Release Notes for the Private Messenger from Stephen Charles Tassio

Version 1.2.0.1

Prerequisites for these start up tasks:

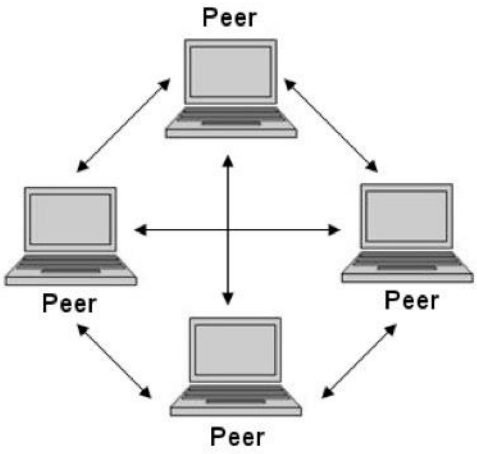
1. ReadMeFirst for SDC Private Messenger 1.2.0.0.pdf
2. EULA.pdf
3. Prerequisites for Private Messenger 1.2.0.0.pdf
4. Generating and Distributing Encryption Keys1.2.0.0.pdf

Story:

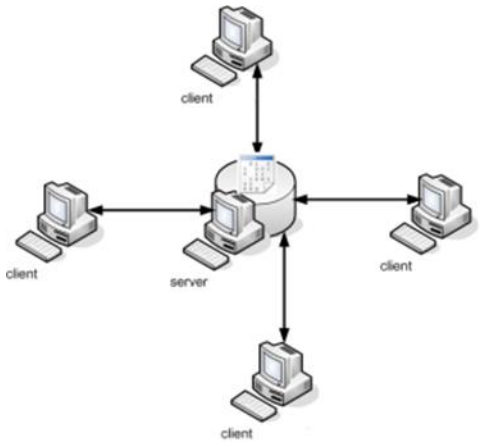
The Private Messenger is an instant messenger and due to its security, privacy and untraceable nature the end users are required to supply encryption keys and other data before each use. In this manner no uninvited person has access to or even knowledge of the chat sessions. No third party server is required all that is needed is two or more people with activated instances of the Private Messenger wishing to communicate, something like a cell phone conversation.

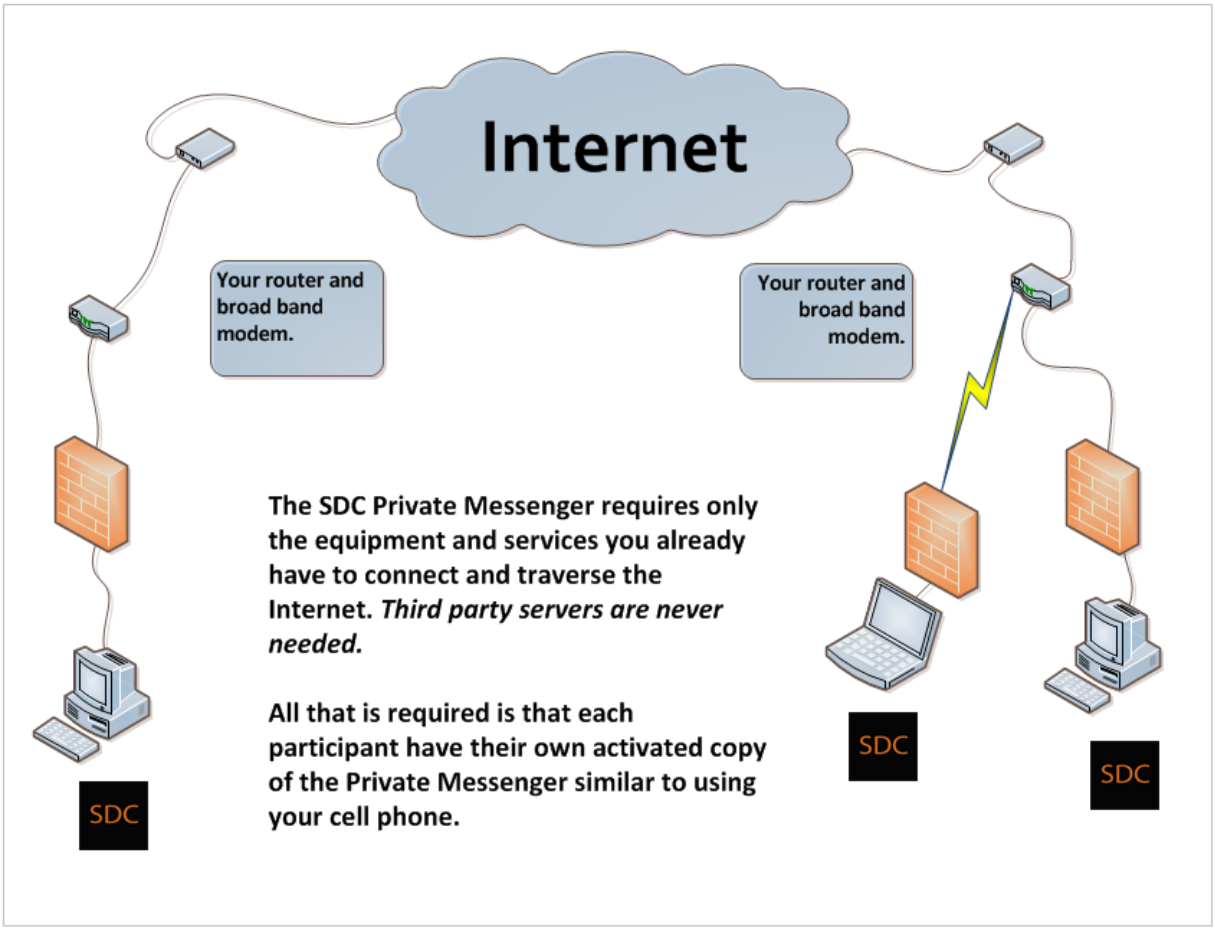
This form of computer interaction is known as 'Peer-to-Peer' as opposed to the more common 'Client-Server' topology.

Peer to Peer



Client Server





All chatting is done point to point. That is in a group of three people, all three might be in a Group Chat but a sender has at all times the option to send to one or all participants. Every chat message sent is direct to the end user from the sender.

When the Private Messenger is started up (after being initially activated and 'Preferences' values entered) it will determine the current external IP address of the computer in use which requires Internet access to accomplish. Alternately, the user can enter his current external IP address in the 'Preferences' tab. This supplied information is stored but the user is free to delete it after use.

During a Group Chat session one of the participants must elect to be the 'Resolver'. This means an additional view is opened in his instance called 'Resolver' that hosts a socket server and runtime memory management for his and all the other participants Messengers.

The person hosting the Resolver has made his current external IP and open Resolver port number known to the other participants. This information is stored by the 'Preferences' view in the participants Messenger but can be deleted at the end of each chat session. When users start their Listener view (including the Private Messenger hosting the Resolver) the Listener contacts the Resolver with its current IP address and port number that it is using for chats. This information is encrypted and is never decrypted except on each users Messenger when a 'Resolve' is performed and all currently 'Registered' participants information is downloaded to their instance.

The Private Messenger periodically refreshes its registration with the Resolver and the Resolver periodically cleans its memory of old registrations. In this way new people joining will not think a departed participant is still in the group in the event the departing persons Messenger did not successfully 'Unregister'. There is no broadcasting of presence or availability. A Messenger is either 'Registered' or it is not.

There is a Quick Chat mode for two people only:

1. In this mode the Resolver does not need to be started up with the participants Registering as before.
2. Both participants communicate their current external IP address and available port number and key information to the other. The information is entered and chatting can commence.

In both Group and Quick modes there are no connections maintained between participants. Every message sent creates a new connection. When it has been delivered the connection is closed. Participants in a chat session can be away from their instance for any length of time. In the Group mode it must be possible for the admin functions to maintain themselves so it must always be possible to make a connection to the Resolver but admin connections are not left in the connected state the same as with chatting.

IMPORTANT:

The current version of the Private Messenger is not designed for 'coffee shop' free Wi-Fi. These sites are configured as their owner pleases which usually means obtaining inbound connections is unpredictable. Using an ISP provided connection is currently the only supported Internet access type.

The encryption used by the Private Messenger is known as 256 bit AES. This means the keys used are 256 bits long. They have to be entered (copy paste ok) each time the Private Messenger is started. One key is for the Messenger's admin functions and the other for the users message contents. The keys are in this format:

e3 41 69 3a 99 57 b3 04 63 b3 48 af b4 85 8b 25 27 cc a0 6a c7 4d 3f d3 d9 a5 a1 7d 90 1c f3 6e

The above numbering system is known as hexadecimal (16). Computers work in binary 1's and 0's known as bits. To make this human readable the digits 0-9 are used as well as the letters a-f or A-F. This allows for all 256 combinations of bits in an 8 bit byte to be represented. From the example first byte above 'e3' equals '1110 0011' in binary bit representation that computers use.

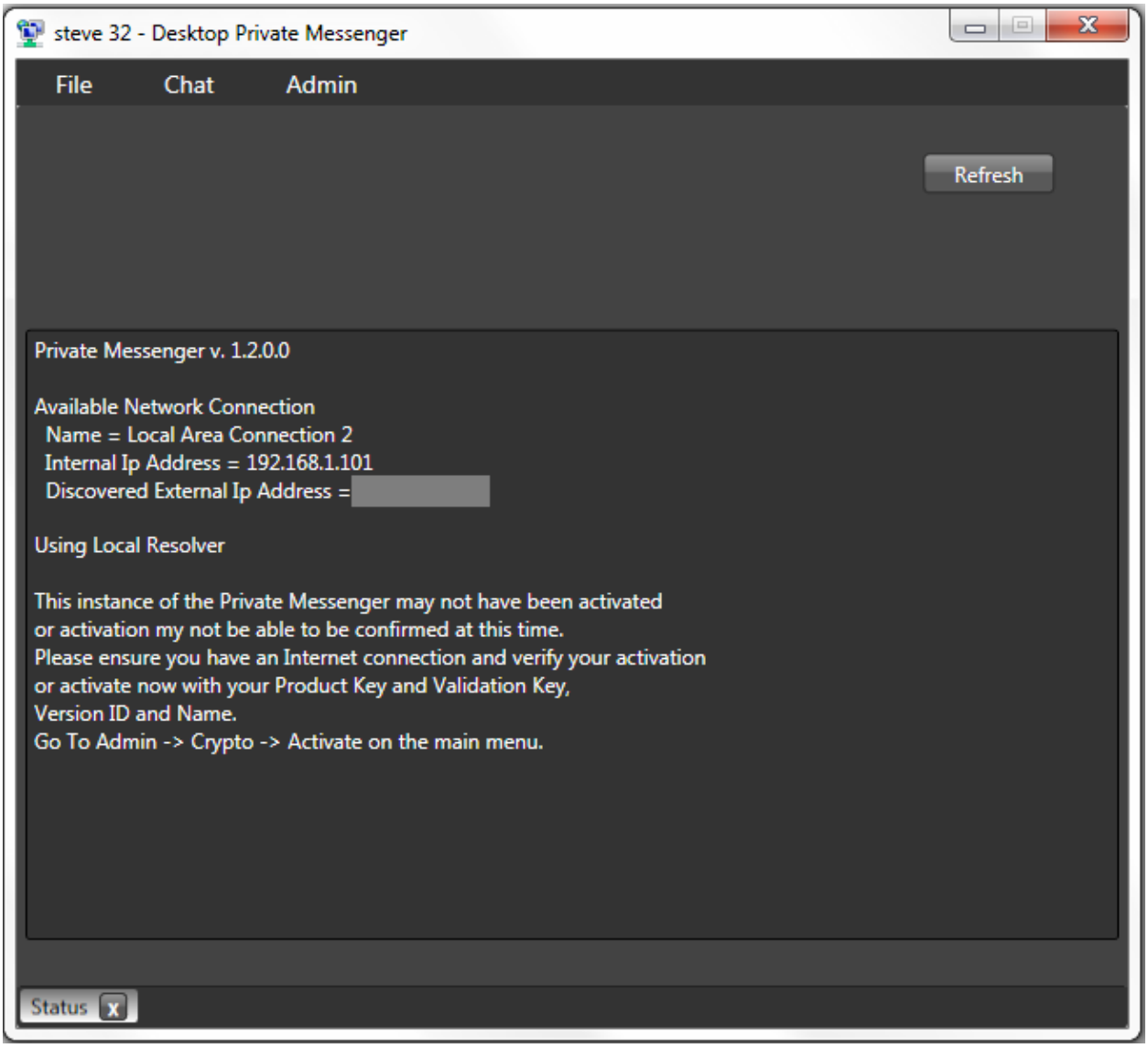
In AES there are also Initialization Vectors in this format:

7a 02 87 97 11 13 38 13 db 70 72 de cd bc 54 9c

The user can supply his own keys and IV's or one of the participants in the group can use the Private Messenger to generate them and communicate them to the other participants. The Private Messenger uses Microsoft FIPS certified crypto libraries to generate the keys and IV's. See the included 'Generating and Distributing Encryption Keys.pdf' for more details on keys.

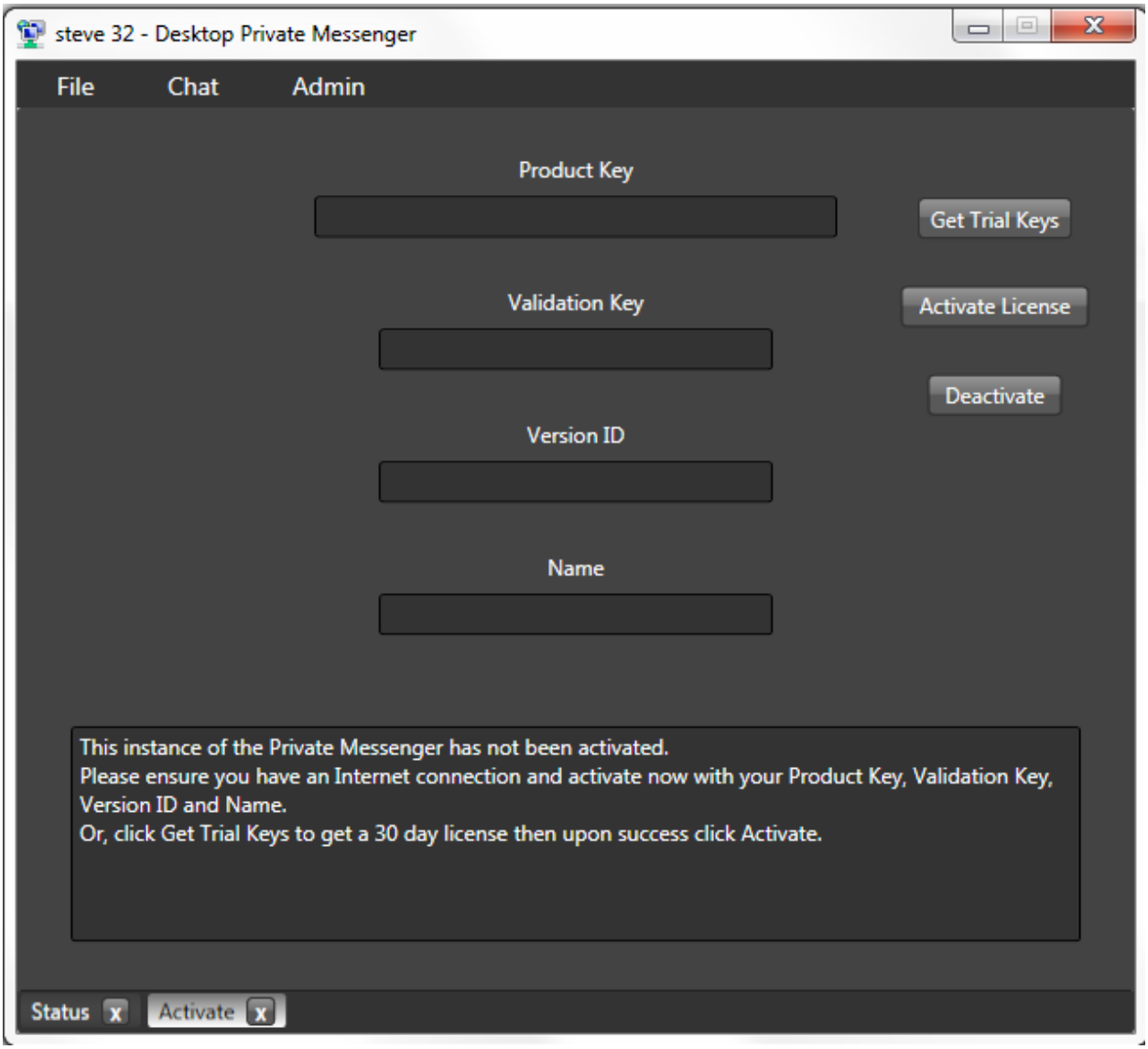
In this release the Private Messenger has the ability to electronically distribute the keys needed for the very first communications after install and activation as well as securely change session keys during an established chat session. See Key Distribution Scenarios appendix for details.

The next thing to do after installation and reading this is to start up the Private Messenger and you will see something similar to this:



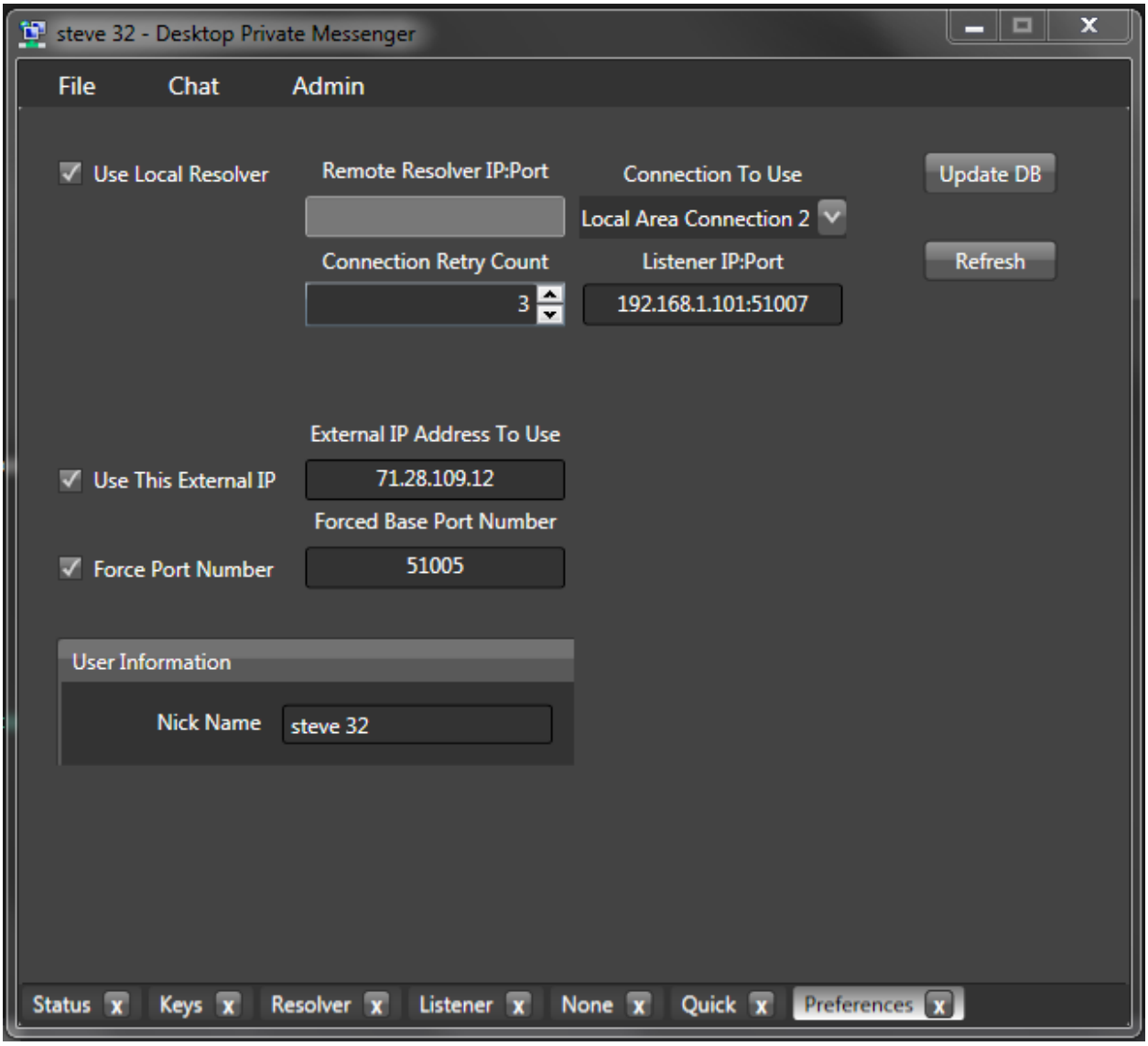
Notice the text. The current version is displayed and Internet connection has been confirmed. If the 'Internet Check' indicates a failure then you must stop and rectify the Internet access problem before attempting Activation. The main point of interest is that Product Key activation is in doubt. This is normal for the first time. Follow the instructions and navigate to the Activation View via the 'Admin' menu item.

On the ‘Activate’ view there are fields to input licensing information:



Place the information from your license purchase confirmation email into the correct places then click the ‘Activate’ button. (Trial version keys are obtained by clicking ‘Get Trial Keys’.)This should result in a successful activation of your instance of the Private Messenger after a few moments. It is now licensed for use as described in the EULA. Be sure to keep the information in your email. The Deactivation functionality allows the Private Messenger to be installed and reactivated on another computer. A successful Deactivation is required before moving to another computer. Attempting to Activate an already activated key could result in the first Activation being Deactivated and/or the current Activation attempt could be refused. At this point going to the Admin -> Preferences view will allow the user to enter the displayed ‘Nick Name’, whether using local or remote Resolver etc.

Preferences view:



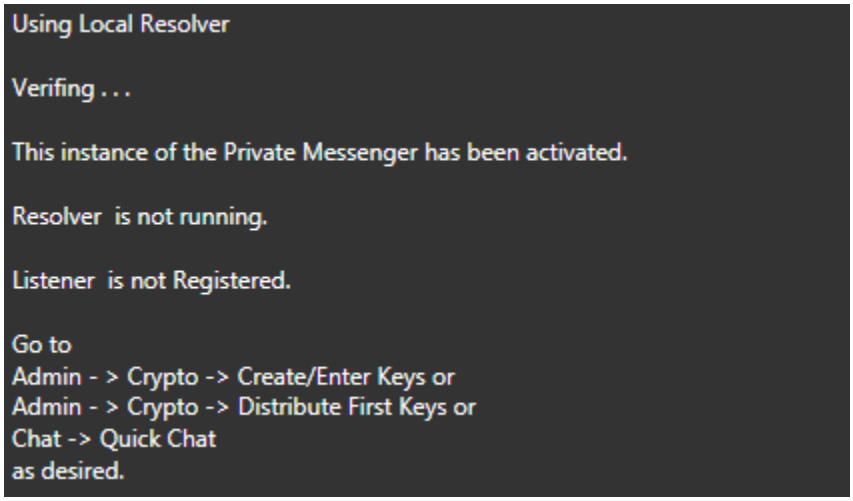
Only a few basic items are persisted for convenience. Connection Retry Count is a global value. It specifies how many times any connection is attempted before abandoning the effort. This includes checking for Internet connection, obtaining external IP address, activation, deactivation, chatting etc. If you are unsure of the reliability of your Internet connectivity set it to 3 or higher because the Private Messenger will cease retrying as soon as the connection is established.

Once the desired information is entered click ‘Update DB’ to save the values then click ‘Refresh’. The display shouldn’t change which means the values entered are in the database.

After this operation going back to the Status tab and clicking ‘Refresh’ the Private Messenger will reload all of the admin information with the new values. The Listener IP:Port value is read only and is informative and not configurable as the Messenger will determine the values thereof.

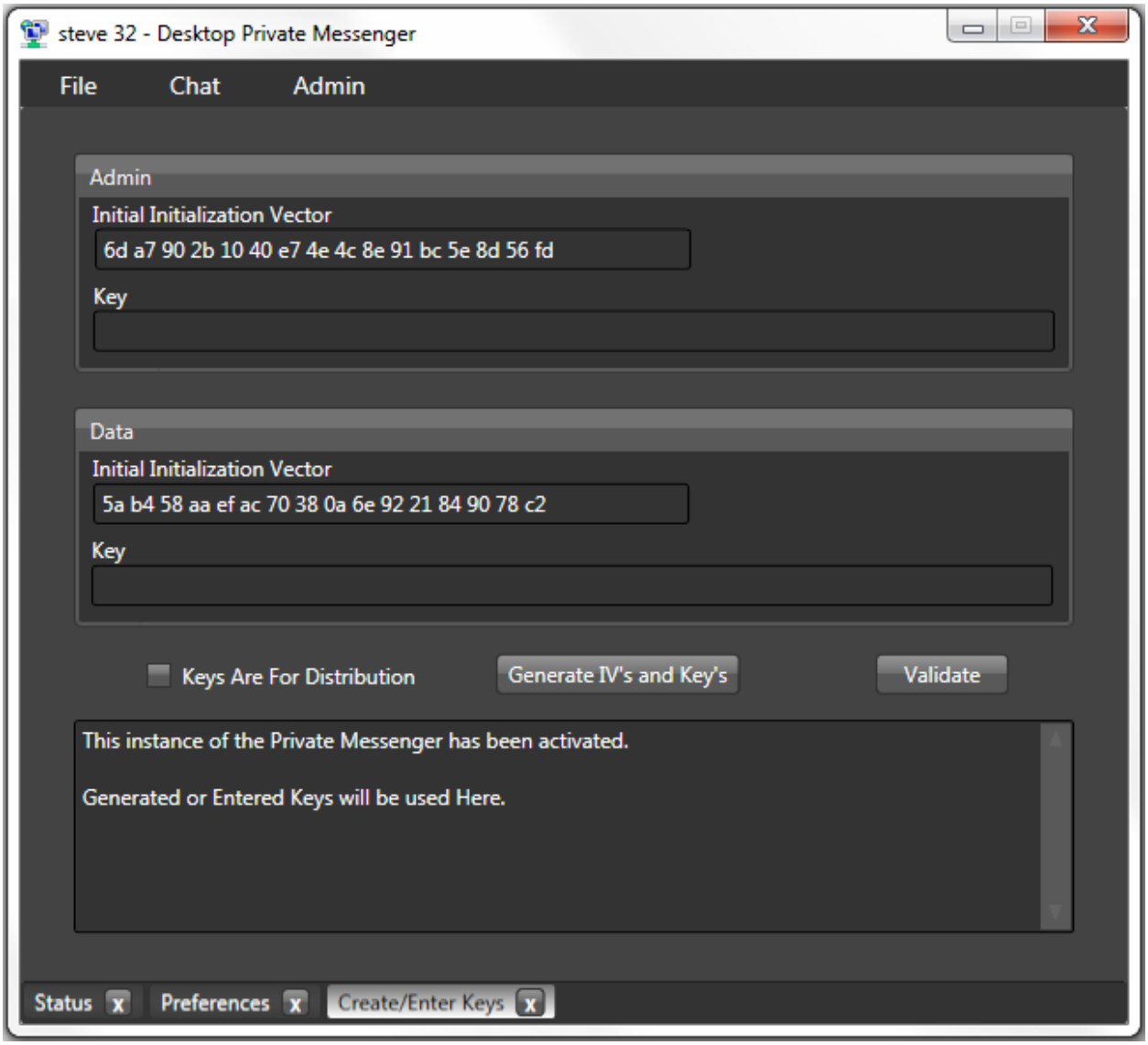
Once the Private Messenger is licensed the next time it is started the Status view will show. Each time the Messenger is started your keys must be entered in one of the manners presented as no sensitive information is stored. There is no ‘speed’ or ‘voice’ dial type of feature for the keys available to ensure maximum control of these items by the user.

Partial Status view after Activation:



There are now 3 options for users to use their desired keys. From this point Create/Enter Keys or Distribute Keys is available.

Keys view:



Notice that the Private Messenger will supply you with cryptographically strong Initial IV's. You are not required to use these and can alter or replace them as you desire. Note that for each subsequent chat message sent a new IV is automatically generated.

If you are planning to update your current keys during this session the first enter and validate your existing keys as normal. Then click 'Keys Are For Distribution'. Then Create/Enter your new keys and Validate. If successful your new keys and IV's are stored for later distribution.

Validation ensures the correct form of the IV's and keys only it cannot determine if the values are what you should use. A message will appear in the status display and if all is well you may proceed as indicated. If you are hosting the Resolver you will go to the Resolver view.

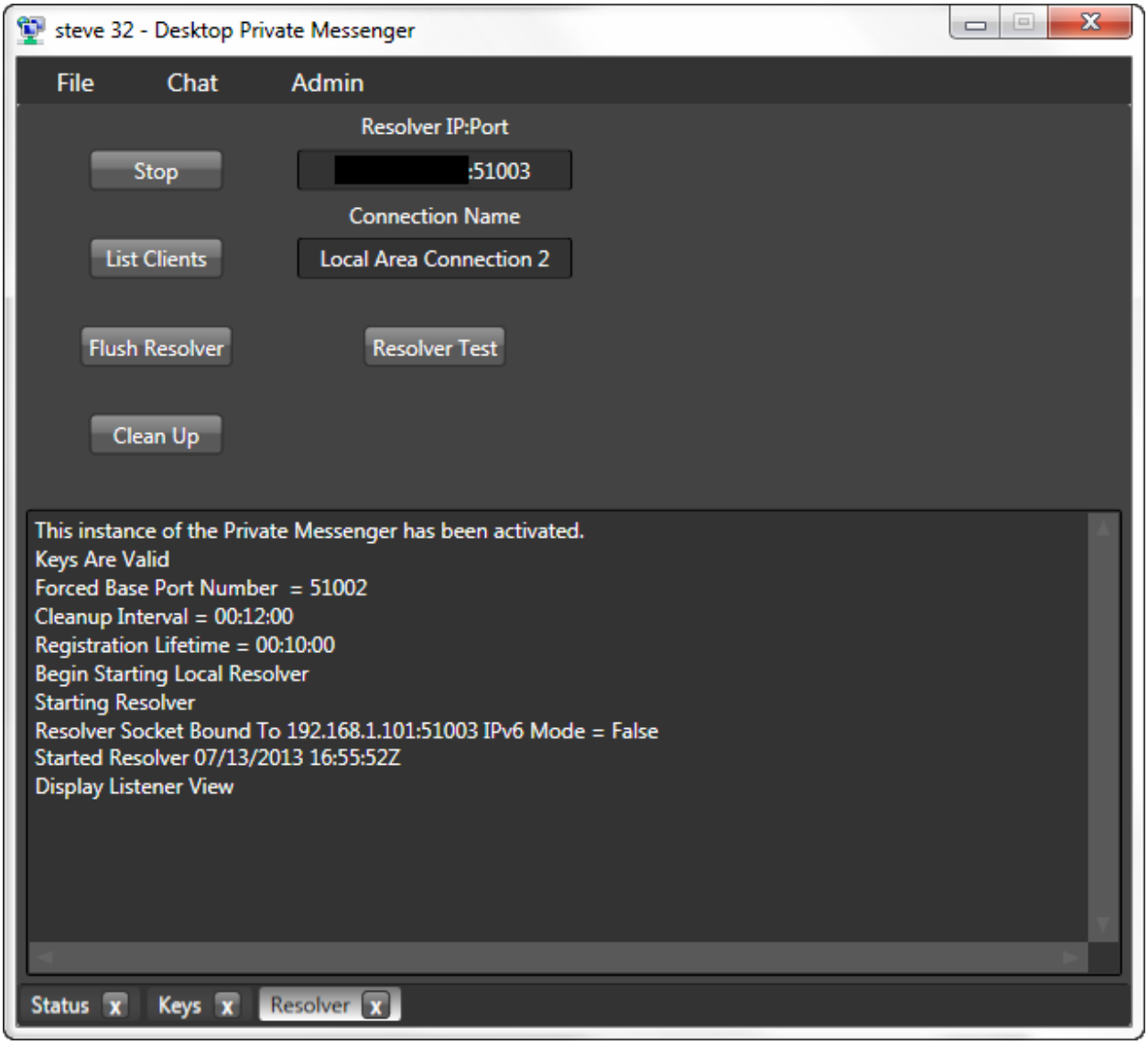
Actual chat:

User sends: "What did you think of that movie last night?"

Sniffed ‘off of the wire’: J8YjShCizXur7EdQ/rIltYJkogr5ERWYXPDEqxj91Ff/FmmArVnWzB9ske8uvUnk

Recipient reads: "What did you think of that movie last night?"

Resolver view:

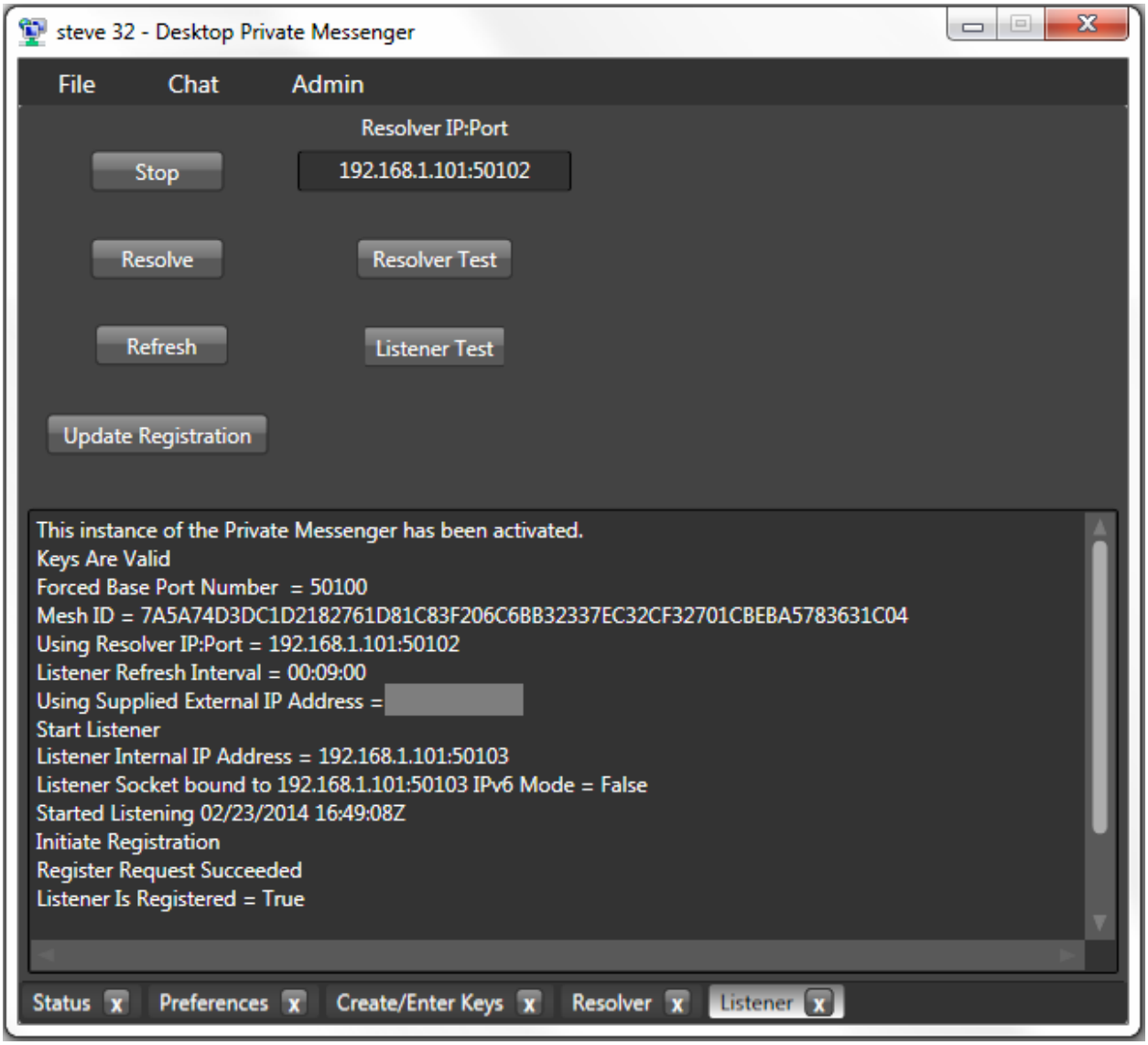


The connection information is from the information discovered at start up and is your current external

IP address and available port.

Follow the instructions and display the Listener View in the Admin menu which is the view that does the actual chat listening.

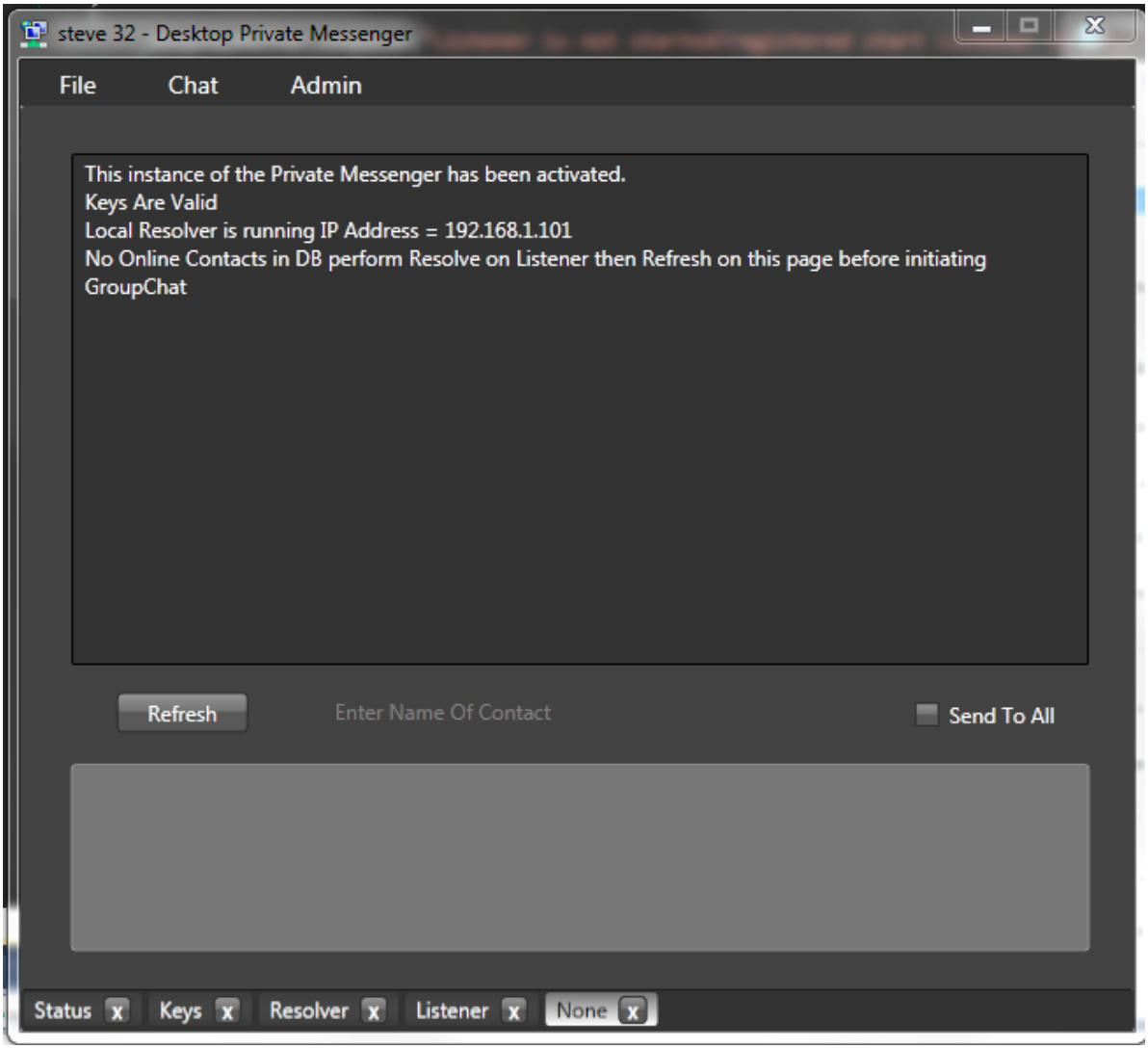
Listener view:



The Listener is now ready for Group Chats. Notice that this Listener is 'Registered'. Even though this instance of the Private Messenger is hosting the Resolver for the Group the Listener has no knowledge of that and simply registered its information to the supplied IP address and port number. As much as possible in a single application there is a complete separation of concerns between admin and message content functionality.

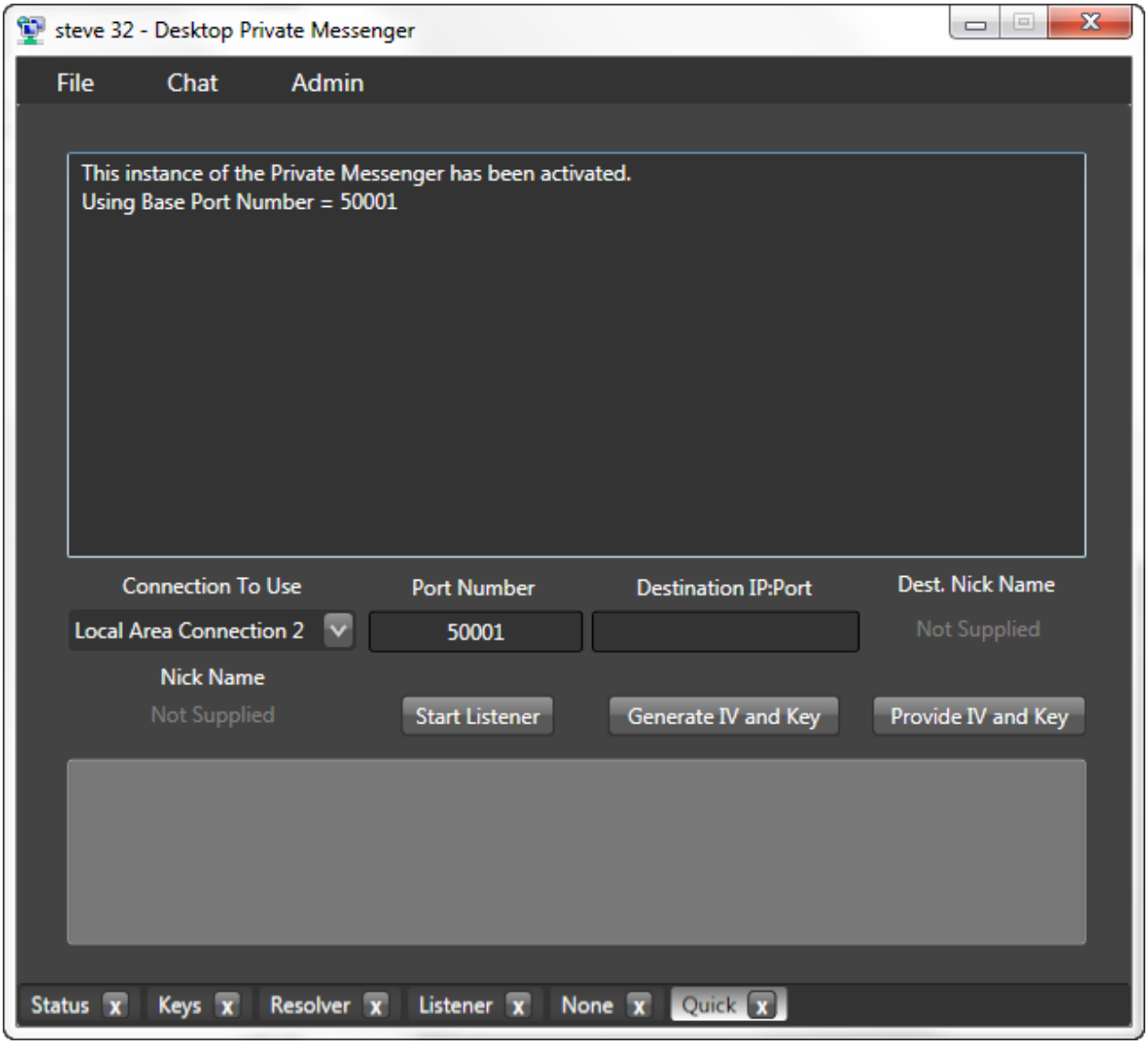
The Private Messenger is now ready for Secure, Private and Untraceable instant messaging.

Group Chat view:



This is a typical instant messenger layout. Status and messages are displayed in the top Display and chat data to send is in the bottom. Sending is via the enter key and newline is shift-enter. If there were one or more participants Registered in the Resolver then their screen names would appear in a list in the top Display. The 'Enter Name Of Contact' ComboBox has this list just select who you wish to send to next. Optionally check 'Send To All'. The next inbound chat will change the contact name so a quick response to that participant can be made. Because of the disconnected nature of the Private Messenger participants can come and go as desired or by an unplanned disconnection.

Quick Chat view:



In this mode no Resolver is needed. This is a normal chat type of view except the connection and key information is added manually. Select which NIC card to use if needed then start the Listener. Key and IV values along current external IP Address and port number must be communicated to remote party as with Group Chat. Notice on the tabs at the bottom of the window this instance can still be hosting the Resolver and engaged in a Group Chat while Quick Chatting to someone who may not even be a Group Chat participant.

IMPORTANT:

Encryption key distribution scenarios:

With this release there are currently 3 methods of distributing keys. The reader should be familiar with Messenger operation and have an Activated instance running.

1. Physically deliver by any method or combination of methods the keys to employ in the chat sessions to each user.
2. Use the new Distribute First Keys functionality to create and securely distribute new keys between users who have nothing other than a running instance of an activated Messenger.
3. Use the new Distribute New Keys functionality to securely distribute new keys to users who are currently online in a chat session employing keys from either method 1 or 2.

Method 1:

This is perhaps the simplest method and is a very secure way of distributing secret keys. This was the method used for many years by governments, financial institutions, corporations and others. The extreme simplicity of this method is also its main drawback. Physically distributing keys on a regular basis is inconvenient and can be expensive. However, those attributes do not diminish the effectiveness of the method even today.

The users agree on a person to generate and distribute the keys to use by meeting, a drop site, overnight mail etc. Keys physically written down can be protected by a one-time pad previously exchanged rendering the loss of the keys merely an inconvenience.

Method 2:

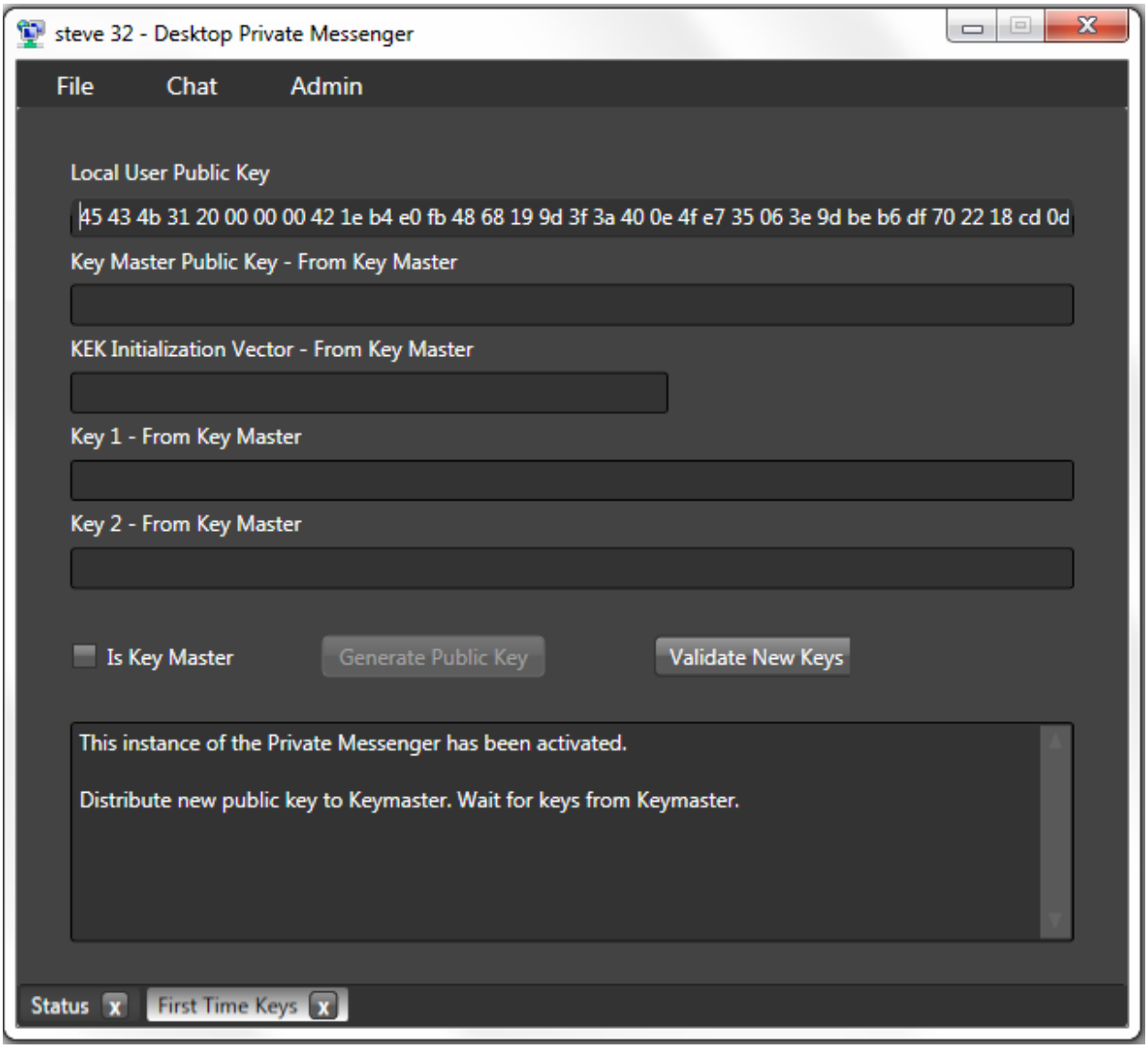
For users not desiring to meet to obtain their first time keys this method allows them have their activated Messengers running and an Internet connection and electronically distribute their first time keys. Physical location is not a concern. The challenge is to generate and securely distribute secret

encryption keys without any prior key exchange having occurred. The solution to overcoming physical distribution was invented by researcher's Diffie and Hellman in 1975.

Without the math each Messenger user has inbuilt functionality that will create two keys for each user. One is a public key the other a private key. Although not to be confused with RSA the keys are used as their name implies. Each user in the chat session exchanges his public key with the designated user called the Key Master. It is not required that user who is hosting the Resolver be the Key Master although he can be. The Key Master will reply with his public key and the new session keys he created and encrypted with the key derived from the users public key.

The user applies this information into his Messenger and using the Diffie-Hellman functionality computes a decryption key from the Key Master's public key. This locally computed key has the same value as the one the Key Master created with the user's public key. Neither of these derived keys have seen the light of day or traversed the Internet or even been stored anywhere. It is created, used once then vanishes for all time. That key is to encrypt and decrypt the new session keys which can be sent over the Internet securely. Once the first keys are Validated and stored for this use on each Messenger each participant Registers his Messengers as before and chatting can begin.

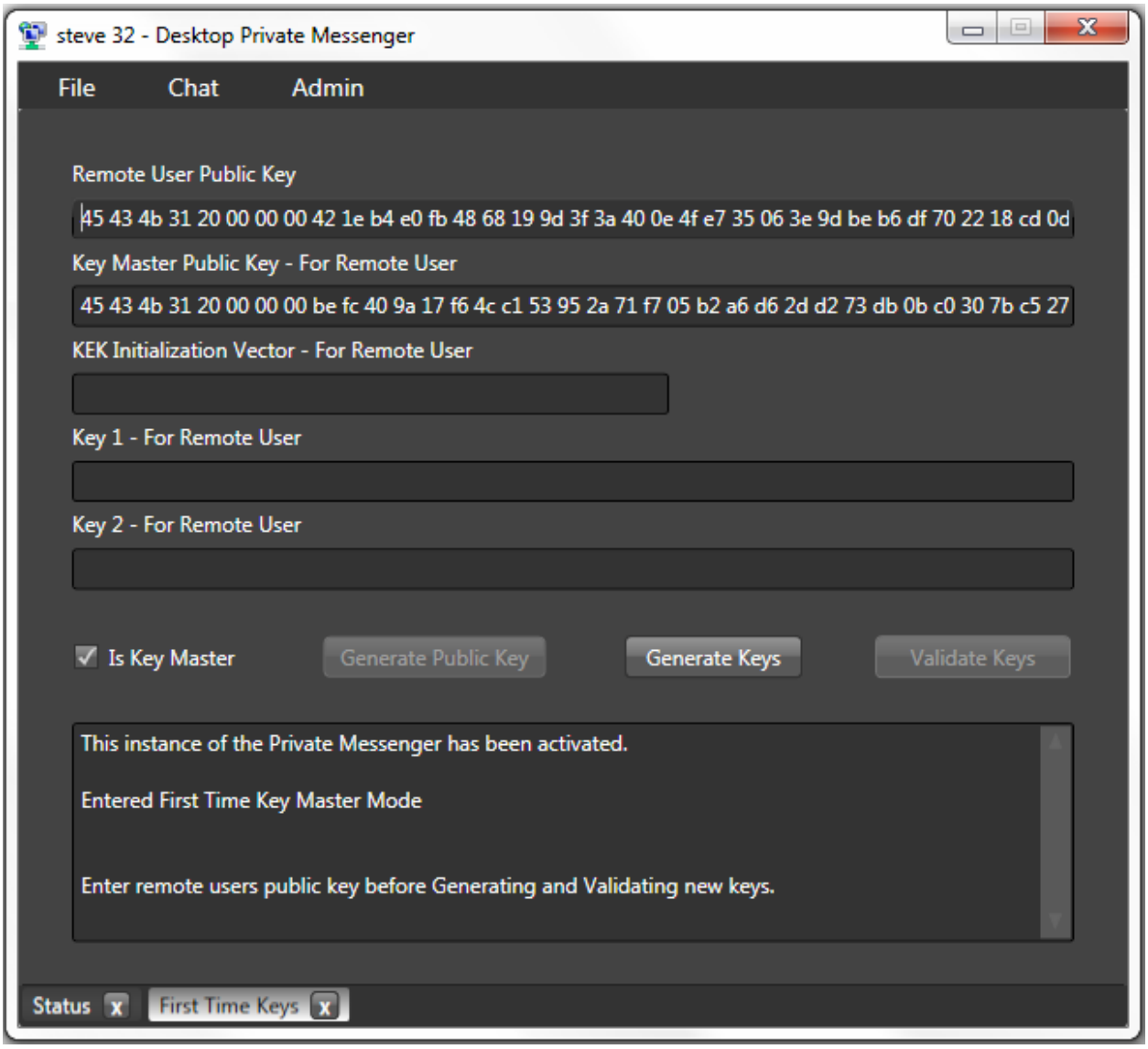
Even though the users exchange their public keys that contain no secrets or data any public key is subject to man in the middle attacks. This means Mallet can intercept public keys switch them out and be in the middle of the conversation. To eliminate that threat with the First Time Key functionality and no prior established secure communications channel the users can communicate their locally generated public over the phone as it has been converted from a long binary sequence of bytes into human readable form by the Messenger. Got to Admin -> Crypto -> Distribute First Keys.



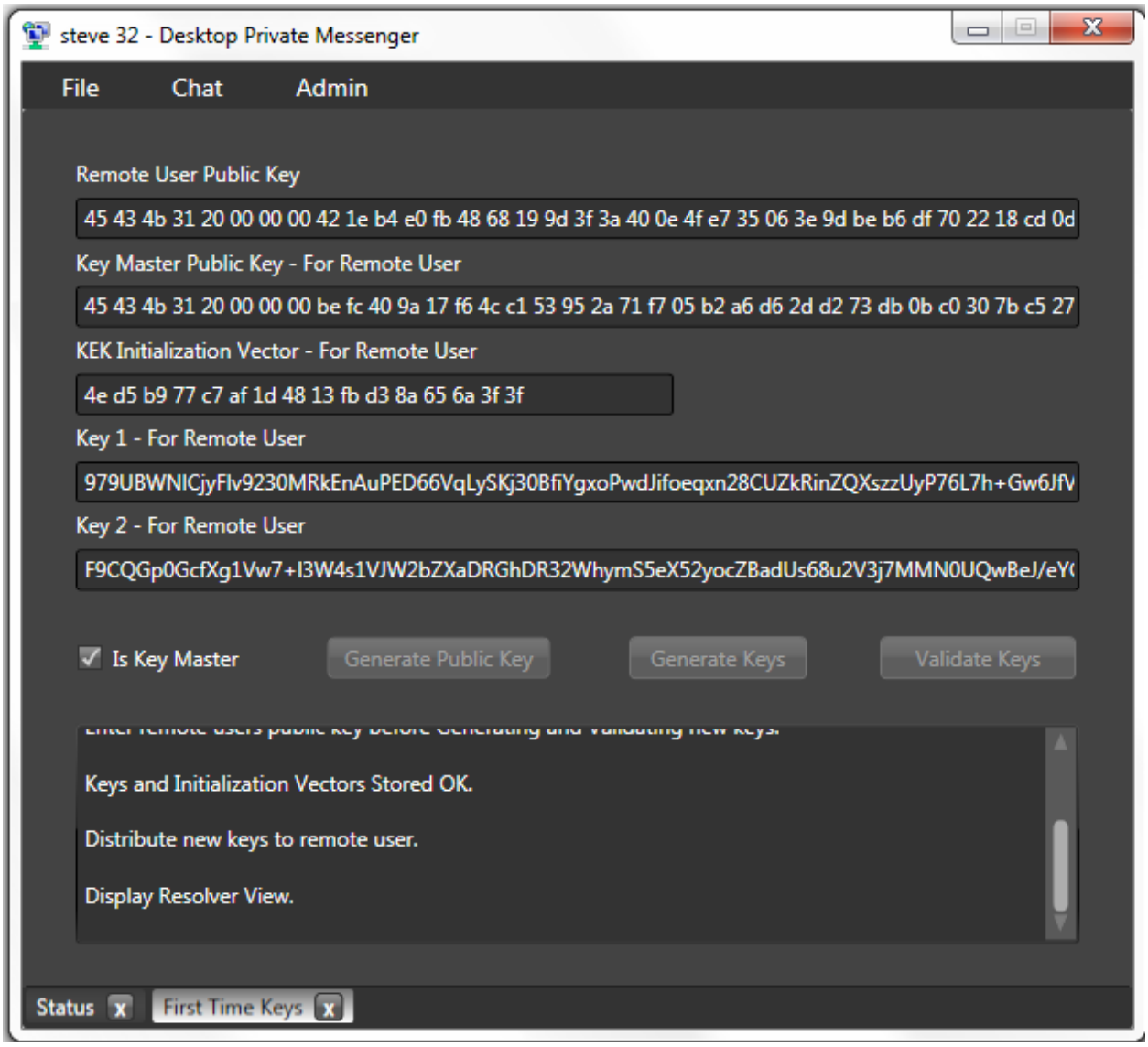
Notice that this view was selected immediately after startup. Notice 'Is Key Master' is not checked. This tells the Messenger functionality to behave as an ordinary group user. Note also that the 'Generate Public Key' button is grayed. It has been clicked once to generate the public key seen above. The public key is longer than the TextBox displaying it. For convenience the public key could be copied into a NotePad or other blank document and broken up into smaller pieces before being read over the phone:

45 43 4b 31 20 00 00 00 42 1e b4 e0 fb 48 68 19
9d 3f 3a 40 0e 4f e7 35 06 3e 9d be b6 df 70 22
18 cd 0d c9 2b 33 73 6e ee 83 a0 38 cd 1d 02 55
6c 93 42 3d 07 b8 31 1e bb e4 4e a4 4d f3 01 0c
3b e4 81 a0 1c b0 b6 ad

This improves accuracy which is paramount and guarantees the integrity of the key which becomes useless to Mallet as soon as the Key Master enters it into his Messenger. The Key Master enters the user's key then generates his own:



Notice the Key Master has generated his public key which can be repeated over the phone as was the user's public key. By clicking the 'Generate Keys' button then the 'Validate Keys' button the Key Master loads the new validated keys into his Messenger for use while leaving them displayed here to send to the user in question which because they are encrypted can be by regular email. The Key Master must repeat this process by restarting the First Time Keys View for each user no information can be reused.

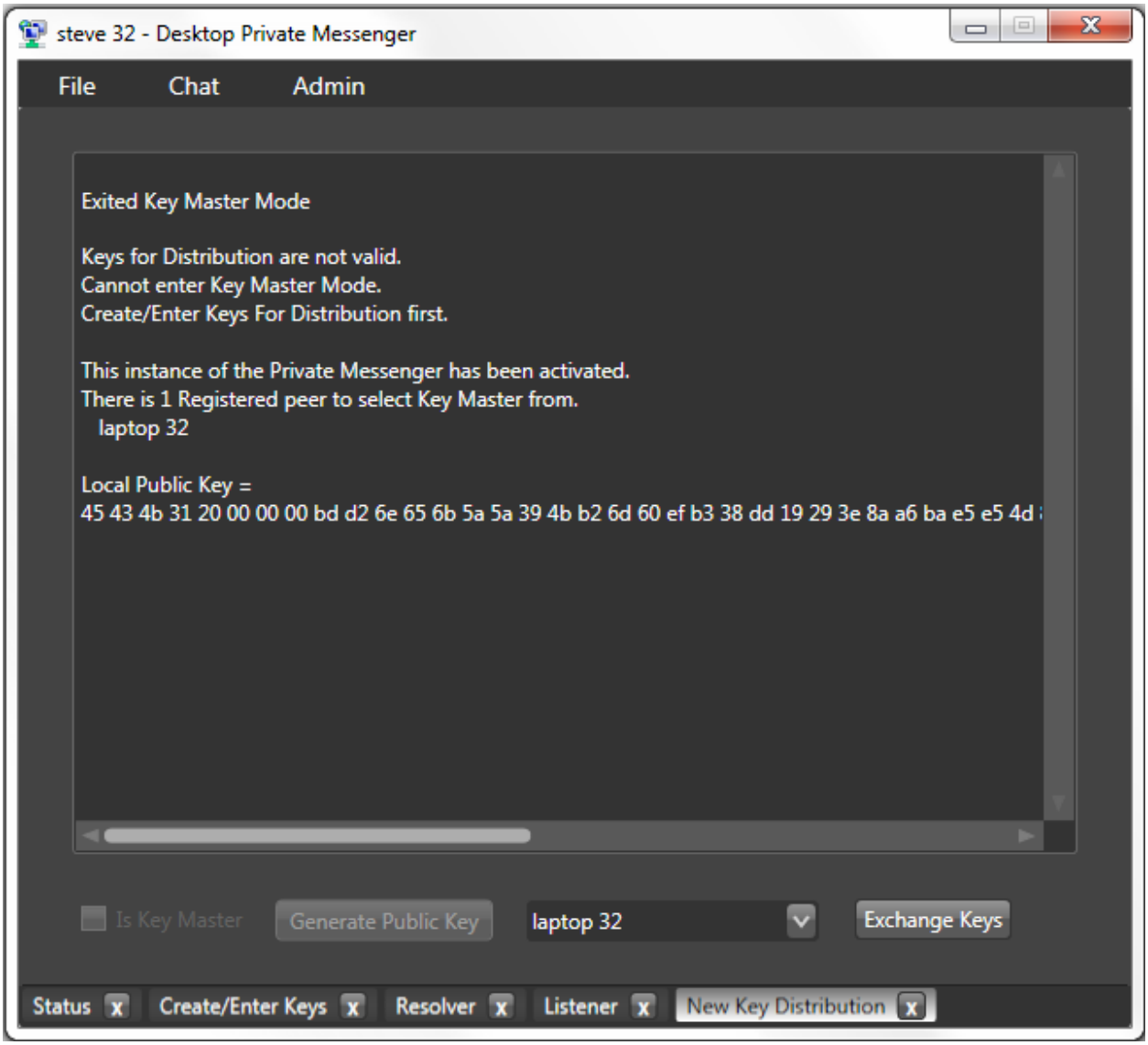


The KEK Initialization Vector, Key 1 and Key 2 can be copied and sent by email or any other method. Note that Key 1 and Key 2 are longer than is displayed too. There is nothing stopping this method from being used for every chat session thereby completely eliminating key storage and protection issues. After the user clicks ‘Validate New Keys’ and it completes then he can continue to the Resolver or Listener as desired.

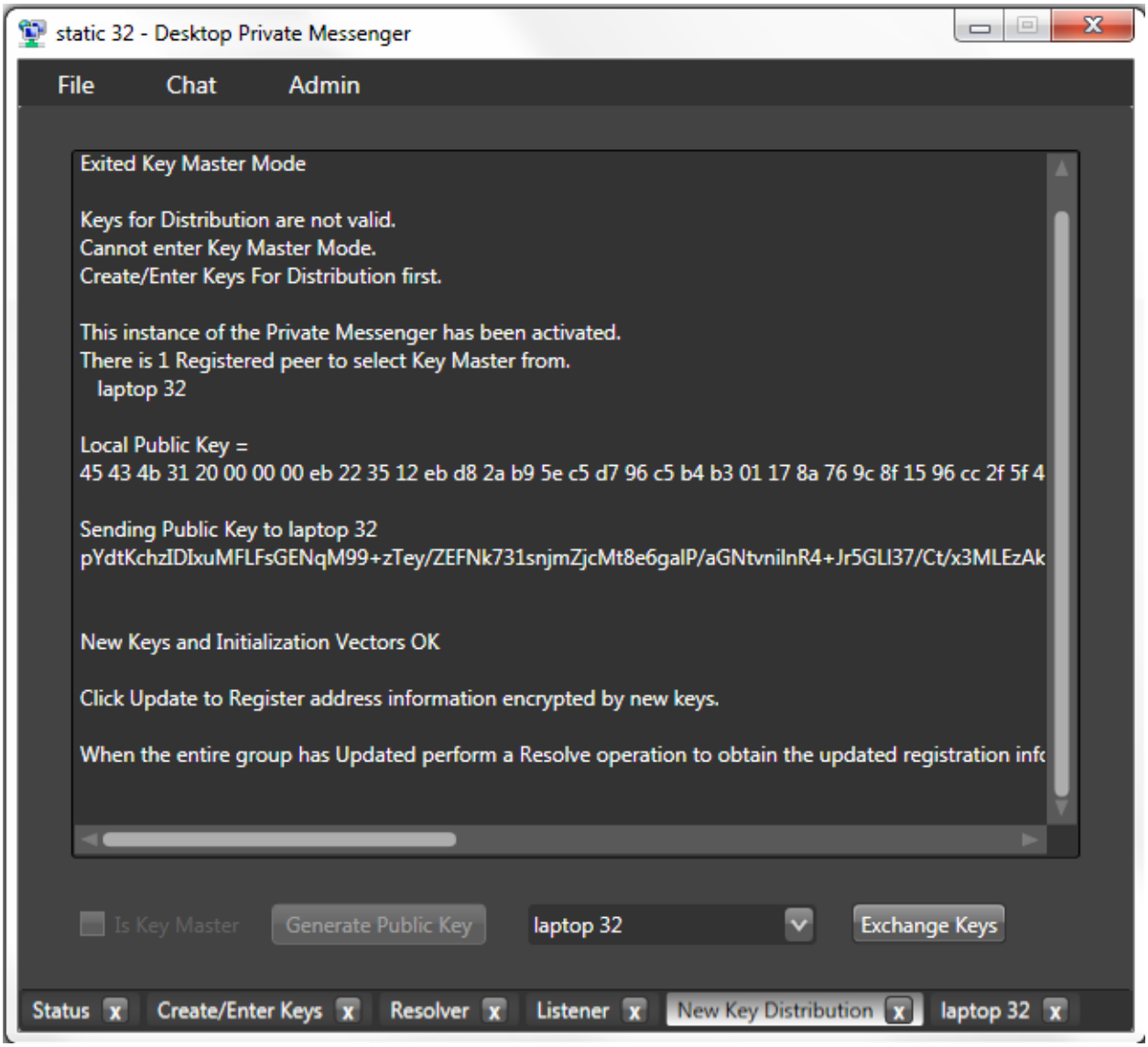
Method 3:

This method allows users who have already established a secure chat group via Method 1 or Method 2 that wish to change their session keys while online. All chat windows must be closed but Listener and Resolvers left running and connected. The user then selects Admin -> Crypto -> Distribute New Keys. There are 2 roles Key Master and user. One Key Master per group as with Resolver host.

User role:



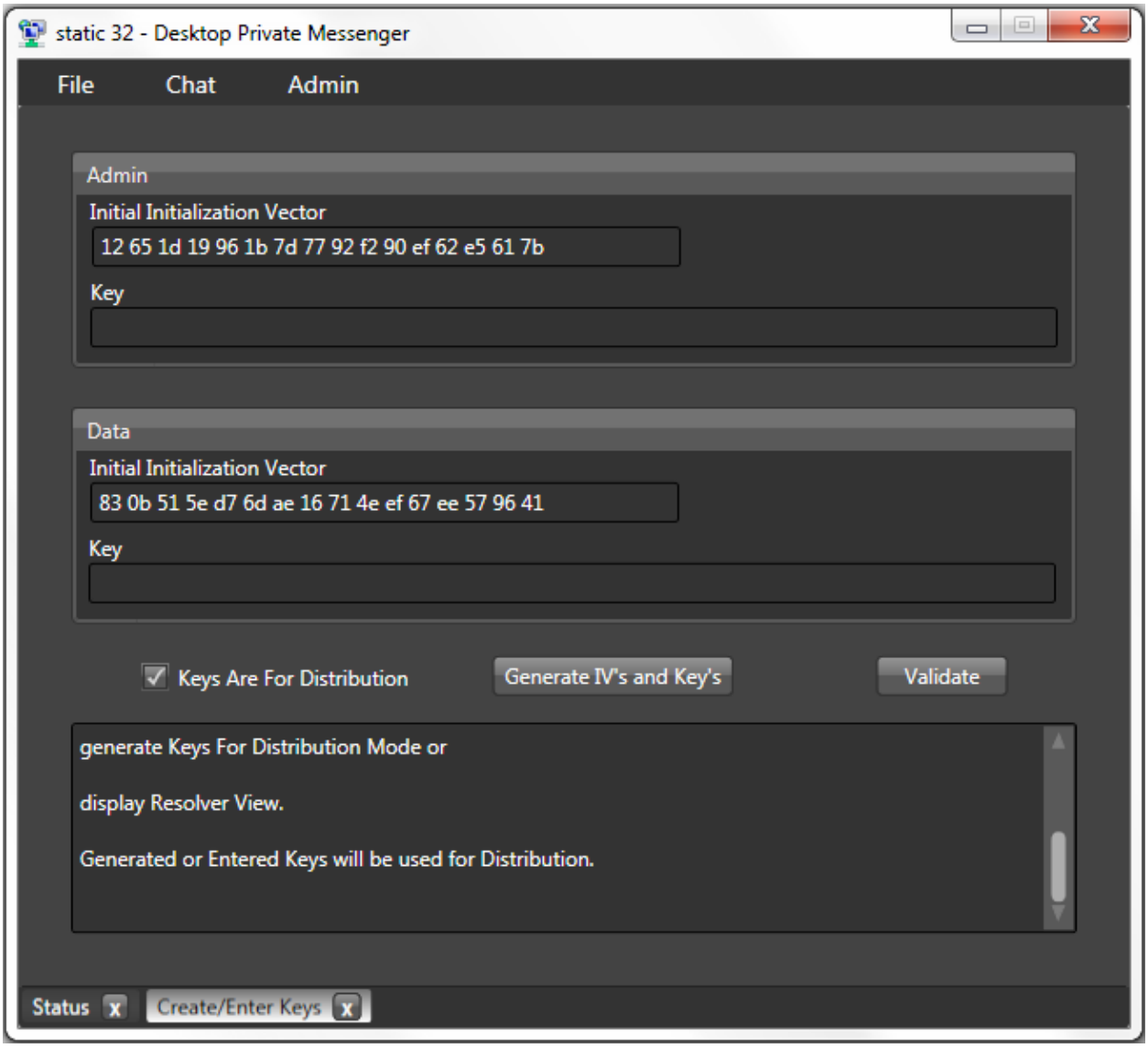
Note the user is online and Registered with at least one other one of whom must be the designated Key Master. Here the user has generated his public key, selected the Key Master and clicked ‘Exchange Keys’. The user waits here until the Key Master’s Messengers signal’s the arrival of the new keys.



When a user sees the return info from the Key Master he goes to the Listener view where this info is repeated and clicks ‘Update Registration’. The Key Master does this only once not for every user he distributes keys to. After all users are Updated everyone must perform a Resolve on the Listener view to get the user data encrypted with the new keys so chatting can continue.

Key Master Role:

This person becomes the center of attention like the Resolver host although for only a brief period of time. Just as with the Resolver starting first so too the Key Master must have his Messenger configured before the rest of the users start their key change process. The Key Master starts up his Messenger, goes to the Create/Enter Keys view as before and enters his copy the session keys currently being used by the group. This is the point at which the process changes from the users. After entering the current keys and validating them the keys TextBoxes are cleared for security as before and now a little convenience.

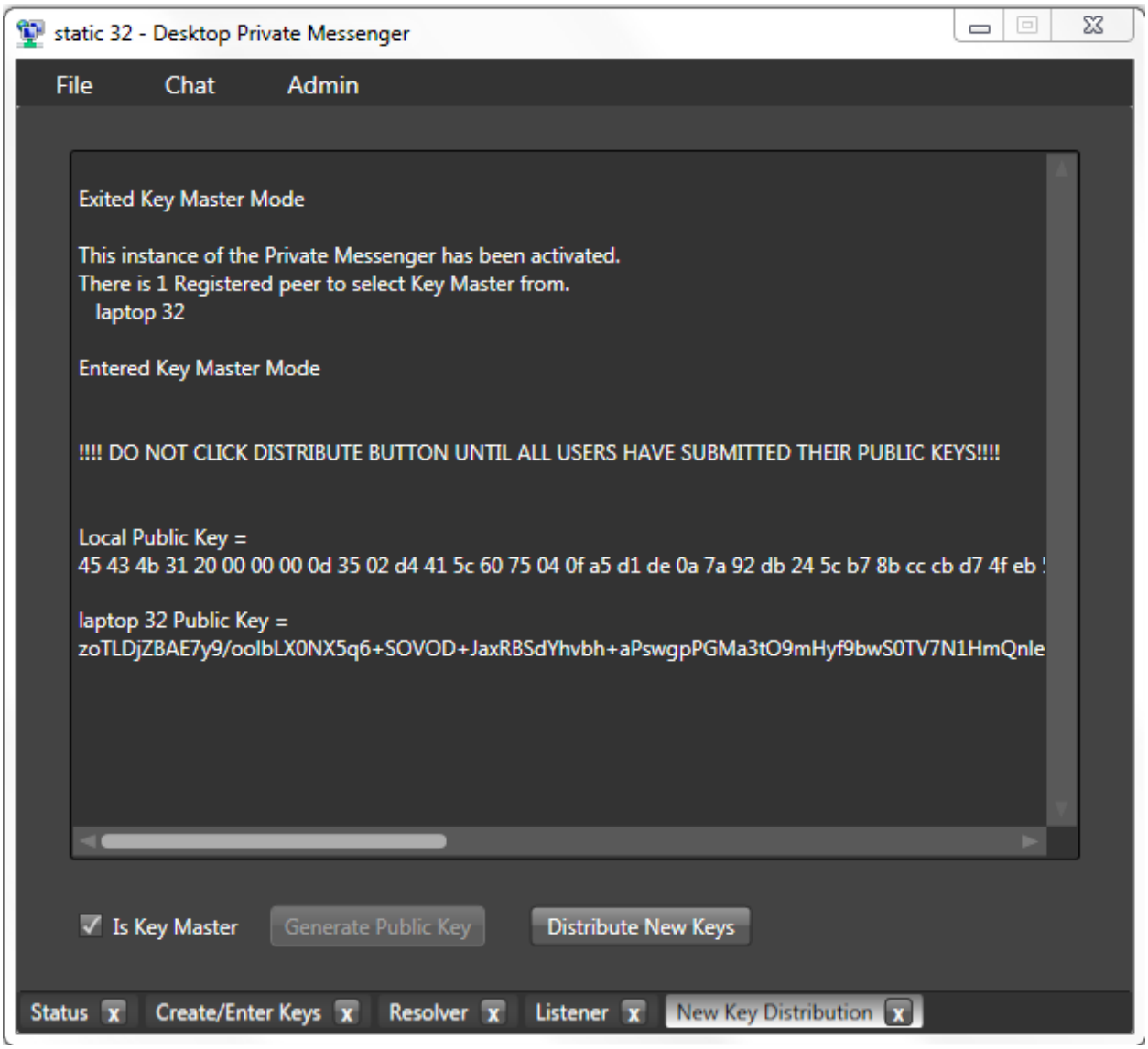


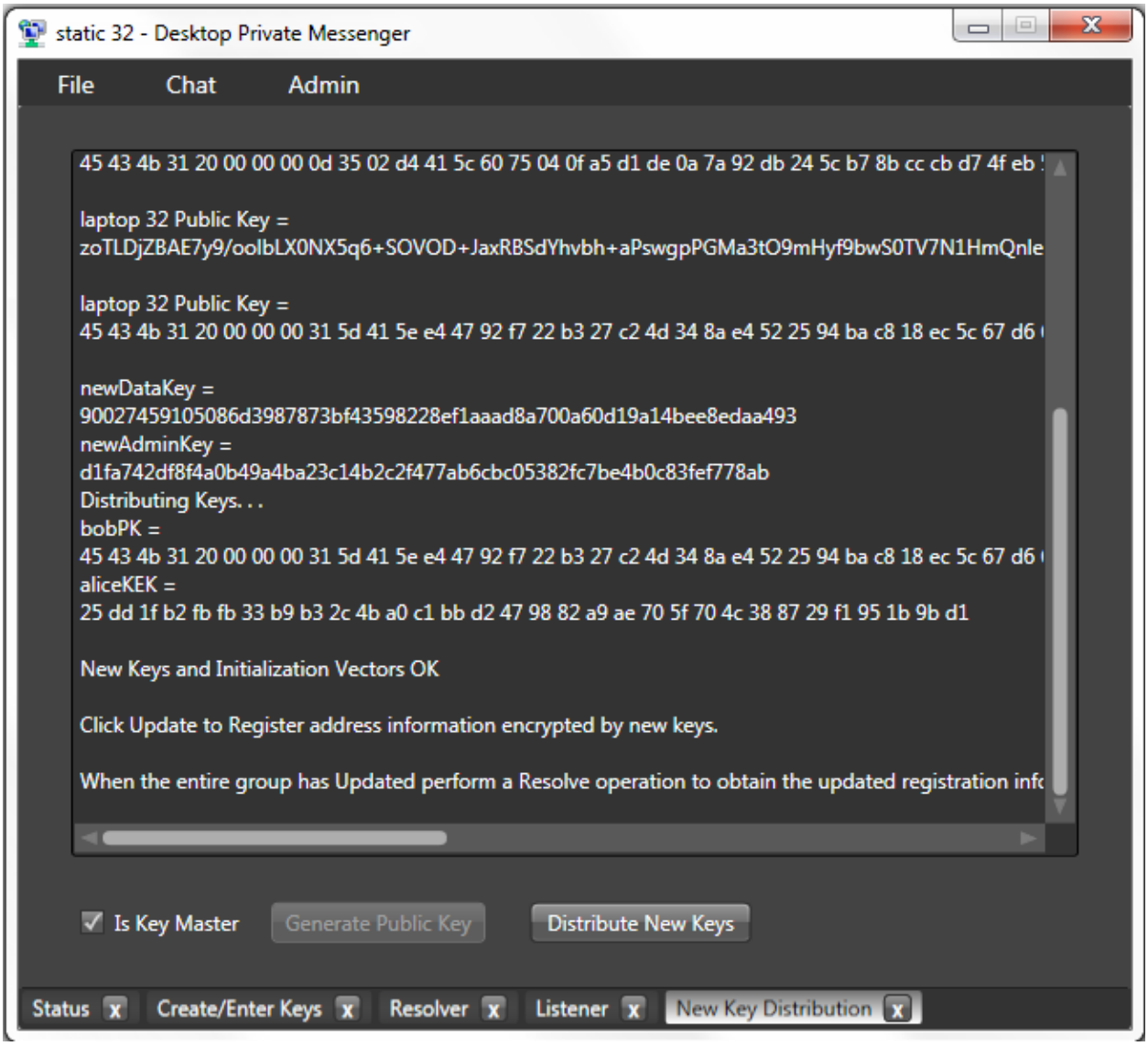
At this point the ‘Keys Are For Distribution’ Check Box is checked and the status message reflects that. Just as with the current keys both keys and IV’s can be entered or the IV’s used if desired. If the Key Master clicks ‘Generate IV’s and Keys’ new cryptographically strong values for those both keys and IV’s will be generated. This feature is both secure and convenient because until the button is clicked those values never existed and hence were not at risk of discovery.

From this point the Key Master continues either to the Resolver view or the Listener view as his role determines. Once the rest of the group is Registered chatting can begin with the current keys or distribution of the new keys can commence. If the group was chatting all chat windows must be closed then navigate to Admin -> Crypto – Distribute New Keys.

Those who are not the Key Master will follow the previous steps for the User Role. The Key Master will follow the subsequent steps below. The distribution process requires timely interaction with the other users in the group. They will select the designated Key Master and click ‘Generate Public Key’ then ‘Exchange Keys’. This causes the new public key to be encrypted with current session keys and sent directly to the Key Master.

Then when the Key Master can see the other group members have submitted ther new public keys he will then click ‘Distribute New Keys’. This causes a copy of the new keys to be encrypted with each users public key derived encryption key and sent to him along with the Key Masters public key which is encrypted with the current session key.


static 32 - Desktop Private Messenger
File
Chat
Admin
Exited Key Master Mode
This instance of the Private Messenger has been activated.
There is 1 Registered peer to select Key Master from.
laptop 32
Entered Key Master Mode
!!!! DO NOT CLICK DISTRIBUTE BUTTON UNTIL ALL USERS HAVE SUBMITTED THEIR PUBLIC KEYS!!!!
Local Public Key =
45 43 4b 31 20 00 00 00 0d 35 02 d4 41 5c 60 75 04 0f a5 d1 de 0a 7a 92 db 24 5c b7 8b cc cb d7 4f eb
laptop 32 Public Key =
zoTLDjZBAE7y9/oolbLX0NX5q6+SOVOD+JaxRBSdYhvbh+aPswgpPGMa3tO9mHyf9bwS0TV7N1HmQnle
Is Key Master
Generate Public Key
Distribute New Keys
Status
Create/Enter Keys
Resolver
Listener
New Key Distribution



Each user and the Key Master will receive indication the the new keys have been successfully decrypted and then placed into the database. Each user and the Key Master must go to the Listener view and click ‘Update Registration’. This action causes each group users Registration information to be placed in the Resolver encrypted with the new keys. Once all group members have updated then they must perform a Resolve on the Listener view to get the information. Now chatting can resume or begin as the case may be.

IMPORTANT:

Error handling:

This means such errors that affect the messenger application itself not data transmission types. The Private Messenger has been designed to fail gracefully and clean up all resources currently in use and write a small log file which can help support trouble shoot the issue. This means the messenger will suddenly disappear as if it crashed which could leave key and message data in memory, however, by gracefully closing the memory in use is cleaned up as if a normal exit had been performed. If something in your environment prevents the messenger from even launching a small log file will be written in that instance as well.

Other less debilitating errors will be captured and displayed on the view in use. These are errors that do not compromise data security and can be recovered from such as 'No Internet Connection Found' etc. If difficulty occurs attempting any connection the initial failure and retry attempts are displayed.

Contents of downloaded zip file: