaf, 12
  - IntLeaf, 15
  - Node, 19
- types.h
  - bytevector, 25
- Utilities.cpp
  - getGroupFromString, 37
  - isElemOfCw, 37
  - isElemOfGq, 37
  - isElemOfMw, 37
  - isElemOfZn, 37
  - isPedersenCommitment, 37
- Utilities.h
  - BOTTOM, 38
  - getGroupFromString, 37
  - isElemOfCw, 37
  - isElemOfGq, 37
  - isElemOfMw, 37
  - isElemOfZn, 38
  - isPedersenCommitment, 38
- Verifier
  - Verifier.cpp, 38
  - Verifier.h, 39
- Verifier.h
  - COMPAT, 39
  - DECRYPT, 39
  - HELP, 39
  - MIX, 39
  - NONE, 39
  - SHUFFLE, 39
- Verifier.cpp
  - Verifier, 38
- Verifier.h
  - RunMode, 39
  - Verifier, 39
- Verifier/DecryptionFactorsVerifier.cpp, 31
- Verifier/DecryptionFactorsVerifier.h, 32
- Verifier/DecryptionVerifier.cpp, 32
- Verifier/DecryptionVerifier.h, 32
- Verifier/FileNames.h, 33
- Verifier/KeyVerifier.cpp, 33
- Verifier/KeyVerifier.h, 34
- Verifier/ProofOfShuffle.cpp, 35
- Verifier/ProofOfShuffle.h, 35
- Verifier/ShufflingVerifier.cpp, 36



---

Verifier/ShufflingVerifier.h, [36](#)  
Verifier/Utilities.cpp, [36](#)  
Verifier/Utilities.h, [37](#)  
Verifier/Verifier.cpp, [38](#)  
Verifier/Verifier.h, [38](#)  
Verifier/main.cpp, [34](#)  
verifyShuffling  
    ShufflingVerifier.cpp, [36](#)  
    ShufflingVerifier.h, [36](#)  
  
width  
    proofStruct, [21](#)  
  
x  
    proofStruct, [21](#)  
  
y  
    proofStruct, [21](#)