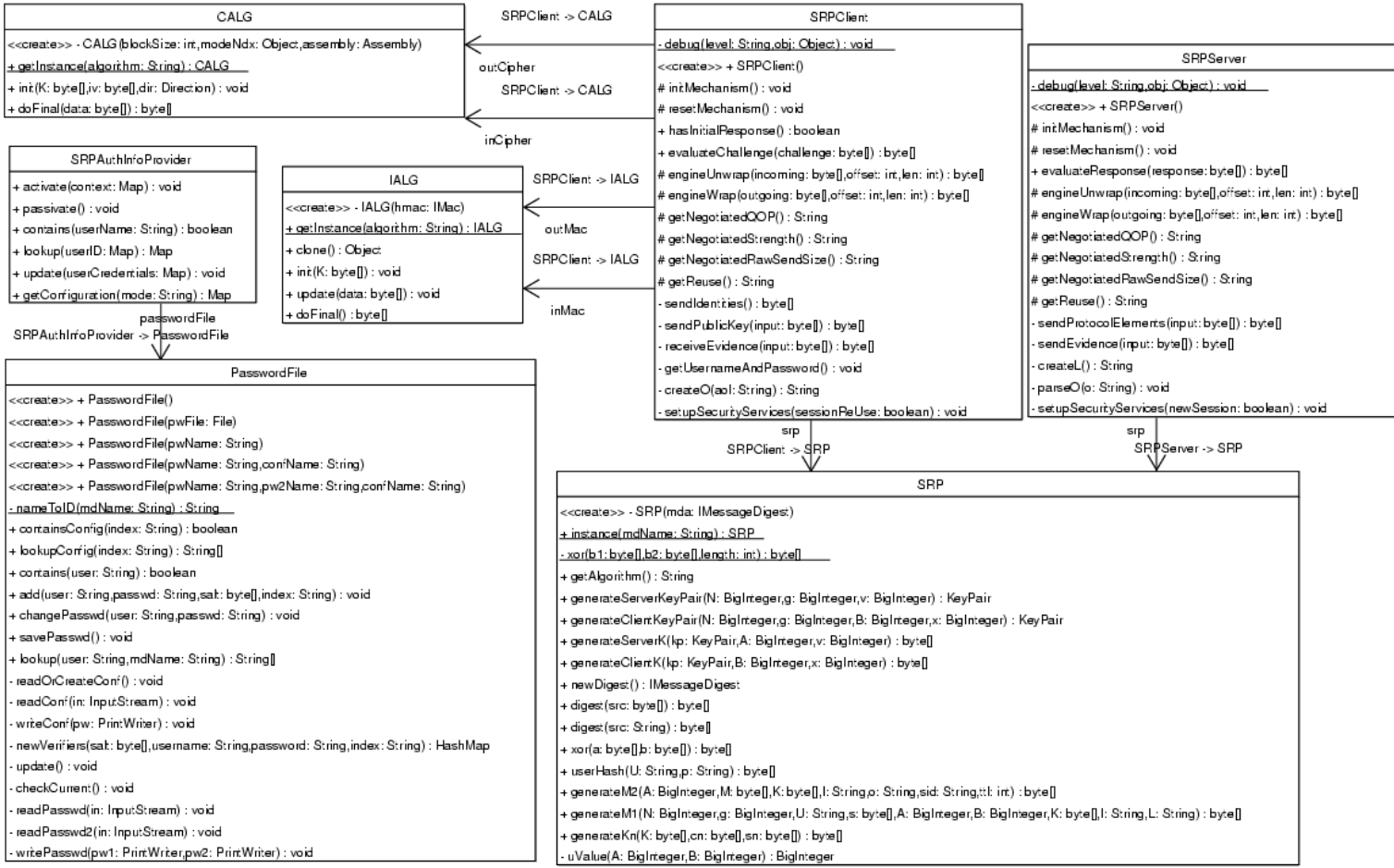
CALG
<<create>> - CALG(blockSize: int,modeNdx: Object,assembly: Assembly)
+ getInstance(algorithm: String) : CALG
+ init(K: byte[],iv: byte[],dir: Direction) : void
+ doFinal(data: byte[]) : byte[]
SRPClient -> CALG
outCipher
SRPClient -> CALG
inCipher
SRPClient
- debug(level: String,obj: Object) : void
<<create>> + SRPClient()
initMechanism() : void
reseMechanism() : void
+ hasInitialResponse() : boolean
+ evaluateChallenge(challenge: byte[]) : byte[]
# engineUnwrap(incoming: byte[],offset: int,len: int) : byte[]
# engineWrap(outgoing: byte[],offset: int,len: int) : byte[]
# getNegotiatedQOP() : String
# getNegotiatedStrength() : String
# getNegotiatedRawSendSize() : String
# getReuse() : String
- sendIdentities() : byte[]
- sendPublicKey(input: byte[]) : byte[]
- receiveEvidence(input: byte[]) : byte[]
- getUsernameAndPassword() : void
- createO(aol: String) : String
- setupSecurityServices(sessionReUse: boolean) : void
SRPServer
- debug(level: String,obj: Object) : void
<<create>> + SRPServer()
initMechanism() : void
reseMechanism() : void
+ evaluateResponse(response: byte[]) : byte[]
# engineUnwrap(incoming: byte[],offset: int,len: int) : byte[]
# engineWrap(outgoing: byte[],offset: int,len: int) : byte[]
# getNegotiatedQOP() : String
# getNegotiatedStrength() : String
# getNegotiatedRawSendSize() : String
# getReuse() : String
- sendProtocolElements(input: byte[]) : byte[]
- sendEvidence(input: byte[]) : byte[]
- createL() : String
- parseO(o: String) : void
- setupSecurityServices(newSession: boolean) : void
SRPAuthInfoProvider
+ activate(context: Map) : void
+ passivate() : void
+ contains(userName: String) : boolean
+ lookup(userID: Map) : Map
+ update(userCredentials: Map) : void
+ getConfiguration(mode: String) : Map
IALG
<<create>> - IALG(hmac: IMac)
+ getInstance(algorithm: String) : IALG
+ clone() : Object
+ init(K: byte[]) : void
+ update(data: byte[]) : void
+ doFinal() : byte[]
SRPClient -> IALG
outMac
SRPClient -> IALG
inMac
passwordFile
SRPAuthInfoProvider -> PasswordFile
PasswordFile
<<create>> + PasswordFile()
<<create>> + PasswordFile(pwFile: File)
<<create>> + PasswordFile(pwName: String)
<<create>> + PasswordFile(pwName: String,confName: String)
<<create>> + PasswordFile(pwName: String,pw2Name: String,confName: String)
- nameToID(mdName: String) : String
+ containsConfig(index: String) : boolean
+ lookupConfig(index: String) : String[]
+ contains(user: String) : boolean
+ add(user: String,passwd: String,salt: byte[],index: String) : void
+ changePasswd(user: String,passwd: String) : void
+ savePasswd() : void
+ lookup(user: String,mdName: String) : String[]
- readOrCreateConf() : void
- readConf(in: InputStream) : void
- writeConf(pw: PrintWriter) : void
- newVerifiers(salt: byte[],username: String,password: String,index: String) : HashMap
- update() : void
- checkCurrent() : void
- readPasswd(in: InputStream) : void
- readPasswd2(in: InputStream) : void
- writePasswd(pw1: PrintWriter,pw2: PrintWriter) : void
srp
SRPClient -> SRP
srp
SRPServer -> SRP
SRP
<<create>> - SRP(mda: IMessageDigest)
+ instance(mdName: String) : SRP
- xor(b1: byte[],b2: byte[],length: int) : byte[]
+ getAlgorithm() : String
+ generateServerKeyPair(N: BigInteger,g: BigInteger,v: BigInteger) : KeyPair
+ generateClientKeyPair(N: BigInteger,g: BigInteger,B: BigInteger,x: BigInteger) : KeyPair
+ generateServerK(kp: KeyPair,A: BigInteger,v: BigInteger) : byte[]
+ generateClientK(kp: KeyPair,B: BigInteger,x: BigInteger) : byte[]
+ newDigest() : IMessageDigest
+ digest(src: byte[]) : byte[]
+ digest(src: String) : byte[]
+ xor(a: byte[],b: byte[]) : byte[]
+ userHash(U: String,p: String) : byte[]
+ generateM2(A: BigInteger,M: byte[],K: byte[],I: String,o: String,sid: String,ttl: int) : byte[]
+ generateM1(N: BigInteger,g: BigInteger,U: String,s: byte[],A: BigInteger,B: BigInteger,K: byte[],I: String,L: String) : byte[]
+ generateKn(K: byte[],cn: byte[],sn: byte[]) : byte[]
- uValue(A: BigInteger,B: BigInteger) : BigInteger