ParrotTalk v3.6/v3.7 Callisto House LLC

INSTALL

To install ParrotTalk in Java, first clone the ASN1 project, then the ParrotTalk project. Link the ASN1 project into ParrotTalk and run the ParrotTalk tests.

OVERVIEW

ParrotTalk is an encrypted connection framework. Currently allowing anonymous 2048-bit key negotiation to establish user-provided encryption cipher and user-provided encoding and decoding, both through a provided SessionAgentMap to a starting SessionAgent server. Please look in the test case ThunkHelloWorldTest for building these maps and running a connection iwth data passing after encryption is established. There is a 3-message negotiation, from Hello_v3_7 to Signature_v3_7. Uses RSA 2048 signature validation and DH 2048 primes to establish the key used within the selected Cipher. The Cipher and Encoder are selected by name through the negotiation protocol. Currently three Ciphers are selectable, AESede, DESede, and DES. There are three encoders tested, asn1der, String, and Bytes. This protocol is described here, in these documents.

Here is the slides describing the version 3.7 protocol:
https://github.com/CallistoHouseLtd/ParrotTalk/blob/master/docs/ParrotTalkFrameDesign-3.7.pdf

Here is the previous yet still supported, version 3.6 protocol:
https://github.com/CallistoHouseLtd/ParrotTalk/blob/master/docs/ParrotTalkFrameDesign-3.6.pdf

and an IETF draft document, specific to version 3.6

https://github.com/CallistoHouseLtd/ParrotTalk/blob/master/docs/draft-withers-parrot-talk-v36-00.pdf

For as to use cases, this encrypted connection has no third party, man-in-the-middle situation by not using Certificates. As such, this is a tight implementation of NSA-proof encryption without explicit authorization beyond knowledge of a host:port. The use cases involve any communication desired to be encrypted with such high encryption. The support will last my lifetime, so we have a settled solution, here in the third version, provided here. It requires version 115 of Cryptography, as a prerequisite. Both work in Squeak (https://squeak.org/) and Pharo (http://pharo.org/). In any image, load these packages, please, then check out the ParrotTalk tests.

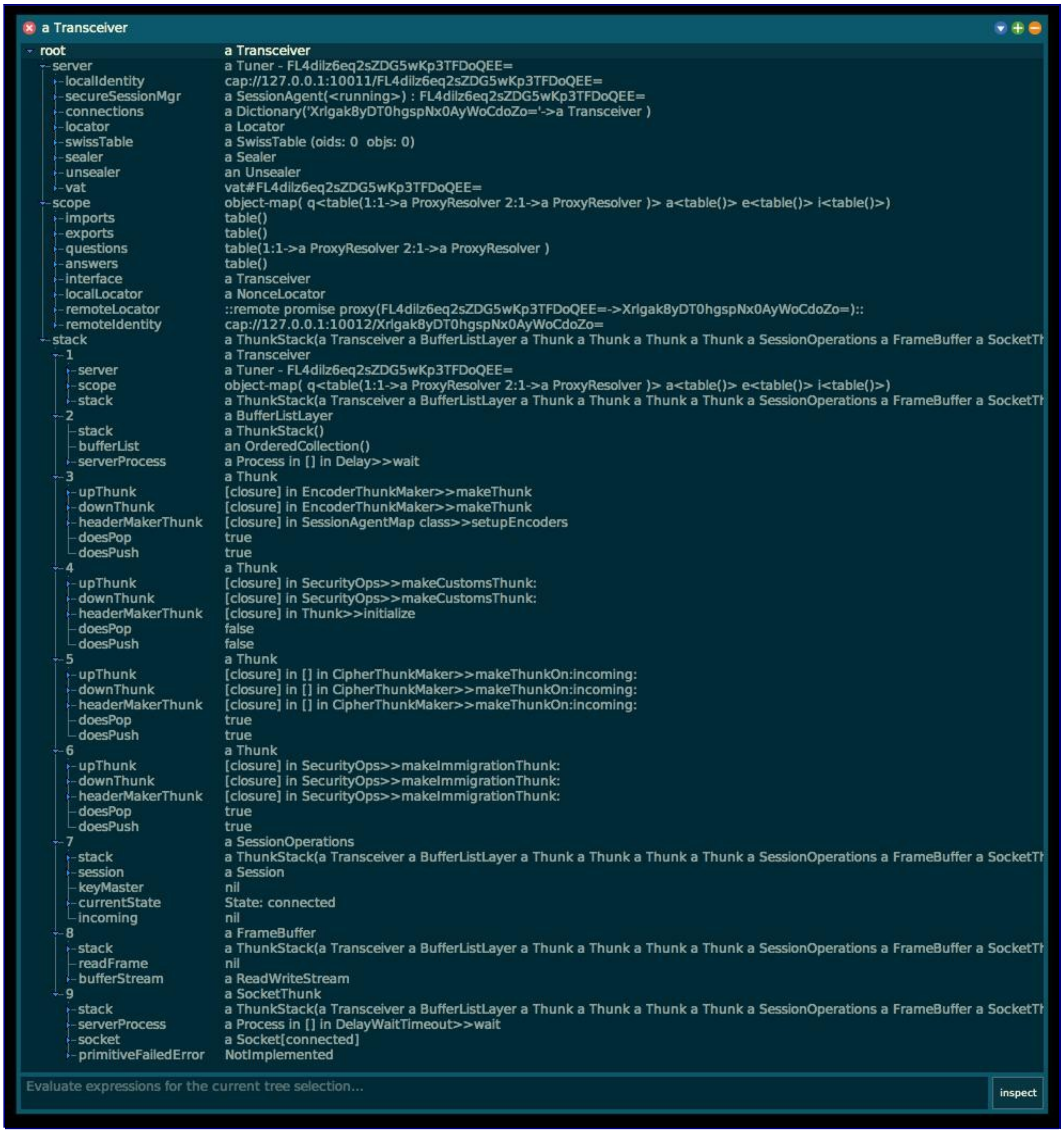
http://www.squeaksource.com/Cryptography/Cryptography-rww.115.mcz

http://www.squeaksource.com/Cryptography/ParrotTalk-rww.25.mcz

For Java, currently supporting version 3.6 only yet still able to connect to a Squeak/Pharo agent, clone these repositories:

https://github.com/CallistoHouseLtd/ASN1

https://github.com/CallistoHouseLtd/ParrotTalk



```
a Transceiver
root                    a Transceiver
 server                 a Tuner - FL4dilz6eq2sZDG5wKp3TFDoQEE=
  localIdentity         cap://127.0.0.1:10011/FL4dilz6eq2sZDG5wKp3TFDoQEE=
  secureSessionMgr      a SessionAgent(<running>) : FL4dilz6eq2sZDG5wKp3TFDoQEE=
  connections           a Dictionary('Xrlgak8yDT0hgspNx0AyWoCdoZo='->a Transceiver )
  locator               a Locator
  swissTable            a SwissTable (oids: 0  objs: 0)
  sealer                a Sealer
  unsealer              an Unsealer
  vat                   vat#FL4dilz6eq2sZDG5wKp3TFDoQEE=
 scope                  object-map( q<table(1:1->a ProxyResolver 2:1->a ProxyResolver )> a<table()> e<table()> i<table()>)
  imports               table()
  exports               table()
  questions             table(1:1->a ProxyResolver 2:1->a ProxyResolver )
  answers               table()
  interface             a Transceiver
  localLocator          a NonceLocator
  remoteLocator         ::remote promise proxy(FL4dilz6eq2sZDG5wKp3TFDoQEE=->Xrlgak8yDT0hgspNx0AyWoCdoZo=)::
  remoteIdentity        cap://127.0.0.1:10012/Xrlgak8yDT0hgspNx0AyWoCdoZo=
 stack                  a ThunkStack(a Transceiver a BufferListLayer a Thunk a Thunk a Thunk a Thunk a SessionOperations a FrameBuffer a SocketTh
  1                     a Transceiver
   server               a Tuner - FL4dilz6eq2sZDG5wKp3TFDoQEE=
   scope                object-map( q<table(1:1->a ProxyResolver 2:1->a ProxyResolver )> a<table()> e<table()> i<table()>)
   stack                a ThunkStack(a Transceiver a BufferListLayer a Thunk a Thunk a Thunk a Thunk a SessionOperations a FrameBuffer a SocketTh
  2                     a BufferListLayer
   stack                a ThunkStack()
   bufferList           an OrderedCollection()
   serverProcess        a Process in [] in Delay>>wait
  3                     a Thunk
   upThunk              [closure] in EncoderThunkMaker>>makeThunk
   downThunk            [closure] in EncoderThunkMaker>>makeThunk
   headerMakerThunk     [closure] in SessionAgentMap class>>setupEncoders
   doesPop              true
   doesPush             true
  4                     a Thunk
   upThunk              [closure] in SecurityOps>>makeCustomsThunk:
   downThunk            [closure] in SecurityOps>>makeCustomsThunk:
   headerMakerThunk     [closure] in Thunk>>initialize
   doesPop              false
   doesPush             false
  5                     a Thunk
   upThunk              [closure] in [] in CipherThunkMaker>>makeThunkOn:incoming:
   downThunk            [closure] in [] in CipherThunkMaker>>makeThunkOn:incoming:
   headerMakerThunk     [closure] in [] in CipherThunkMaker>>makeThunkOn:incoming:
   doesPop              true
   doesPush             true
  6                     a Thunk
   upThunk              [closure] in SecurityOps>>makeImmigrationThunk:
   downThunk            [closure] in SecurityOps>>makeImmigrationThunk:
   headerMakerThunk     [closure] in SecurityOps>>makeImmigrationThunk:
   doesPop              true
   doesPush             true
  7                     a SessionOperations
   stack                a ThunkStack(a Transceiver a BufferListLayer a Thunk a Thunk a Thunk a Thunk a SessionOperations a FrameBuffer a SocketTh
   session              a Session
   keyMaster            nil
   currentState         State: connected
   incoming             nil
  8                     a FrameBuffer
   stack                a ThunkStack(a Transceiver a BufferListLayer a Thunk a Thunk a Thunk a Thunk a SessionOperations a FrameBuffer a SocketTh
   readFrame            nil
   bufferStream         a ReadWriteStream
  9                     a SocketThunk
   stack                a ThunkStack(a Transceiver a BufferListLayer a Thunk a Thunk a Thunk a Thunk a SessionOperations a FrameBuffer a SocketTh
   serverProcess        a Process in [] in DelayWaitTimeout>>wait
   socket               a Socket[connected]
   primitiveFailedError NotImplemented
Evaluate expressions for the current tree selection...
```

This is the protocol stack in Java.

Log

Following is a log of the 2-vat test in Java:

```
0 [main] INFO club.callistohouse.raven.presentation.AbstractPauwauTest - 2 Introducers starting
2092 [main] DEBUG club.callistohouse.utils.transport.NIOConversation - listen: 10001
2108 [main] DEBUG club.callistohouse.utils.transport.NIOConversation - listening:
callisto/127.0.1.1:10001
3208 [main] DEBUG club.callistohouse.utils.transport.NIOConversation - listen: 10002
3208 [main] DEBUG club.callistohouse.utils.transport.NIOConversation - listening:
callisto/127.0.1.1:10002
3209 [main] INFO club.callistohouse.raven.presentation.AbstractPauwauTest - 2 Introducers on the air
3505 [first-conversation] DEBUG club.callistohouse.utils.transport.NIOConversation - x_accept:
/127.0.1.1:10001 from /127.0.0.1:52938
Nov 17, 2018 11:46:34 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing Calling
Nov 17, 2018 11:46:34 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectProtocolAccepted
3816 [Time-limited test] DEBUG club.callistohouse.raven.core.RavenTerminal - sending message:
DeliverMessage(receiverId: 0 answerId: -1 resolverId: 0 action: lookupSwiss args:
[296297781521628139829099877809200071033173557439])
3817 [Time-limited test] DEBUG club.callistohouse.raven.core.RavenTerminal - sending message:
DeliverMessage(receiverId: -1 answerId: -2 resolverId: 0 action: getTheAnswer args: [])
Nov 17, 2018 11:46:34 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing Answering
Nov 17, 2018 11:46:34 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectProtocolOffered
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedProtocolOffered
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectIWant
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedProtocolAccepted
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectIAm
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedIWant
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectGiveInfo
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedIAm
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectReplyInfo
```

```
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedGiveInfo
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectGo
Nov 17, 2018 11:46:35 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedReplyInfo
Nov 17, 2018 11:46:39 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ExpectGoToo
Nov 17, 2018 11:46:40 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedGo
Nov 17, 2018 11:46:40 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing Connect
Nov 17, 2018 11:46:40 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing ReceivedGoToo
Nov 17, 2018 11:46:40 AM com.github.oxo42.stateless4j.StateMachine publicFire
INFO: Firing Connect
9686 [first-conversation] DEBUG club.callistohouse.raven.core.RavenTerminal - received message:
DeliverMessage(receiverId: 0 answerId: -1 resolverId: 0 action: lookupSwiss args:
[296297781521628139829099877809200071033173557439])
9688 [first-conversation] DEBUG club.callistohouse.raven.core.RavenTerminal - received message:
DeliverMessage(receiverId: -1 answerId: -2 resolverId: 0 action: getTheAnswer args: [])
9691 [first-vat] DEBUG club.callistohouse.raven.core.RavenTerminal - sending message:
DeliverOnlyMessage(receiverId: 1 action: value args: [GalaxyObject])
9697 [first-vat] DEBUG club.callistohouse.raven.core.RavenTerminal - sending message:
DeliverOnlyMessage(receiverId: 2 action: value args: [42])
9699 [second-conversation] DEBUG club.callistohouse.raven.core.RavenTerminal - received
message: DeliverOnlyMessage(receiverId: 1 action: value args: [FarRefImpl(hash: 14991145)])
9701 [second-conversation] DEBUG club.callistohouse.raven.core.RavenTerminal - received
message: DeliverOnlyMessage(receiverId: 2 action: value args: [42])
9710 [second-vat] DEBUG club.callistohouse.raven.handlers.RemoteHandler - GCAnswer
9711 [second-vat] DEBUG club.callistohouse.raven.core.RavenTerminal - sending message:
GCAnswerMessage(answerId: -1)
9715 [second-vat] DEBUG club.callistohouse.raven.handlers.RemoteHandler - GCAnswer
9717 [first-conversation] DEBUG club.callistohouse.raven.core.RavenTerminal - received message:
GCAnswerMessage(answerId: -1)
9717 [second-vat] DEBUG club.callistohouse.raven.core.RavenTerminal - sending message:
GCAnswerMessage(answerId: -2)
9718 [Time-limited test] INFO club.callistohouse.raven.presentation.AbstractPauwauTest - **The
value is: 42**
9719 [main] INFO club.callistohouse.raven.presentation.AbstractPauwauTest - 2 Introducers stopping
9720 [main] INFO club.callistohouse.raven.presentation.AbstractPauwauTest - 2 Introducers off the air
```

Negotiation Protocol

I designed the next version of my ParrotTalk negotiation protocol, version 3.7, a 5 message exchange to build a 256-bit AESede connection, with an user specified encoding. This is an advance from the 8 messaging version 3.6. I took inspiration from the new TLS 1.3 message exchange, which is a 3 message exchange.

I decided to keep the 2 message protocol negotiation messages, ProtocolOffered and ProtocolAccepted. This allows one to specify in the SessionAgentMap, the versions supported, v3.6 and v3.7, as well as the preferred protocol version to use.

The first two msgs (ProtocolOffered/ProtocolAccepted) are negotiating which protocol version to use, so I added a class SessionProtocolSelector which handles these two methods. I have the option of negotiating ParrotTalk-3.6 or ParrotTalk-3.7. Based upon which protocol version is negotiated I manipulate the stack to pop the ProtocolSelector and push the SessionOperations for each protocol version. I split the stateMap to handle the Protocol messages in the Selector and the handshake traffic for each version in the specified SessionOperations. v3.6 uses the current SessionOperations minus the protocol negotiation states which moved over to the selector.

Due to using an ASN1 specification for headers, it will be possible to specify the protocol version and skip the protocol negotiation messages. Therefore, the v3.6 becomes a 3-message handshake, starting with Hello_v3_7 as the first message. The receiving side willl have a SessionProtocolSelector and be able to discern the use of v3.7 from the initial Hello_v3_7 header and skip the protocol negotiation and install the SessionOperations_v3_7 and initiate the answer state machine for v3.7.

The new SessionOperaations_v3_7 will have an updated stateMap to handle the 3-way msg exchange (Hello_v3_7, Response_v3_7 and Signature_v3_7). Thus after I add the protocol version to the AgentMap, the SessionAgent will install the Selector with the specified protocol version and when 3.7 or 3.6 is negotiated the stack will be thunked with the correct SessionOperations and the SecurityOps from the selector (to capture the local and remote traffic for signatures) will be installed in the protocol SessionOperations. Since the SessionOperations holds the stateMap and all of the process and send methods for the version of the protocol, the only difference between the two protocol versions is the specific SessionOperations.

The SecurityOps and all other thunks are the same between each version. That's freaking cool, I think. That just that one Operations thunk is the difference between versions. If I decide to implement SSL on the ThunKStack, the other change will need to be the ReceivingFrameBuffer, since the SSL frames are fundamentally different from ParrotTalk's frames. Ditto with SSH. Anyways, I needed to share my thoughts on the changes I am making. Thanks for your consideration! Thanks for reading my ramblings!