cryptocast.crypto

BroadcastEncryptionClient

+recv(size:int,buffer:byte[])

<<generic>>
Decryptor<S>

+decrypt(cipher:byte[]): S

Field<BigInteger>

<<generic>>
Field<T>

+add(a:T,b:T): T
+multiply(a:T,b:T): T
+negate(a:T): T
+invert(a:T): T
+zero(): T
+one(): T
+randomElement(): T
+subtract(a:T,b:T): T
+divide(a:T,b:T): T
+pow(a:T,e:int): T

<<bind>>

<<generic>>
ShareCombinator<S, T>

+restore(shares:Collection<T>): Optional<S>

<<generic>>
BroadcastSchemeUserManager<ID>

+getIdentity(i:int): ID
+revoke(id:ID)
+isRevoked(id:ID): boolean

IntegersModuloPrime

+add(a:BigInteger,b:BigInteger): BigInteger
+multiply(a:BigInteger,b:BigInteger): BigInteger
+negate(a:BigInteger): BigInteger
+invert(a:BigInteger): BigInteger
+zero(): BigInteger
+one(): BigInteger
+randomElement(): BigInteger

<<generic>>
BroadcastEncryptionServer<ID>

+run()
+send(data:byte[])
+revoke(id:ID)

<<generic>>
BroadcastSchemeKeyManager<ID>

+getPersonalKey(id:ID): PrivateKey

<<generic>>
Polynomial<T>

+getField(): Field<T>
+evaluate(x:T): T
+evaluateMulti(xs:T[]): T[]
+getCoefficient(i:int): T
+getDegree(): int
+createRandomPolynomial(field:Field<T>,degree:int): Polynomial<T>

<<generic>>
Encryptor<S>

+encrypt(secret:S): byte[]

<<generic>>
LagrangeInterpolation<T>

+computeCoefficients(): T[]

<<bind>>