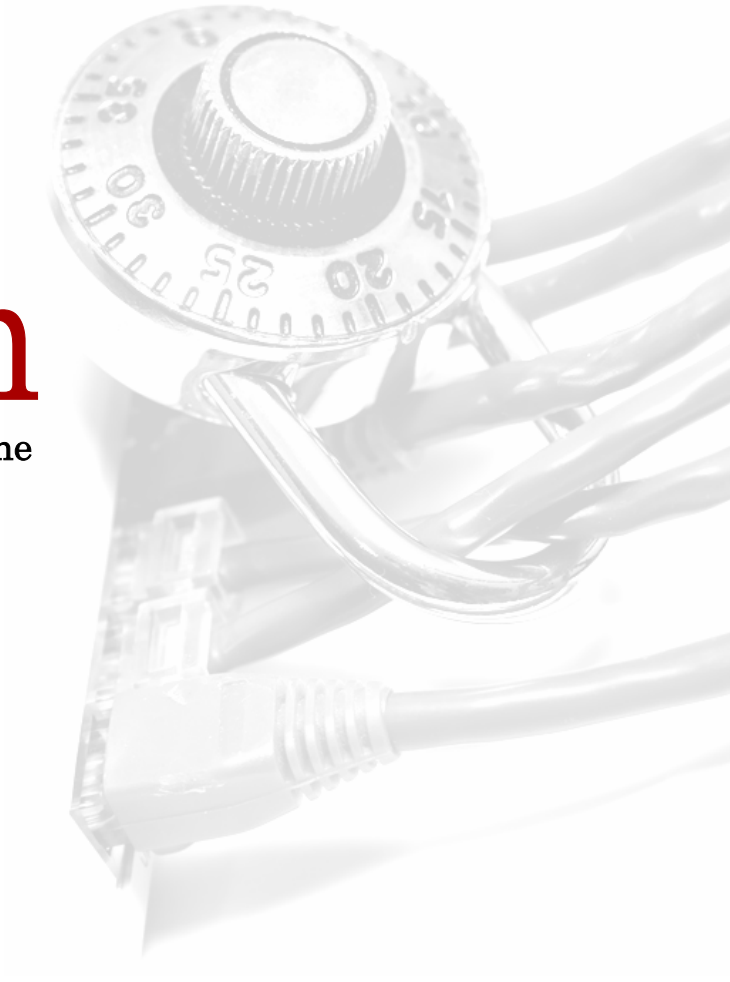
User's Guide

Kryptofon

Secure VoIP Phone



Mikica B Kocic
KTH, ID2013, 2010-11-30

This application was written as a final project of
KTH course ID2013 *Internetprogrammering I*

Contents

Introduction 1

Setting Up, Starting, and Quitting Kryptofon 1
Application Window 3

Messaging 5

Identifying Yourself to Other Kryptofon Users 5
Connecting and Disconnecting from the Chat Server 5
Sending Text Messages 7
Receiving Text Messages 8

Voice Calls 9

Listing Users 10
Making Calls 10
Receiving Calls 12
Clearing Calls 14
Sending Messages during the Call 15

Key Handling 17

Generating Keys 18
Authorizing Public Keys 19

Miscellaneous 21

Getting Help 21
Clearing Log and Saving Log Dumps 21

Commands Cheat Sheet 22

Introduction

Welcome to Kryptofon, a Java based application for secured voice and short message communication between internet users.

The desire for secrecy was one of the basic needs through out the centuries of human civilization. Nowadays, in the world of Internet, our voice and text communication can be, more than ever, easily intercepted jeopardizing our privacy. Kryptofon is a cryptographic solution to that remedy. It provides encrypted communication and verification between Kryptofon peers on the Internet.

Setting Up, Starting, and Quitting Kryptofon

Before you install and start using Kryptofon, please take a few minutes to make sure your computer meets the minimum requirements needed to run Kryptofon.

Requirements

Kryptofon is a Java based networking application that uses audio devices for voice capturing and playing sounds.

To use Kryptofon you need the following:

- Java Runtime Environment (JRE) version 1.6 or later (Java SE 6+) with Java Cryptography Extension (JCE)
- Audio device with a headset and a microphone
- Access to public or private IP network and some chat server. A chat server is used as a private branch exchange (PBX) for messaging between Kryptofon peers. Kryptofon uses SU DSV's public chat server `atlas.dsv.su.se:9440` by default.

Delivery

Kryptofon is distributed as a single Java Archive (JAR) file.

Setting Up and Starting Kryptofon

To install Kryptofon

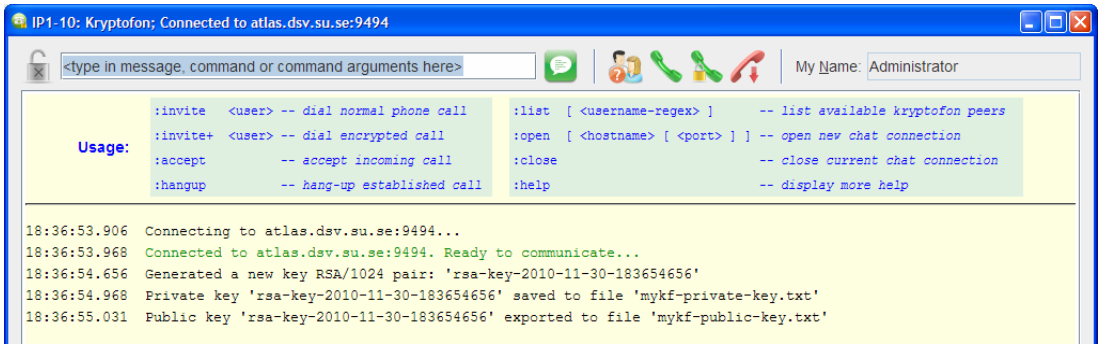
1. Create folder where you want Kryptofon installed
2. Download and copy of `kryptofon.jar` file into the created folder
3. Optionally create a shortcut icon to the JAR file in your desktop environment depending on the Operating System (OS) used

To start Kryptofon

You can start Kryptofon either from your Operating System's (OS) desktop environment or from the OS command line interpreter.

To start Kryptofon from the desktop, it is usually enough to double-click the Kryptofon JAR icon or the shortcut icon you have created during installation steps.

Kryptofon starts and displays the application window, with the short "usage" message.



Note: If started for the first time, Kryptofon would generate private/public key pairs and save them in files. For more details, see "Key Storage" under the chapter "Key Handling" on page 17.

To start Kryptofon from the OS command line, make sure first that the Java Runtime Environment binaries are in the search path for executables, then issue command:

```
java -jar Kryptofon.jar
```

When started from the OS command line, Kryptofon shows trace and debugging information to standard output stream:

```
Administrator@hagar /c/Work/space/ip1/10
$ java -jar kryptofon.jar
18:51:11.109 Trace [AWT-EventQueue-0] Private Key directory: C:\Documents and Settings\Administrator\.mykf\
18:51:11.890 Trace [AWT-EventQueue-0] Input Buffer Size = 320
18:51:11.968 Trace [AWT-EventQueue-0] Output Buffer Size = 320
18:51:11.968 Trace [Tick-send] Thread started
18:51:11.984 Trace [AWT-EventQueue-0] Created 8kHz 16-bit PCM audio interface; Sample size = 320 octets
18:51:11.984 Trace [Tick-play] Thread started
18:51:12.000 Trace [Ringer] Thread started
18:51:12.015 Trace [CipherEngine] New local symmetric cipher: Blowfish/CBC/PKCS5Padding
18:51:12.031 Trace [AWT-EventQueue-0] Bound to UDP port 47000
18:51:12.031 Trace [UDP] Thread started
18:51:12.046 Trace [AWT-EventQueue-0] Created instance of the class CryptoPhoneApp
18:51:12.062 Trace [Chat-atlas.dsv.su.se:9494] Thread started
18:51:12.156 Trace [CipherEngine] Instantiated asymmetric cipher: RSA/ECB/PKCS1PADDING
18:51:12.234 Trace [CipherEngine] Serialized Public Key in Base64; length = 812
```

This behavior can be suppressed by redirecting application's standard output stream to OS null device, e.g. to `/dev/null` on Linux or to `NUL:` on Windows.

Command line options

You can customize how Kryptofon starts by using options in the command line. Kryptofon recognizes following command line options:

```
java -jar Kryptofon.jar  [ server [ port ] ]
```

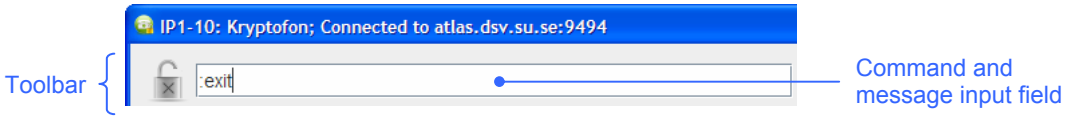
where:

- *server* is the hostname or IP address of the chat server. The default server is: `atlas.dsv.su.se`
- *port* is the TCP port where the chat server can be found. The default TCP port is: `9494`

Note: After you have started Kryptofon, you can issue command `:open` to open a new connection to a different chat server. See how "To connect to a chat server" on page 5 for more details.

To quit Kryptofon

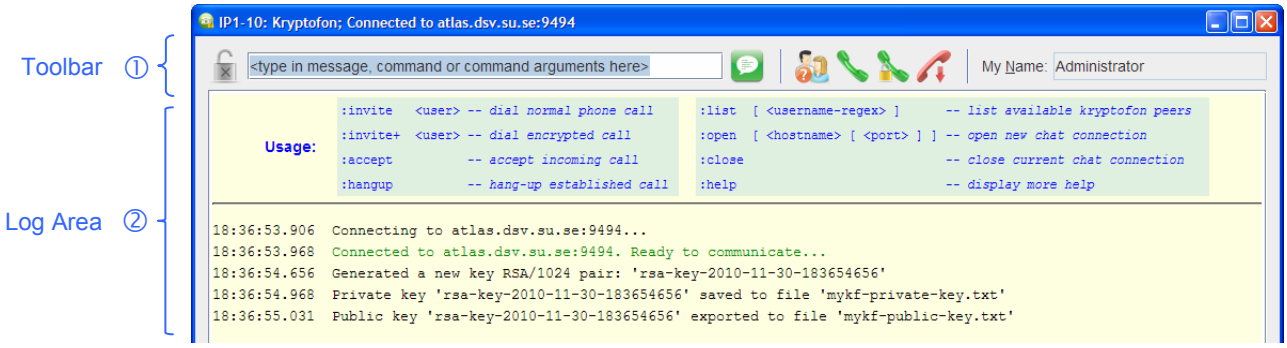
1. Enter command `:exit` in the command and message input field in the Toolbar
2. Press `ENTER`



Alternatively, you can close Kryptofon clicking on the close button in windows title bar or pressing `Alt+F4` or similar shortcut key depending on the OS you are using.

Application Window

Kryptofon application window consist of two major parts: Toolbar ① and Log Area ②.



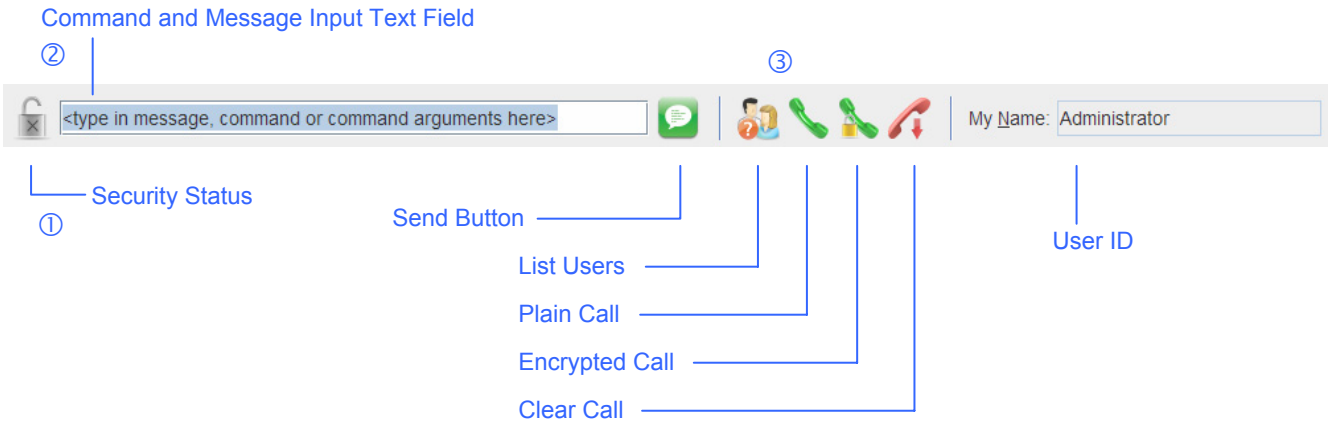
Log Area

Log Area contains both messages from the Kryptofon application and messages received from remote chat users and/or Kryptofon peers.

All messages shown in Log Area are stamped with the current time (with millisecond resolution) and highlighted using different color to emphasize different significance of the message contents. Generally, the red color is used for error and warning messages, while the green signals positive announcements.

Toolbar

Toolbar contains ① security status icon image, ② command and message input text field and ③ a number of buttons performing common functions. Buttons related to call establishement are grouped together and divided from the rest of the buttons.



Security Status

The security status icon shows current security state of the application, which may be:

- × Unsecured communication
- ? Secured (encrypted) communication but with an unauthenticated peer
- ✔ Secured communication with a verified (trusted) peer

Note: Encrypted communication is possible only during an established call between Kryptofon peers. However, the call between peers doesn't automatically assume that the communication is secured; peers may agree that the call is established without any encryption (unsecured).

Messaging

To communicate with other Kryptofon users you need a connection to some chat server. The only requirement imposed on the chat server is that it broadcasts unmodified incoming messages to all participants.

Kryptofon uses `atlas.dsv.su.se` as the default chat server.

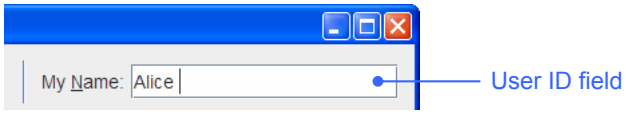
However, beside the chat server, you will also need a username to identify yourself and distinguish yourself from other users, otherwise the others won't be able to reach you.

Identifying Yourself to Other Kryptofon Users

Kryptofon sets your initial username to be the same name you have chosen to identify yourself to your OS, i.e. your OS login name is your Kryptofon user ID by default.

To change username

1. Select the User ID field in the Toolbar (click to field or press `Alt+N`)
2. Enter new username
3. Press `ENTER`



Kryptofon parses your text making it a proper username by removing leading and trailing white spaces and replacing inner spaces with a dashes; e.g. " Alice The User " parses as "Alice-The-User".

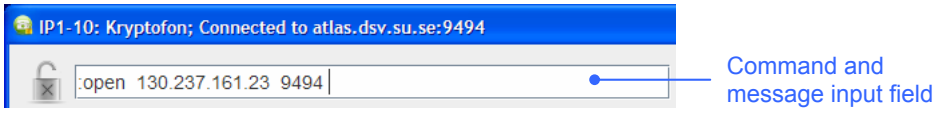
Note: You can change your username at any time except during the call.

Connecting and Disconnecting from the Chat Server

As mentioned earlier, Kryptofon uses `atlas.dsv.su.se` as the default chat server. To set up different chat server before Kryptofon starts, see "Command line options" on page 3.

To connect to a chat server

1. Select the Command and Message Input field in the Toolbar (click or press `Alt+I`)



2 Enter `:open host port`, where *host* is the name or IP address of the chat server and *port* is the TCP port chat server listens to.

3 Press ENTER

Example

To connect to chat server at IP address `130.237.161.23` and TCP port `9494`, issue command:

```
:open 130.237.161.23 9494
```

The result may look like:

```
12:31:06.359  Connection lost!
12:31:06.359  java.net.SocketException: socket closed
12:31:06.359  Closing connection atlas.dsv.su.se:9494...
12:31:06.359  ... connection closed atlas.dsv.su.se:9494
12:31:06.359  Connecting to 130.237.161.23:9494...
12:31:06.359  Connected to 130.237.161.23:9494. Ready to communicate...
```

To handle lost connection

In case of lost connection to the chat server, Kryptofon will try three times to reconnect, with 2-seconds delay between retries.

```
Reconnecting in 2 seconds...
Retry #2 of max 3:
13:07:32.109  Connecting to localhost:2000...
```

After failing for the third time, Kryptofon will require from you to manually open connection to some other chat server or to quit.

```
Retry #3 of max 3:
13:07:36.109  Connecting to localhost:2000...
13:07:37.296  I/O exception while connecting
13:07:37.296  java.net.ConnectException: Connection refused: connect
13:07:37.296  Closing connection localhost:2000...
13:07:37.296  ... connection closed localhost:2000
--------------------------------------------------------------------
Press ENTER to quit or type

   :open [ <hostname> [ <port> ] ]

to open new connection...
```

To close connection

1 Select the Command and Message Input field in the Toolbar (click or press `Alt+I`)

2 Enter `:close` command

3 Press ENTER.

Kryptofon closes connection to current chat server.

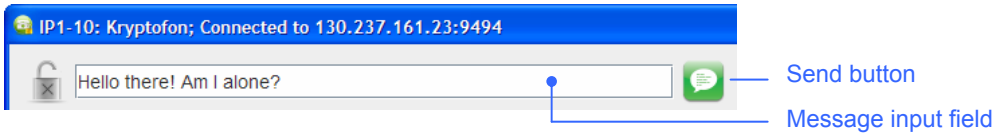
```
12:40:19.343  Connection lost!
12:40:19.343  java.net.SocketException: socket closed
12:40:19.343  Closing connection 130.237.161.23:9494...
12:40:19.343  ... connection closed 130.237.161.23:9494
```

Note: To send messages, you need a connection to some chat server. After closing current connection, if you try to send a message you will quit Kryptofon – you must enter some command not to quit.

Sending Text Messages

To send text message

1. Select the Message Input field in the Toolbar (click the field or press `Alt+I`)
2. Type in your message. Beware not begin the message with colon ' : ', otherwise the input text will be parsed as command.
3. Press `ENTER` or click the Send button.



Kryptofon sends your message to chat server, which broadcasts the message to all connected users.

```
13:05:07.593  Connecting to 130.237.161.23:9494...
13:05:07.593  Connected to 130.237.161.23:9494. Ready to communicate...
13:05:12.125  Alice: Hello there! Am I alone?
```

Note: To be sure that the message is broadcasted, you can use `:broadcast` command. See how "To broadcast non-encrypted message" on page 16.

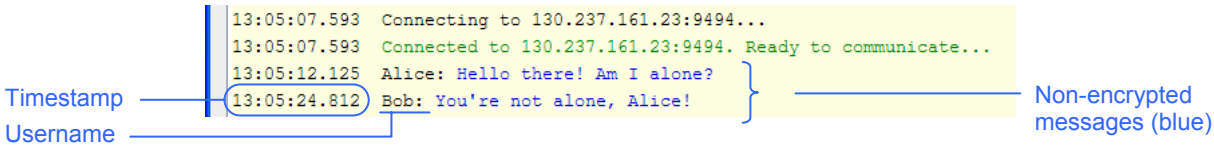
To send encrypted text message

You must have established secured (encrypted) call with remote Kryptofon peer to be able to send encrypted messages.

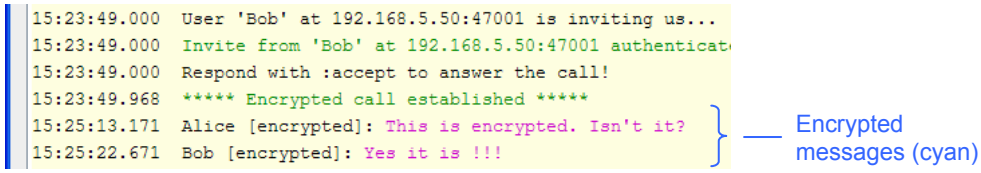
For more information see how "To invite user to encrypted call" on page 11, and how "To send encrypted message" on page 15.

Receiving Text Messages

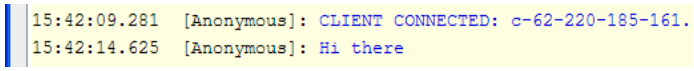
Kryptofon displays all incoming messages prefixed with the current time stamp and remote username.



Non-encrypted broadcast messages are highlighted in blue color, while encrypted messages are highlighted in cyan.



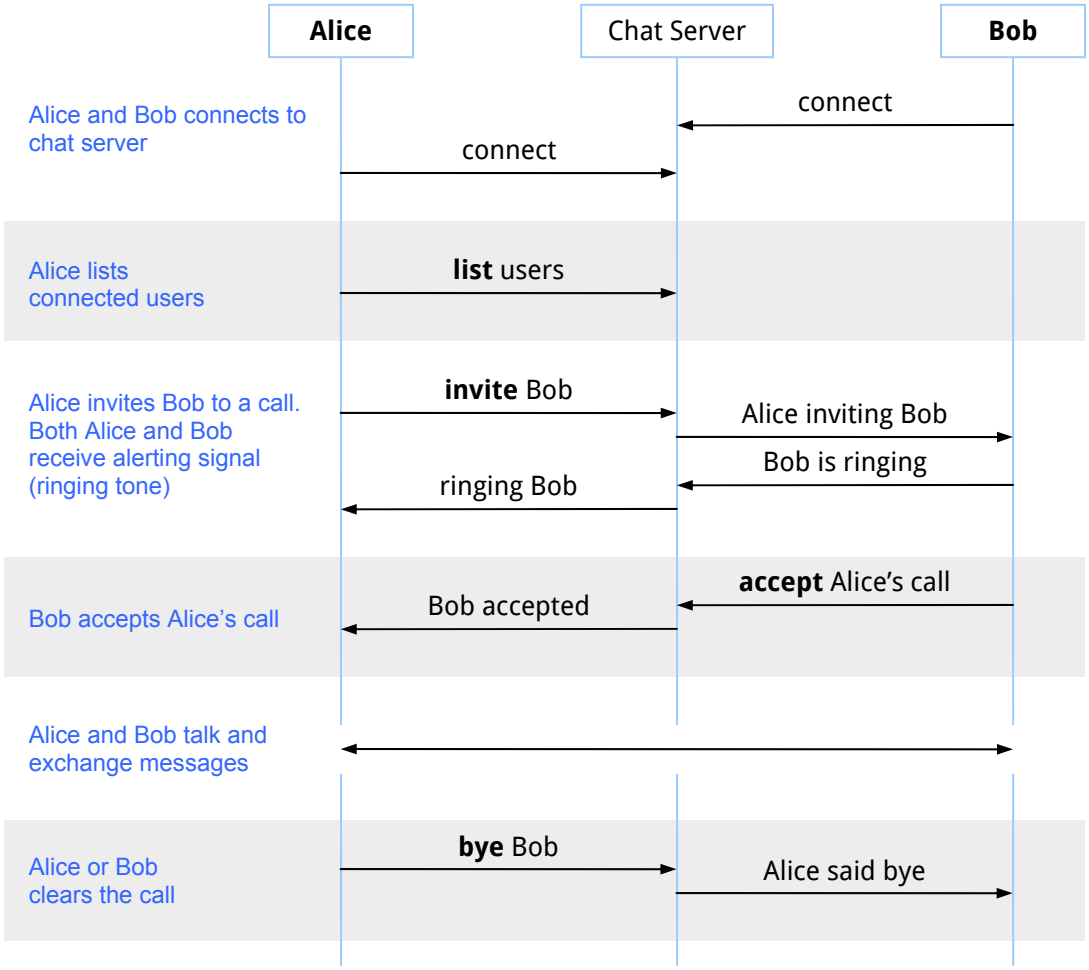
Note: Messages from non-Kryptofon peers are tagged with username `anonymous`.



Voice Calls

The following schematic diagram shows typical call scenario between two Kryptofon users, Alice and Bob.

In this scenario Alice initiates call to Bob, and Bob answers the call. (More details about the protocol can be found in the separate document “Kryptofon System Internals”.)



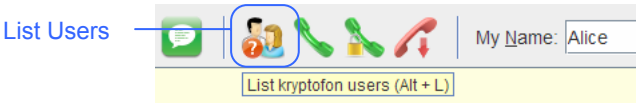
Reminder: Status icon displays current security state of the call:

- × Unsecured communication
- ? Secured (encrypted) communication but with unverified (unauthorized) peer
- ✔ Secured communication with verified (trusted) peer

Listing Users

To list Kryptofon peers connected to chat server

1 Click the List Users button or press `Alt+L`.



Kryptofon broadcasts poll request to all users on the chat server, but only Kryptofon peers responds to this poll with "I'm alive" reply.

```
18:49:10.049  Listing users...
18:49:10.049  -- User 'Alice' is alive.
18:49:10.189  -- User 'Bob' is alive.
18:49:10.189  -- User 'Mallory' is alive.
18:49:10.189  -- User 'Eve' is alive.
```

Note: To list users matching some regular expression criteria, you may issue `:list` command with parameters. For example, to poll all users with names beginning with "mal", enter command `:list ^mal.*`

In the previous example, only Mallory's Kryptofon will answer the poll:

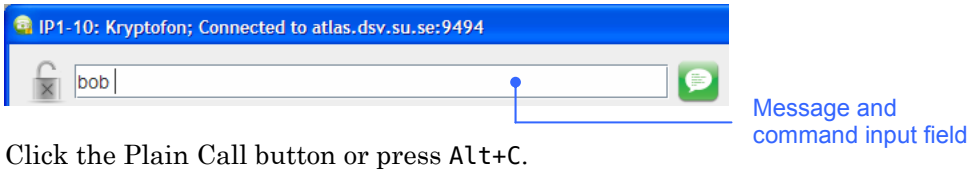
```
18:58:39.455  Listing users...
18:58:39.596  -- User 'Mallory' is alive.
```

Making Calls

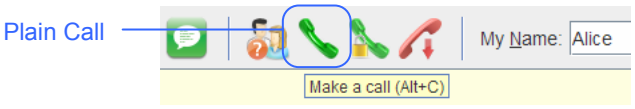
To invite user to plain (non-secured) call

1 Select the Message and Command Input field in the Toolbar (click or press `Alt+I`)

2 Enter the ID of the user you want to invite to a call.



3 Click the Plain Call button or press `Alt+C`.



Kryptofon sends invitation to remote peer. The peer should respond that remote user is alerted (ringing).

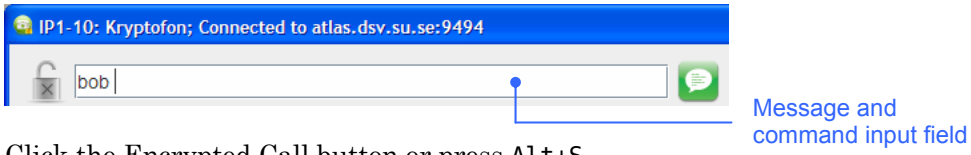
```
19:33:42.611  Inviting 'bob' to un-encrypted voice call...
19:33:42.767  User 'Bob' at 192.168.5.50:47001 is alerted...
19:33:42.767  Reply from 'Bob' at 192.168.5.50:47001 authenticated with public key
'rsa-key-2010-11-30-183654656'
```

In case that user is not present, invite fails with in the timeout of ~3 sec.

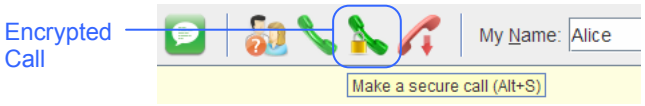
```
19:42:26.674  Inviting 'Mary' to un-encrypted voice call...
19:42:30.392  It seems that kryptofon user 'Mary' is not connected.
19:42:30.392  Use :list to query available users...
```

To invite user to encrypted call

1. Select the Message and Command Input field in the Toolbar (click or press `Alt+I`)
2. Enter the ID of the user you want to invite to a call.



3. Click the Encrypted Call button or press `Alt+S`.



Kryptofon sends invitation to encrypted voice call to remote peer. The peer responds that remote user is alerted (ringing).

```
19:49:43.908  Inviting 'Bob' to encrypted voice call...
19:49:44.064  User 'Bob' at 192.168.5.50:47001 is alerted...
19:49:44.064  Reply from 'Bob' at 192.168.5.50:47001 authenticated with public key
'rsa-key-2010-11-30-183654656'
```

In case that user is not present, invite fails with in the timeout of ~3 sec.

```
19:52:04.549  Inviting 'Mary' to encrypted voice call...
19:52:08.392  It seems that kryptofon user 'Mary' is not connected.
19:52:08.392  Use :list to query available users...
```

Note: Remote user may force your encrypted invite into the plain (non-encrypted) call.

Kryptofon warns you in such case and displays “unsecured” status image  (see “Security Status” section on page 4).

```
20:03:38.205  Inviting 'bob' to encrypted voice call...
20:03:38.330  User 'Bob' at 192.168.5.50:47001 is alerted...
20:03:38.330  Reply from 'Bob' at 192.168.5.50:47001 authenticated with public key
'rsa-key-2010-11-30-183654656'
20:03:43.017  User 'Bob' at 192.168.5.50:47001 has accepted our invite
20:03:43.033  ***** Un-encrypted call established *****
```

Receiving Calls

You may receive three types of call invitations:

- Invitation to a plain call without encryption

```
19:58:23.814  User 'Alice' at 192.168.5.50:47000 is inviting us...
19:58:23.814  The call will be without encryption.
19:58:23.814  Respond with :accept to answer the call!
```

- Invitation to a secured (encrypted) call from an unverified peer

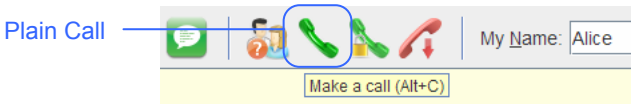
```
20:23:04.142  User 'Mallory' at 192.168.5.50:47003 is inviting us...
20:23:04.158  Invite from 'Mallory' at 192.168.5.50:47003 could not be authenticated.
20:23:04.158  Respond with :accept to answer the call!
```

- Invitation to a secured (encrypted) call from a verified peer

```
20:01:46.377  User 'Alice' at 192.168.5.50:47000 is inviting us...
20:01:46.377  Invite from 'Alice' at 192.168.5.50:47000 authenticated with public
key 'rsa-key-2010-11-30-183654656'
20:01:46.377  Respond with :accept to answer the call!
```

To answer non-encrypted call

1 Click the Plain Call button or press `Alt+C`.

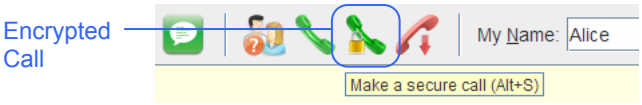


Kryptofon answers the call and informs you of the security status of the established call.

```
19:58:23.814  User 'Alice' at 192.168.5.50:47000 is inviting us...
19:58:23.814  The call will be without encryption.
19:58:23.814  Respond with :accept to answer the call!
19:59:38.236  ***** Un-encrypted call established *****
```

To answer encrypted call

1 Click the Encrypted Call button or press `Alt+S`.



Kryptofon answers the call and informs you of the security status of the established call.

```
20:41:39.361  User 'Alice' at 192.168.5.50:47000 is inviting us...
20:41:39.361  Invite from 'Alice' at 192.168.5.50:47000 authenticated with public key
'rsa-key-2010-11-30-183654656'
20:41:39.361  Respond with :accept to answer the call!
20:41:42.017  ***** Encrypted call established *****
```

In case of unverified remote peer, Kryptofon displays warning.

```
20:44:10.830  User 'Mallory' at 192.168.5.50:47003 is inviting us...
20:44:10.830  Invite from 'Mallory' at 192.168.5.50:47003 could not be authenticated.
20:44:10.830  Respond with :accept to answer the call!
20:44:14.674  ***** Encrypted call established *****
```

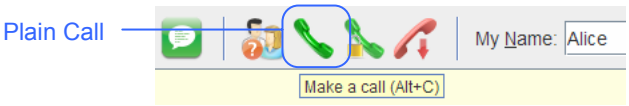
However, the call will be still considered secured (encrypted). Kryptofon leaves to you responsibility to verify authenticity of the remote peer during the (encrypted) conversation.

Note: If you click the Encrypted Call when invited to non-encrypted call, the call will be still answered as non-encrypted.

To answer encrypted call forcing it to be non-encrypted

To accept invitation to encrypted call and force the call to be non-encrypted:

1 Click the Plain Call button or press `Alt+C`.



Kryptofon answers the call and informs you of the “non-encrypted” security status of the established call.

Example

In the following example Alice invites ① Bob to encrypted call, but the Bob accepts ② the call as non-encrypted.

Alice's Kryptofon:

```
21:06:56.877  Inviting 'Bob' to encrypted voice call...   ①
21:06:57.033  User 'Bob' at 192.168.5.50:47001 is alerted...
21:06:57.033  Reply from 'Bob' at 192.168.5.50:47001 authenticated with public key
'rsa-key-2010-12-01-202231111'
21:06:58.658  User 'Bob' at 192.168.5.50:47001 has accepted our invite
21:06:58.674  ***** Un-encrypted call established *****   ②
```

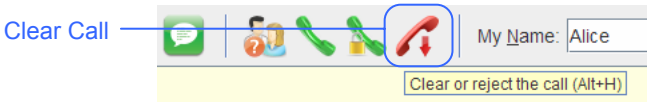
Bob's Kryptofon:

```
21:06:56.892  User 'Alice' at 192.168.5.50:47000 is inviting us...
21:06:56.908  Invite from 'Alice' at 192.168.5.50:47000 authenticated with public key
'rsa-key-2010-12-01-202231111'
21:06:56.908  Respond with :accept to answer the call!
21:06:58.674  ***** Un-encrypted call established *****   ②
```

Clearing Calls

To clear existing call

1 Click the Clear Call button or press `Alt+H`.

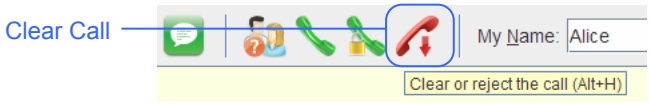


Kryptofon clears existing call and informs you of the call status.

```
21:28:18.080  Reply from 'Bob' at 192.168.5.50:47001 authenticated with public key
'rsa-key-2010-12-01-202231111'
21:28:19.158  User 'Bob' at 192.168.5.50:47001 has accepted our invite
21:28:19.267  Secret key from 'Bob' at 192.168.5.50:47001 authenticated with public key
'rsa-key-2010-12-01-202231111'
21:28:19.267  ***** Encrypted call established *****
21:28:23.986  ***** Call Ended *****
```

To reject incoming invitation

1 Click the Clear Call button or press `Alt+H`.



Kryptofon rejects invitation and informs you about that, and the remote Kryptofon informs also its user about the rejection.

Example Alice invites Bob to encrypted call. Bob rejects invitation ① from Alice.

```
21:30:07.283  User 'Alice' at 192.168.5.50:47000 is inviting us...
21:30:07.299  Invite from 'Alice' at 192.168.5.50:47000 authenticated with public key
'rsa-key-2010-12-01-202231111'
21:30:07.299  Respond with :accept to answer the call!
21:30:09.971  Rejecting invite from 'Alice' at 192.168.5.50:47000  ①
```

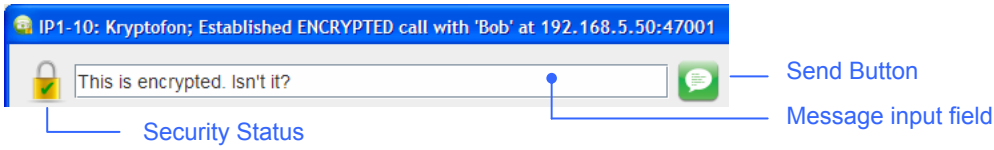
Alice receives rejection ② from Bob.

```
21:30:07.283  Inviting 'Bob' to encrypted voice call...
21:30:07.471  User 'Bob' at 192.168.5.50:47001 is alerted...
21:30:07.471  Reply from 'Bob' at 192.168.5.50:47001 authenticated with public key
'rsa-key-2010-12-01-202231111'
21:30:09.971  User 'Bob' at 192.168.5.50:47001 rejected our invite  ②
21:30:09.971  ***** Call Ended *****
```

Sending Messages during the Call

To send encrypted message

1. Verify that Status Icon in the Toolbar indicates secured communication.
   Icon image must be either  or  (see “Security Status” on page 4)
2. Select the Message Input field in the Toolbar (click the field or press `Alt+I`)
3. Type in your message. Beware that message doesn’t begin with colon ‘ : ’, otherwise the input text will be parsed as command.
4. Press `ENTER` or click the Send button.



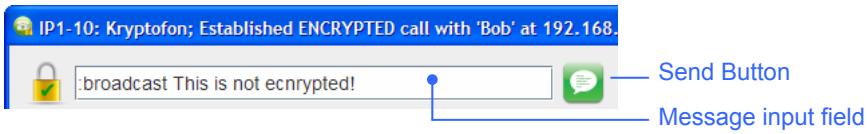
Kryptofon sends your message encrypted to remote peer via chat server.

```
15:23:49.000  User 'Bob' at 192.168.5.50:47001 is inviting us...
15:23:49.000  Invite from 'Bob' at 192.168.5.50:47001 authenticat
15:23:49.000  Respond with :accept to answer the call!
15:23:49.968  ***** Encrypted call established *****
15:25:13.171  Alice [encrypted]: This is encrypted. Isn't it?
15:25:22.671  Bob [encrypted]: Yes it is !!!
```

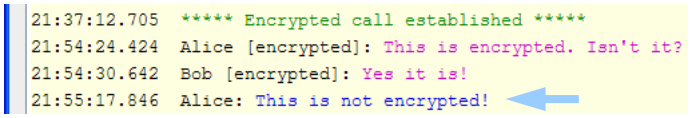
Note that encrypted messages are highlighted in cyan (see “Receiving Text Messages” on page 8).

To broadcast non-encrypted message

1. Select the Message Input field in the Toolbar (click the field or press `Alt+I`)
2. Enter `:broadcast` command followed by your message
3. Press `ENTER` or click the Send button.



Kryptofon broadcasts your message (with no encryption) to all remote chat clients.



Note that received broadcasts are highlighted in blue (see “Receiving Text Messages” on page 8).

Key Handling

To ensure secured communication, Kryptofon uses two types of ciphering:

Symmetric ciphering with a common secret key. Kryptofon uses symmetric ciphering for encryption of voice protocol data units (PDUs) and text messages.

Asymmetric ciphering with a private/public key pair. Kryptofon uses asymmetric ciphering for verification of the remote peer and secure exchange (encrypted transfer) of secret keys used for ciphering of voice and text.

Note: Kryptofon could also use, in principle, asymmetric ciphering for encryption of voice PDUs and text messages. However, in reality, asymmetric ciphering algorithms are slower and more CPU consuming than symmetric counterparts, which is an essential drawback when it comes to transferring voice in real time.

A random secret key is generated every time you launch Kryptofon. A new secret key may be generated at any time on your request e.g. just before answering invitation to a call.

Your private/public key pair is usually generated only once: when you start Kryptofon for the first time.

Key Storage

Kryptofon stores your private/public key pair in `mykf-private-key.txt` file and your public key in file `mykf-public-key.txt`. Both keys are stored in your home directory under the subdirectory `.mykf`

On Linux, access rights for `.mykf` directory and the private key file are adjusted so that no one but you (the owner) may read the contents.

Authorized Public Keys

Kryptofon loads authorized public keys from `mykf-authorized-keys.txt` file found in `.mykf` subdirectory of your home directory. If the file cannot be found, Kryptofon creates an empty file and adjusts (if possible) file's permissions so that no one but you may read the content.

Kryptofon loads authorized public keys from the file on the startup, so if you change its contents while Kryptofon is running, you will need to reload new contents with `:reauth` command.

Note: You can have encrypted calls with remote users that you didn't authenticate.

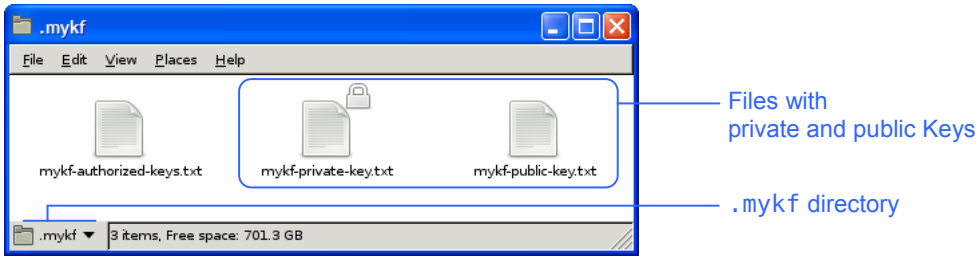
Kryptofon warns you in this case for an unauthenticated user, however, you can, contrary to other non-interactive cryptographic solutions, try to verify identity of the peer during the (encrypted) conversation.

Generating Keys

To generate new private/public key pair

To force Kryptofon to generate a new private/public key pair, you have to remove the file with the saved private key.

1. Quit Kryptofon (see how "To quit Kryptofon" on page 3)
2. Open file manager and navigate to your home directory, then to `.mykf` subdirectory



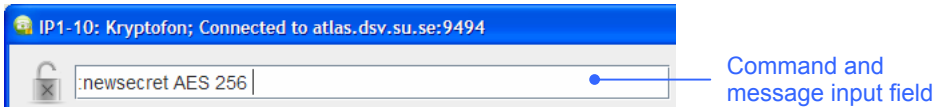
3. Remove file `mykf-private-key.txt`
4. Remove file `mykf-public-key.txt`
5. Start Kryptofon (see how "To start Kryptofon" on page 2)

Kryptofon generates your new private/public key pair on startup.

```
17:02:31.125  Generated a new key RSA/1024 pair: 'rsa-key-2010-12-02-170231125'
17:02:31.400  Private key saved to file '/home/h10/mikica/.mykf/mykf-private-key.txt'
17:02:31.443  Public key exported to file '/home/h10/mikica/.mykf/mykf-public-key.txt'
```

To generate new secret key

1. Select the Command and Message Input field in the Toolbar (click or press `Alt+I`)



2. Enter `:newsecret` *`algorithm keysize`*, where *`algorithm`* is a name of symmetric ciphering algorithm supported by JCE (e.g. Blowfish or AES) and *`keysize`* is the used key size for the algorithm. If you do not specify any of parameters, the defaults are Blowish and 32-bit key size.
3. Press `ENTER`

Kryptofon sets new algorithm and generates random secret key.

Example To generate new symmetric secret key with 256-bit AES algorithm, issue command:

```
:newsecret AES 256
```

The result may look like:

```
17:50:56.031  Connecting to atlas.dsv.su.se:9494...
17:50:56.031  Connected to atlas.dsv.su.se:9494. Ready to communicate...
17:50:56.343  Loaded private key 'rsa-key-2010-12-02-145437625' from file 'mykf-private-key.txt'
17:50:59.718  New symmetric cipher: AES/256
```

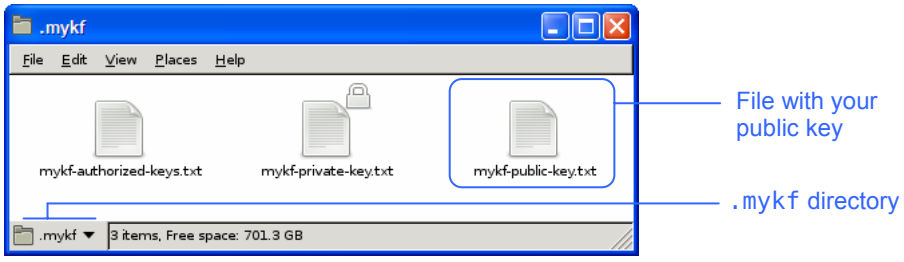
Authorizing Public Keys

Kryptofon exports your public key every time it generates new public/private key pair. You may send your public key to remote user either using Kryptofon or using some external encryption/decryption computer program such as PGP.

To find your public key

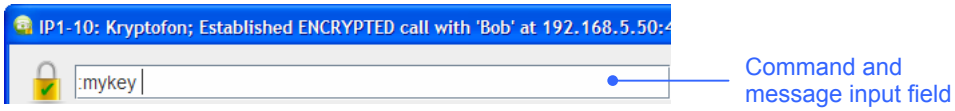
1. Open file manager and navigate to your home directory
2. Open `.mykf` subdirectory

Your will find your public key stored in file `mykf-public-key.txt`.



To send your public key using Kryptofon

1. Select the Command and Message Input field (click the field or press `Alt+I`)
2. Enter command `:mykey`
3. Press `ENTER`



Outside a call (or during unsecured call), Kryptofon broadcasts your public key to all connected users. However, during the secured call, Kryptofon sends your public key as encrypted message visible only to your peer.

Remote user can cut-and-paste the line with the public key from the Log Area into personal authorized keys file.

```
22:36:13.609  ***** Encrypted call established *****
22:39:20.937  Alice [encrypted]:
========= BEGIN PUBLIC KEY =========

H4sIAAAAAAAAAFvzloG1uIhBNLmosqAkX88vMTc1JaA0KScz2Tu18v229WKR5YkXmBiYfBjYk/Nzc1PzSkoYhHyy
 rsa-key-2010-12-02-145437625

========= END PUBLIC KEY ===========
```

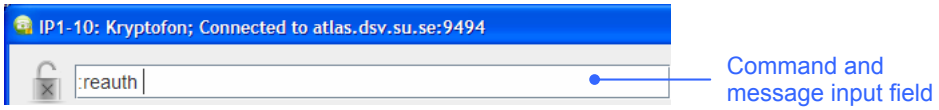
To authorize other user's public key

1. Open file manager and navigate to your home directory
2. Open `.mykf` subdirectory
3. Edit file `mykf-authorized-keys.txt`
4. Append the contents of the received public key to the file (only the line with the key and the comment)

Note: Kryptofon, when started for the first time, creates an empty `mykf-authorized-keys.txt` file and adjusts file permissions so that no one but you can read file contents. You should be careful not to accidentally blot the file permissions while editing the file.

To reload authorized keys

1. Select the Command and Message Input field in the Toolbar (click or press `Alt+I`)



2. Enter `:reauth`
3. Press `ENTER`

Kryptofon reloads file with authorized public keys and informs you about loaded keys.

```
18:07:02.843  Connecting to atlas.dsv.su.se:9494...
18:07:02.921  Connected to atlas.dsv.su.se:9494. Ready to communicate...
18:07:03.218  Loaded private key 'rsa-key-2010-12-02-145437625' from file
'mykf-private-key.txt'
18:07:19.265  Listing users...
18:07:19.265  -- User 'Administrator' is alive.
18:08:06.031  Loaded 2 authorized keys: rsa-key-2010-12-02-145437625,
rsa-key-2010-12-02-170231125
```

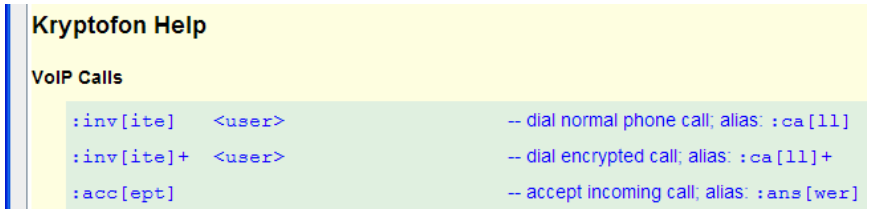
Miscellaneous

Getting Help

To display help

1 Press F1

Kryptofon displays command line reference.



Clearing Log and Saving Log Dumps

To clear screen

1 Select the Command and Message Input field in the Toolbar (click or press Alt+I)

2 Enter :cls command

3 Press ENTER

Kryptofon clears Log Area and displays initial "usage" info.

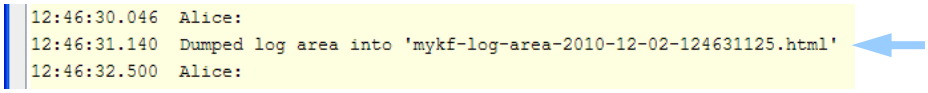
To save log into file

1 Select the Command and Message Input field in the Toolbar (click or press Alt+I)

2 Enter :dump command

3 Press ENTER.

Kryptofon saves the contents of the log area into mykf-log-area-*timestamp*.html file in the current directory (where *timestamp* is the current date and time).



Note: The :dump command accepts a filename as argument. To save log area into specific file, for example test.html, enter command :dump test.html

Commands Cheat Sheet

Chat Server Connection

:op[en] [*host* [*port*]]	open new chat connection
:clo[se]	close current chat connection

Instant Messages

:br[oadcast] *message*	broadcast (always un-encrypted) message

Kryptofon Peers

:who	send "wwhhoo" message to chat server to list all connected clients to chat server
:li[st] [*username-regex*]	list Kryptofon users connected to chat server

VoIP Calls

:inv[ite] *user* **:ca[ll]**	invite user to a normal (non-encrypted) voice call
:inv[ite]+ *user* **:ca[ll]+**	invite user to secured (encrypted) voice call
:acc[ept] **:ans[wer]**	accept incoming invitation
:by[e] **:ha[ngup]**	clear established call or reject incoming invitation

Key Handling

:my[key]	if in the call, display my public key to remote peer; otherwise, broadcast my public key to everyone
:reauth	reload authorized public keys
:newsecret [*algorithm* [*keysize*]]	initialize symmetric ciphering and generate new secret

Application

:du[mp]	save what you see (log area) as HTML file
:qu[it] **:exit**	quit application
:help	display command line reference
:cls	clear screen and display short usage info