appGATE™

MindTerm

MindTerm 2.0 User's Guide

The contents of this document are subject to revision and can be changed without notice. AppGate AB shall have no liability for any error or damage resulting from the usage of this document.

1 Overview

1.1 Overview of SSH (Secure Shell)

From the beginning, SSH was designed to be a replacement for the rsh/rcp/rlogin programs on Unix computers to do exactly what they do (i.e. remote login and copying files between Unix hosts), but in a secure way. Unlike the old Unix r-commands, SSH uses strong cryptography to protect traffic from eavesdropping, and offers optional strong authentication with public keys. Symmetric-key exchange, as well as SSH server authentication, is always done using public key encryption. The ability to provide transparently encrypted tunnels was also added to SSH. With this ability, non-secure programs based on tcp-connections may also be used securely over insecure networks.

For more detailed information about how the SSH protocol works and what benefits it provides, see the following web-sites:

`http://www.employees.org/~satch/ssh/faq/ssh-faq-2.html`

`http://www.openssh.com/faq.html`

SSH has become a de facto standard for remote administration and access to all sorts of Unix systems. With that has also come the need for clients that run on platforms other than Unix. This need has only partly been fulfilled with the great variety of free and commercial clients, mostly for Win95/98/NT, that have emerged.

1.2 Overview of AppGate MindTerm

The goal of AppGate MindTerm is to provide a single client for all platforms that can be used in a simple way to leverage the benefits of the SSH protocol. Not only does MindTerm offer the ability to run on many different platforms, it also offers the unique advantage of being accessible through a normal web browser as a java applet. This is invaluable for persons who are mobile and cannot install SSH clients wherever they go. This means that a company can give its employees access to a secure login shell as well as secure tunnels for ftp, smtp, pop, imap, etc. from "the road" using only a normal web browser with no installation required on the client side.

Apart from portability, MindTerm also offers some other unique features, such as ftp-proxying, built in scp file-transfer, connection keep-alive, etc., which are not part of standard SSH clients.

AppGate MindTerm can be heavily customized for specific needs. For instance, it may be slimmed-down to support only one block cipher and have no menus (its size can shrink to <150k). When the local file system is not accessible or local configuration files are not desired, all parameters may be set either on the command line or through applet parameters. Optionally, MindTerm can execute a single command (i.e. Pine or Midnight Commander) on the SSH server. When the command completes, MindTerm exits. This can be done both when running it in stand-alone mode (like when running the normal Unix SSH client) AND when running it as an applet.

2 Installation

2.1 Stand-alone Mode

In order to use AppGate MindTerm as a stand-alone client, complete the following:

1. Download `mindtermbin.zip` or compile the source files. The source files may be bundled into a jar file.
2. Unzip the file and extract it to a directory that already exists on the system (i.e. C:\mindterm).
3. Download a Java Runtime (JDK or JRE) for your platform. MindTerm should work with any 1.1.x or 1.2 JDK/JRE. It also works with Netscape's and Microsoft Internet Explorer's browser-supplied Java Runtimes.

Java Runtimes can be found at the following web-sites:

- Linux:
  `http://www.blackdown.org/java-linux.html`
  `http://www.alphaworks.ibm.com/tech/linuxjvm`
- Win32 and Solaris:
  `http://www.javasoft.com/products/`
- Macintosh:
  `http://www.apple.com/java/`
- Other platforms:
  `http://java.sun.com/cgi-bin/java-ports.cgi`

2.2 Applet Mode

To use as an applet, download the `mindtermbin.zip` file, or compile the source files (optionally bundling them into a jar file). Assuming you have the jar file (e.g. mindterm.jar) you must write an html-page as in the example in section 7 'Applet Usage' on page 37. If you are using a cryptographically signed binary version of AppGate MindTerm as an applet from your Netscape or IE browser, you will be able to use it exactly as the stand-alone version (or any other SSH client), i.e. connect to any host, set up tunnels, save/load settings from file, use system clipboard etc. The applet might also be given these permissions "manually" depending on your browser or applet viewer.

Please read this entire text before starting to use MindTerm!!

3 Quick Start

The following examples show how to start AppGate MindTerm as a stand-alone program with no parameters changed:

Linux/jdk1.1.x:

```
/usr/local/java/bin/java -classpath /usr/local/java/lib/
classes.zip:mindtermfull.jarcom.mindbright.application.MindTerm
```

Win32/jdk1.1.x:

```
c:\jdk1.1.x\bin\java -classpath
c:\jdk1.1.6\lib\classes.zip;c:\mindbright\mindtermfull.jar
com.mindbright.application.MindTerm
```

Win32/jre1.1.x:

```
c:\jdk1.1.x\bin\java -cp c:\mindbright\mindtermfull.jar
com.mindbright.application.MindTerm
```

Win32/jdk/jre1.2:

```
c:\jdk1.2.x\bin\java -cp c:\mindbright\mindtermfull.jar
com.mindbright.application.MindTerm
```

```
c:\jdk1.2.x\bin\javaw -cp c:\mindbright\mindtermfull.jar
com.mindbright.application.MindTerm
```

NOTE: The javaw runtime version does not create a DOS shell window for the console, making it more convenient for “real” usage.

Win32/jview (Microsoft’s JVM supplied with IE4 and later):

```
jview /cp:p mindtermfull.jar com.mindbright.application.MindTerm
```

MacOS/MRJ:

First, get the JBindery application. It is found in the MRJ SDK at the following website:

```
http://developer.apple.com/java/text/
download.html#sdk
```

Then, drop the mindtermfull.jar file onto the JBindery icon and give it the class name ‘com.mindbright.application.MindTerm’. Save it and run MindTerm with just a double-click.

NOTE: When using Windows 2000, double-clicking on the jar file should start MindTerm automatically.

By default, AppGate MindTerm handles most things automatically. Settings are handled on a per-server basis, and are automatically saved and loaded as needed. MindTerm saves all settings in its home directory on the client machine. By default, the home directory is:

`/home/`*`user`*`/mindterm`

for UNIX clients, and

`C:\Documents and Settings\`*`user`*`\mindterm\`

for Windows 2000 clients, for example.

Apart from the settings files, this directory also contains the `hostkeys` file, which is used for server identification. The SSH2 host key files are saved in the same format as SSH Inc.'s client. The public key identity files used with the public key authentication method (RSA or DSA) are also stored in the user's MindTerm home directory.

To change the home directory of MindTerm, the directory to use must be given as a command-line parameter (`--h /home/`*`user`*`/`*`directory`*), or with an applet parameter (`<param name=sshhome value="c:\`*`dir`*`\">`).

AppGate MindTerm may be started with all of the necessary settings on the command-line (or as applet parameters). This can be useful, for example, in creating double-clickable shortcuts for running AppGate MindTerm with specific settings. Also, instead of listing all the parameters every time the program is launched, AppGate MindTerm can be pointed to a file that contains all of the settings needed. For example, in Win95/98/NT:

```
javaw -cp c:\mindterm\mindtermfull.jar
     com.mindbright.application.MindTerm --q
     --fc:\mindterm\companyssh.mtp
```

This will launch AppGate MindTerm with the settings found in the file `C:\mindterm\companyssh.mtp` . This file may, for instance, directly connect to the server without prompting the user for server and user name, run the Pine mail program to read mail, and exit MindTerm when Pine is exited. The above command can, of course, be saved as a Windows shortcut.

Note about Java Runtimes

For best results, we recommend downloading the JDK from Javasoft for Windows clients, even though many operating systems come with a Java Runtime preinstalled (Win95/98/NT that have IE4 or later has the jview runtime, MacOS 8 and later have the MRJ runtime installed).

The Windows example on the previous page could be rewritten as:

```
jview /cp:p c:\mindterm\mindtermfull.jar
   mindbright.application.MindTerm --q --f
   c:\mindterm\companyssh.mtp --p none --m no
   pine
```

and saved as a shortcut, and it would run on most Windows machines without having to download a separate Java Runtime.

To create a short name for a server (and/or multiple settings for a single server), the ‘Settings--New Server...’ dialog box (Figure 3. 'New Server screen' on page 14) should be used. Using this box, configure all of the settings for this session and check the ‘Save as Alias’ check box. The settings will be saved in an ‘.mtp’ file for future use. For more information, refer to section 4.3.1 on page 14.

Another way to create a new settings file is to connect to an existing server (one which already has a settings file) and do (File - Save As...). Then the settings file may be edited manually as required. Remember, settings files must have the extension '.mtp' and reside in the home directory of AppGate MindTerm. All settings in MindTerm have decent default values, and can normally be run without any parameters.

For detailed descriptions of all of the available parameters, see section 5 'Parameters' on page 25 and/or Appendix A 'Connection Parameters' on page 42.

4 Menus

The easiest way to learn how AppGate MindTerm works and what features it provides is to look through this brief walkthrough of the menus in AppGate MindTerm. Given within parentheses is the keyboard shortcut for each menu item, if one exists.

4.1 File Menu

4.1.1 New Terminal (Ctrl+Shift+N)

This will create a new AppGate MindTerm window with the same settings as the first AppGate MindTerm window of this session. All parameters (command-line or applet) given to MindTerm at startup will affect each new terminal created.

4.1.2 Clone Terminal (Ctrl+Shift+O)

This will create a new AppGate MindTerm window with the exact same settings as the window it is created from. If the window contains a connected session, the new window will be automatically logged in to the same SSH server. If the server is using SSH1, the same authentication that was used in the original window will be used. In SSH2, the cloned window will be a new session channel running a shell using the same connection to the server. In other words, additional authentication will not be required.

4.1.3 Connect... (Ctrl+Shift+C)

This launches the Connect dialog. From this dialog, you may either select to connect to a host whose settings you have saved, or you may create settings for a new host. Note when selecting "New Server" a new dialog is shown which is identical to the one described in section '4.3.1 SSH Connection...'.

4.1.4 Disconnect (Ctrl+Shift+D)

This forces the current session to be disconnected. Note that this will cause all tunnels to be closed and the shell to be abandoned without logging out. The preferred way to disconnect is to logout in the shell.

4.1.5 Load Settings...

Loads settings from a file (extension .mtp) without connecting to the server.

4.1.6 Save Settings (Ctrl+Shift+S)

Saves current settings.

4.1.7 Save Settings As...

Creates a new settings file and saves current settings to it. This is useful for creating a short name for a server, or for having more than one set of settings for a specific server.

4.1.8 Create Keypair...

Creates an identity to be used with authentication type 'public key'. Two files are created, one containing the private key (default name 'identity') and one containing only the public key (default name 'identity.pub'). The contents in the file with the extension .pub must be copied to the file 'authorized_keys' on the server (typically found in ~/.ssh/). These key-files are identical to the ones used with the Unix version of SSH.

4.1.9 SCP File Transfer...

In this dialog, you may choose files and/or directories to transfer to or from the SSH server, as shown in fig. 1 on the next page.

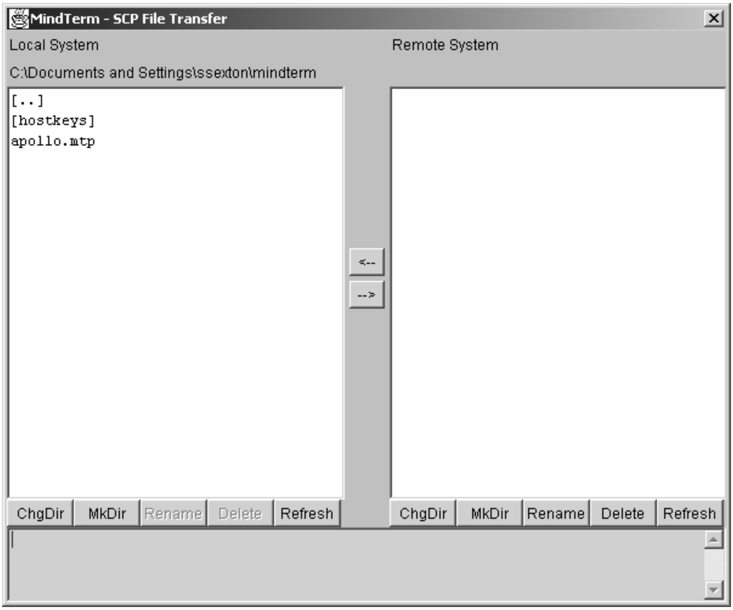


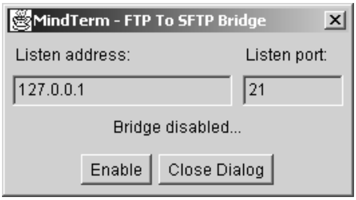
Figure 1. SCP File Transfer screen

Local files and directories may be copied to and from the local and remote systems by selecting them and choosing the proper arrow to move them. The left side of the screen represents the local system, and the right side of the screen represents the remote system. The directory assumed on the remote side is the user's home-directory (just like with the standard UNIX scp client).

The directory displayed in the window may be changed simply by clicking the ‘ChgDir’ button, and a new directory may be created with the ‘MkDir’ button.

4.1.10 FTP to SFTP Bridge...

This menu selection displays the window shown in fig. 2 on the next page.



Listen address: 127.0.0.1

Listen port: 21

Bridge disabled...

Figure 2. FTP to SFTP Bridge screen

This represents a proxy on the local side that should be enabled if an ordinary FTP client with an SFTP server in SSH2.

4.1.11 Capture to File...

This captures terminal output to a file. Capture starts immediately when the file has been selected and ends when this menu item is selected again. Note that while capturing is active, the menu item remains selected.

4.1.12 Send ASCII File...

This will send the contents of the selected file to the terminal as input (i.e. would be the same as if the contents were typed from the keyboard).

4.1.13 Close (Ctrl+Shift+E)

This closes the current window. Note that when closing a window without logging out, you are aborting the SSH connection abnormally. It is advisable to logout in the shell before closing/exiting AppGate MindTerm.

4.1.14 Exit (Ctrl+Shift+X)

This closes all windows and exits AppGate MindTerm. Note that when closing windows without logging out, you are aborting the SSH connection abnormally. It is advisable to logout in the shell before closing/exiting MindTerm.

4.2 Edit Menu

NOTE: The system clipboard is only available in signed applets. In some implementations of the Java Runtime, the system clipboard is not available. In the case where the system clipboard is not available, copy/paste operations will only work within and between MindTerm terminal windows.

4.2.1 Copy (Ctrl+Ins)

This copies the selected text to the clipboard. The selection is done by clicking and holding down the left mouse button while dragging the mouse over the area to select. Let go of the mouse button when the desired text is selected.

4.2.2 Paste (Shift+Ins)

This pastes the contents of the clipboard to the terminal as input (i.e. would be the same as if typed from keyboard) .

4.2.3 Copy & Paste

This does a copy, followed by a paste.

4.2.4 Select All (Ctrl+Shift+A)

This selects all content in the scrollback buffer and in the terminal.

4.2.5 Find... (Ctrl+Shift+F)

This displays the Find dialog, from which the scrollback buffer and terminal contents can be searched for words. The search can be done case-sensitive or case-insensitive. Each word found is highlighted. An alert is sounded when no more matches are found.

4.2.6 Clear Screen

This clears the screen and sets the cursor position to the upper left corner.

4.2.7 Clear Scrollback

This clears the contents of the scrollback buffer.

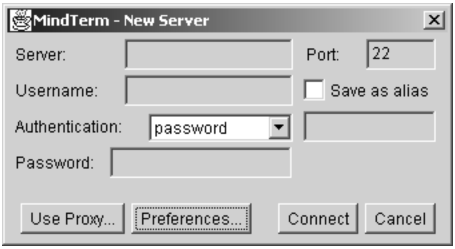
4.2.8 VT Reset

This resets the terminal settings to default (i.e. clears line-draw graphics mode which might be mistakenly set by displaying a binary file).

4.3 Settings Menu

4.3.1 New Server... (Ctrl+Shift+H)

Selecting this menu option will display the screen shown in fig. 3 below.



MindTerm - New Server

Server:

Port: 22

Username:

□ Save as alias

Authentication: password

Password:

Figure 3. New Server screen

In this dialog, parameters may be set for each server AppGate MindTerm is to connect to. The options should be detailed in the following manner:

Server: Server name or IP address of the server to connect to.

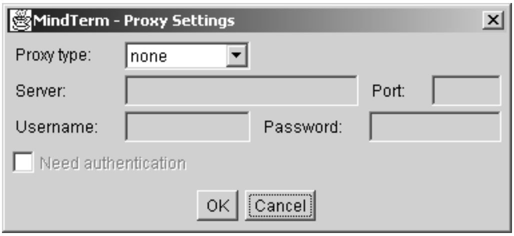
Username: User name to connect as.

Port: Port on server to connect to (22 by default).

Authentication: Authentication method to use when connecting to this server. The options below ‘Authentication’ will change depending on the authentication method selected.

Save as alias: This checkbox should be checked if these setting should be saved for future use.

To use a proxy, click the ‘Use Proxy...’ button. The screen shown in fig. 4 below will be displayed. For further details on this screen, refer to 4.3.5 'Proxy...' on page 19.



MindTerm - Proxy Settings

Proxy type: none

Server:

Port:

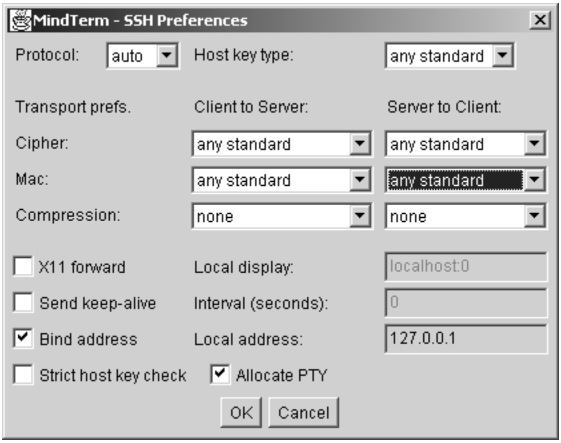
Username:

Password:

□ Need authentication

Figure 4. Proxy Settings screen

To view more options, click the ‘Preferences...’ button. That will display the screen shown in fig. 5 on the next page.



MindTerm - SSH Preferences

Protocol: auto

Host key type: any standard

Transport prefs.	Client to Server:	Server to Client:
Cipher:	any standard	any standard
Mac:	any standard	any standard
Compression:	none	none

□ X11 forward — Local display: localhost:0

□ Send keep-alive — Interval (seconds): 0

☑ Bind address — Local address: 127.0.0.1

□ Strict host key check

☑ Allocate PTY

Figure 5. Preferences screen

In this dialog, the following parameters may be set:

Protocol: The SSH protocol to use (auto, SSH2, SSH1). The default is ‘auto’. MindTerm will automatically conform to the SSH protocol used on the server.

Host key type: Which type of host key to use (any standard, ssh-rsa, ssh-dss). The default selection is ‘any standard’.

Transport prefs: These settings are transport preferences for traffic to and from the SSH server. Configurable parameters are Cipher, Mac, and Compression. These are separately configurable for traffic ‘Client to Server’ and ‘Server to Client’.

X11 forward: This checkbox should be checked if X11 data is to be forwarded from the server to the client. If this box is checked, **Local display** will need to be set to the display on which the X11 data is to be shown (default is localhost:0).

Send keep-alive: This checkbox should be checked if keep-alive information is to be sent to the server. If this box is checked, **Interval** should be filled in with the seconds between keep-alive signals.

Bind address: The local address to bind to for forwards. By default, this is set to 127.0.0.1, which is the address of localhost.

Strict host key check: This checkbox should be checked if the host key sent by the server MUST be checked (i.e. it can’t be accepted by the user manually) for a connection to be made.

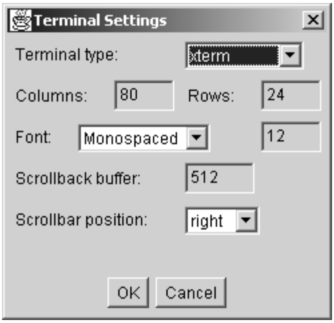
Allocate PTY: This checkbox should be checked if a PTY terminal should be allocated with this connection.

When connected, you can set the parameters for the current session. Note that some changes will not take effect until the next time you connect to this server.

When the client is not connected, a new session is created if one is not found with the name of the server. In this case, the same dialog appears that is shown when selecting "New Server..." from the Connection dialog (see section 4.1.3).

4.3.2 Terminal... (Ctrl+Shift+T)

This menu selection will display the dialog box shown in fig. 6 below.



Terminal type: xterm

Columns: 80 Rows: 24

Font: Monospaced 12

Scrollback buffer: 512

Scrollbar position: right

Figure 6. Terminal settings screen

In this dialog, you may set the basic terminal parameters, such as terminal type, size, and font. The following parameters may be set in this dialog:

Terminal type: terminal type to be used in the current session.

Columns: the number of columns to be used in the terminal.

Rows: the number of rows to be used in the terminal.

Font: the font to be used in the terminal (defaults to ‘Monospaced’).

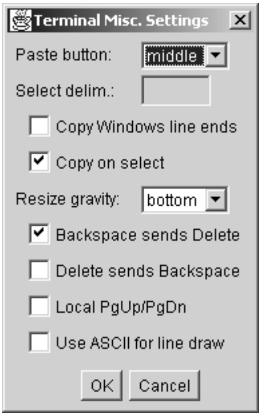
Scrollback buffer: number of lines to save in scrollback buffer.

Scrollbar position: where to place scrollbar in terminal window.

The parameters set in this dialog are (names as given in section 5):

4.3.3 Terminal Misc... (Ctrl+Shift+M)

This dialog contains some additional settings for the terminal, as shown in fig. 7 below.



Terminal Misc. Settings

Paste button: middle

Select delim.:

☐ Copy Windows line ends

☑ Copy on select

Resize gravity: bottom

☑ Backspace sends Delete

☐ Delete sends Backspace

☐ Local PgUp/PgDn

☐ Use ASCII for line draw

Figure 7. Terminal Misc. Settings dialog box

The parameters set in this dialog are:

Paste button: which mouse button to use when pasting.

Select delim: delimiter characters for click-selection.

Copy Windows line ends: Check this checkbox if ‘end-of-line’ characters should be copied when copy/pasting.

Copy on select: Check this checkbox if selected text should automatically be copied to the clipboard.

Resize gravity: the location of the fixed point of the screen when resizing.

Backspace sends Delete: <Backspace> key sends ‘delete’.

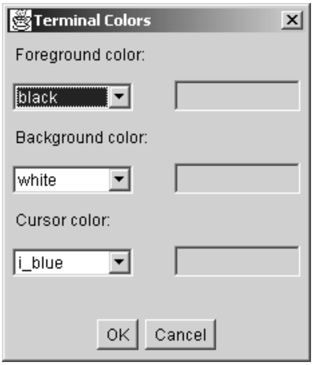
Delete sends Backspace: <Delete> key sends ‘backspace’.

Local PgUp/PgDn: Check this box if the local <PgUp> and <PgDn> keys should be activated.

Use ASCII for line draw: Check this box if ASCII line-draw characters should be used instead of drawing.

4.3.4 Terminal Colors...

This menu choice will bring up the screen shown in fig. 8 below.



Foreground color: black

Background color: white

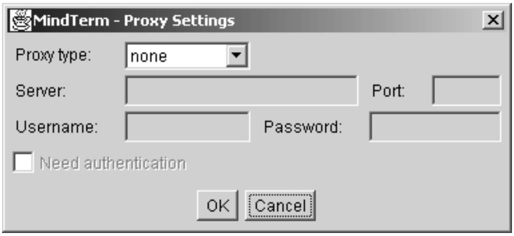
Cursor color: i_blue

Figure 8. Terminal Colors screen

The foreground color, background color, and cursor color may be chosen from the drop-down list boxes on this screen.

4.3.5 Proxy...

This menu selection will bring up the screen shown in fig. 9 on the next page.



Proxy type: none

□ Need authentication

Figure 9. Proxy Settings screen

The ‘Proxy Settings’ screen displays the following options:

Proxy type: Select what type of proxy server to use. The choices are none, http, socks4, and socks5.

Server: name of the proxy server to connect through.

Port: port on proxy server to connect through.

Username: user name on proxy server to connect as. The ‘Need authentication’ checkbox must be checked if login is required.

Password: password for user connecting. The ‘Need authentication’ checkbox must be checked if login is required.

4.3.6 Reset To Defaults

This menu selection resets any changed parameters to their initial default settings. These are the built-in defaults that have been overridden with the options set with the command-line/ applet parameters, or if a settings file is used with the ‘--f’ option at startup.

4.3.7 Auto Save Settings

This enables and disables the automatic saving of settings. When disabled, you must explicitly save the settings to a file when needed. When enabled, settings are saved whenever you disconnect from a server or when you exit AppGate MindTerm. Note that when both auto-save and auto-load is enabled (which is default), settings files are created automatically and the user never has to save and/or load them.

4.3.8 Auto Load Settings

This option enables and disables the automatic loading of settings. When disabled, you must explicitly load settings from a file if you need to. When enabled, AppGate MindTerm tries to load a settings file with the same name as what you give at the "SSH Server:" prompt or in the (Settings - SSH Connection...) dialog. These files are located in the AppGate MindTerm home directory. Thus, the "server" you give at the prompt does not necessarily have to be the name of the server; it is mainly the name of the settings file to load. Normally, the user does not have to worry about the settings files since they are handled automatically. However, to create short names for servers, and to create multiple settings files for a single server, you must explicitly create settings files.

4.3.9 Save Passwords

This option will enable the automatic saving and filling in of passwords used on the servers to which AppGate MindTerm connects. If this option is checked, passwords will be saved after their first use. After that, MindTerm will save them with the server settings.

4.4 VT Options Menu

This menu is the **Virtual Terminal** options menu. These control the appearance of the AppGate MindTerm screen. The options presented in this menu may be either ‘true’ or ‘false’. If the option is checked, it is considered ‘true’. If it is left unchecked, the option is considered to be ‘false’. The following options are available in the menu:

- Reverse Video
- Auto Wraparound
- Reverse Wraparound
- Insert mode
- Auto Linefeed
- Scroll to Bottom On Key Press
- Scroll to Bottom On Tty Output
- Visible Cursor
- Local Echo
- Visual Bell
- Map <CTRL>+<SPC> To ^@ (<NUL>)
- Toggle 80/132 Columns
- Enable 80/132 Switching

4.5 Tunnels Menu

4.5.1 Basic...

In this dialog, local tunnels may be set up. When connected, the tunnel is created instantly and is ready to use. Tunnels created here are saved in the settings file of the current session the settings files are in use. The protocol selection is mostly a convenience function. Please note, however, that to create FTP tunnels, the protocol should be set to FTP. Otherwise, the tunnel will not have the FTP plug-in enabled.

The local port to set is any unused port. This is the port that the programs using the tunnel will be pointed to. By default, tunnels will be set up to listen on all local addresses (i.e. 127.0.0.1 and the localhost address).

In the dialog "SSH Connection..." under "More options...", you can set the address to use as local address. For instance, if you want the tunnels to listen on 127.0.0.1, you can set that there. Also, using the "Advanced..." tunnels dialog (see section 4.4.2), you can set the local address on a per-tunnel basis. Using this method, AppGate MindTerm may be configured to have more than one tunnel on a single port using different local addresses.

The remote host is the address of the server that will answer connections to the tunnel on the SSH server end of the connection. Likewise, the remote port is the port on which it answers. To remove a tunnel, select it and click "Delete". To add a tunnel, fill in all the fields and click "Add". As a shortcut, double-clicking on a tunnel specification will copy its values to the fields, making it convenient to add, delete, and edit tunnels.

4.5.2 Advanced...

This dialog is mainly for advanced users who know the details concerning the use of SSH tunnels and their capabilities/limitations. Using this box, both local (as with the "Basic..." dialog) and remote tunnels may be set up. Note, however, that remote tunnels are not opened until the next time a connection is made.

The syntax for defining tunnels in this dialog is the same as with entering them on the command-line or as applet parameters (see section 5). For local tunnels, the explicit local address that the tunnel will listen on may be set, regardless of the setting of the "localhst" parameter. As in the "Basic..." dialog, double-clicking a definition-string will copy it to the edit box.

4.5.3 Current Connections...

This dialog lists the currently open connections through the tunnels that are set up. It does not list the tunnels themselves, only the active connections through them. A tunnel may be closed by selecting it and clicking close.

4.6 Help Menu

4.6.1 Help Topics...

This menu option has not been completed yet.

4.6.2 About MindTerm

Check here for information about AppGate MindTerm, especially build date, version, and the platform you are running for reference when reporting bugs to AppGate.

5 Parameters

When started either as an applet or as a stand-alone program, AppGate MindTerm is fully configurable. All configurable parameters may be set on the command-line (see section 6), or as applet parameters (see section 7). Additionally, when access to the local file system is available, all settings may be saved to a file on a per-server basis. Therefore, each new SSH server will have its settings in a separate file. This is, by default, done automatically if the local file system is accessible.

Table 1: SSH Parameters

Name	Description (values in parentheses when applicable)
server	Name of server to connect to.
real-server	Real address of sshd if it is behind a firewall; only used with protocol plug-ins.
local-bind	Address to use as localhost for forwards.
port	Port on server to connect to (0-65535).
username	User name to login as.
ssh1-cipher	Name of block cipher to use in ssh1 (blowfish-cbc/3des-cbc/idea-cbc)
auth-method	Method of authentication, either single or comma separated (password/publickey/tis/secureid/cryptocard/keyboard-interactive).
private-key	Name of file containing private key when using public key authentication.
display	Display definition (i.e. <host>:<screen>)
alive	Connection keep-alive interval in seconds (0-600, 0 means none)

Table 1: SSH Parameters

Name	Description (values in parentheses when applicable)
x11-forward	Indicates whether X11 display is forwarded or not (true/false)
password	Password for normal authentication (only saved if ‘Save Passwords’ is checked)
passphrase	Passphrase for public key keypair file (only saved if ‘Save Passwords’ is checked)
proxy-type	Type of proxy server to connect through (none/http/socks4/socks5)
proxy-port	Port on proxy server to connect through
proxy-user	Username if authentication is used on proxy server
proxy-password	Password if authentication is used on proxy server
sftpbridge-host	Interface to listen on in FTP to SFTP bridge (empty if disabled)
sftpbridge-port	Port to listen on in FTP to SFTP bridge
strict-hostid	Strict host key check; can only connect to known hosts
protocol	Preferred protocol (auto/ssh1/ssh2)
force-pty	Indicates whether to allocate a pty or not (true/false)

There are additional parameters available ONLY if the SSH2 protocol is used. The table below outlines those parameters.

Table 2: SSH2 Parameters

Name	Description
kex-algorithms	Kex algorithms to use in preferred order (diffie-hellman-group1-sha1/diffie-hellman-group-exchange-sha1)

Table 2: SSH2 Parameters

Name	Description
server-host-key-algorithms	Host key algorithms to accept in preferred order (ssh-rsa/ssh-dss)
enc-algorithms-cli2srv	Encryption algorithms client to server
enc-algorithms-srv2cli	Encryption algorithms server to client
mac-algorithms-cli2srv	Mac algorithms client to server
mac-algorithms-srv2cli	Mac algorithms server to client
comp-algorithms-cli2srv	Compression algorithms client to server
comp-algorithms-srv2cli	Compression algorithms server to client
package-version	Package version to send to server in protocol version

There are also parameters used to configure the AppGate MindTerm terminal. They are outlined in the table below.

Table 3: Terminal Parameters

Name	Description (values in parentheses when applicable)
rev-video	Reverse video in terminal (true/false)
autowrap	Autowrap of line if output reaches edge of window (true/false)
rev-autowrap	Reverse autowrap when exceeding left edge of window (true/false)
insert-mode	Toggles insert mode (true/false)
auto-linefeed	Enable auto-linefeed (true/false)
repos-input	Reposition scroll area to bottom on keyboard input (true/false)

Table 3: Terminal Parameters

Name	Description (values in parentheses when applicable)
repos-output	Reposition scroll area to bottom on output to screen (true/false)
local-pgkeys	Use PgUp, PgDn, Home, and End keys locally or escape them to the shell (true/false)
copy-crnl	Place <CR><NL> instead of <NL> at the end of lines in copy/paste (true/false)
visible-cursor	Toggles if cursor is visible or not (true/false)
ascii-line	Use ASCII line-draw characters instead of drawing (true/false)
local-echo	Enable local echo (true/false)
visual-bell	Toggles audible and visual bell (true/false)
map-ctrl-space	Map <Ctrl>+<space> to <NUL> where <NUL> = <Ctrl> + @ (true/false)
80x132-toggle	Toggle 80/132 columns (true/false)
80x132-enable	Enable 80/132 switching (true/false)
copy-select	Copy directly on mouse selection (true/false)
font-name	Name of font to use in terminal
font-size	Size of font to use in terminal
geometry	Geometry of terminal (‘<cols>x<rows>’)
term-type	Name of terminal to emulate (xterm, linux, scoansi, att6386, sun, aixterm, vt220, vt100, ansi, vt52, xterm-color, linux-lat, at386, vt320, vt102)
save-lines	Number of lines to save in the scrollback buffer (0-8192)
scrollbar	Scrollbar position (none/left/right)
bg-color	Background color (<name> or ‘<r>, <g>, <b>’)
fg-color	Foreground color (<name> or ‘<r>, <g>, <b>’)

Table 3: Terminal Parameters

Name	Description (values in parentheses when applicable)
cursor-color	Cursor color (<name> or ‘<r>, <g>, <b>’)
resize-gravity	Resize gravity, fixpoint of screen when resizing (top/bottom)
backspace-send	Character to send on BACKSPACE (BS/DEL)
delete-send	Character to send on DELETE (BS/DEL)
select-delim	Delimiter characters for click-selection (“<characters>”)
paste-button	Mouse button for paste (shift+left/middle/right)

There are also special parameters to configure the tunnels. They are the following:

`local0, local1, ... , localN`

`remote0, remote1, ... , remoteN`

Their syntax is as follows:

```
localN : [/<plugin>/][<local-ip>:]<local-
port>:<remote-ip>:<remote-port>
```

```
remoteN : [/<plugin>/]<remote-port>:<local-
ip>:<local-port>
```

The tunnel parameters are enumerated. Therefore, if you have three local forward definitions, they will be `local0`, `local1`, and `local2`. The same applies to `remoteN`. These properties are used in the exact same way as all other properties. They can either be entered on the command-line, as applet parameters, or in the settings files.

For example to set up tunnels to telnet, imap, and smtp on the local ports 4711, 4712, and 4713 to the remote side:

```
java -cp mindbright.jar
   mindbright.application.MindTerm -server
   www.mindbright.se -local0
   4711:localhost:23 -local1
   4712:localhost:143 -local2
   4713:localhost:25
```

NOTE: 'localhost' here means "locally" on the SSH server. The telnet, imap, and smtp servers all run on the same machine as the SSH server.

There is also an optional local command shell (activated with '--c' or 'cmdsh') where all settings can be viewed and/or altered. To enter this command shell, press ctrl-D at the prompt (i.e. before having logged in), or select the 'Local Command Shell' option in the 'Settings' menu. If the terminal is running in "dumb" mode, press ENTER after pressing ctrl-D.

The following is displayed when entering the command shell:

```
...entering local command-shell (type ‘h’ for help).

mindterm>
```

The following table presents the commands that are available in the command shell.

Table 4: Command Shell Commands

Command	Description
go	Start SSH session with current settings.
quit	Quit program (or disconnect if connected).
add <l | r> [/<plug>/]<port>:<host>:<port>	Add local/remote forward.
del <l | r><listen-port>|*	Delete local/remote forward (* = all).
list [ssh | term]	Lists SSH and/or terminal settings.
set [<parameter><value>]	Set value of an SSH parameter.
tset [<parameter><value>]	Set value of a terminal parameter.
key [<bits>]	Generate RSA key pair (of length <bits>).

Table 4: Command Shell Commands

Command	Description
help	Display this list.

6 Stand-alone Usage

When run as a stand-alone application, AppGate MindTerm takes two types of command-line options. One type is preceeded with a single hyphen ('-'). These are the parameters (see section 5) followed by their respective value. For example:

```
java -cp mindbright.jar
   mindbright.application.MindTerm -server
   www.mindbright.se -port 22 -x11fwd true -
   authtyp rsa
```

The other type of options are given with two preceeding hyphens ('--'). These are the special stand-alone options. When run with the standalone option '--?', the following is displayed:

```
usage: MindTerm [options] [properties] [command]
Options:
--d             No terminal window, only dumb command-line and port-
                forwarding.
--e             Exit MindTerm after logout (i.e. single session).
--f <file>      Use settings from the given file.
--h dir         Name of the MindTerm home dir (default: ~/mindterm/).
--m <no | pop | popN>
                Use no menus or popup (on mouse-button N) menu instead of
                menubar.
--p <save | load | both | none>
                Sets automatic save/load flags for property files.
--q             Quiet; do not query for server/username if given.
--v             Verbose; display verbose messages.
--x             Save passwords in encrypted property-files.
--D             Debug; display extra debug info.
--V             Version; display version number only.
--?             Help; display this help.
```

These are the valid stand-alone options.

The stand-alone options MUST be first among the Java command-line options (right AFTER the Java class-name). For example:

```
java -cp mindbright.jar
   com.mindbright.application.MindTerm --p
   both --h /home/mats/mindterm -server
   www.mindbright.se -port 22 -x11fwd true -
   authtyp rsa
```

NOTE: ‘-cp’ in this example is a command-line option to the Java Runtime.

The parameters (given with one preceding hyphen) are by default saved in settings files on a per-server basis. The settings files are automatically loaded when connecting to a specific server. The automatic save and load feature can be disabled, in which case settings must be explicitly loaded/saved. The settings file can also be manually edited as an ordinary text-file (Java properties file).

The following examples show how to start AppGate MindTerm as a stand-alone program:

Linux/jdk1.1.x:

```
/usr/local/java/bin/java -classpath /usr/local/java/lib/
classes.zip:mindtermfull.jarmindbright.application.MindTerm
```

Win32/jdk1.1.x:

```
c:\jdk1.1.x\bin\java -classpath
c:\jdk1.1.6\lib\classes.zip;c:\mindbright\mindtermfull.jar
mindbright.application.MindTerm
```

Win32/jre1.1.x:

```
c:\jdk1.1.x\bin\java -cp c:\mindbright\mindtermfull.jar
mindbright.application.MindTerm
```

Win32/jdk/jre1.2:

```
c:\jdk1.2.x\bin\java -cp c:\mindbright\mindtermfull.jar
mindbright.application.MindTerm
```

```
c:\jdk1.2.x\bin\javaw -cp c:\mindbright\mindtermfull.jar
mindbright.application.MindTerm
```

NOTE: The javaw runtime version does not create a DOS shell window for the console, making it more convenient for “real” usage.

Win32/jview (Microsoft’s JVM supplied with IE4 and later):

```
jview /cp:p mindtermfull.jar mindbright.application.MindTerm
```

MacOS/MRJ:

First, get the JBindery application. It is found in the MRJ SDK at the following website:

http://developer.apple.com/java/text/download.html#sdk

Then, drop the mindtermfull.jar file onto the JBindery icon and give it the class name mindbright.application.MindTerm. Save it and run AppGate MindTerm with just a double-click.

A command-line to run on the server when connected may also be specified. After all options on the command-line starting MindTerm, anything further is passed to the remote server as a command-line to run.

7 Applet Usage

As stated previously, all configurable parameters may be set with applet-params. For example:

```
<applet archive="mindterm.jar" code=com.mindbright.application.MindTerm.class width=580
height=400>
<!-- These parameters are parameters that are listed in paragraph 5. -->
<param name=port value="22">
<param name=cipher value="blowfish">
<param name=gm value="80x32+0-0">
<param name=forcpty value="true">
<param name=local0 value="4711:wintermute:23">
<param name=local1 value="/ftp/4712:wintermute:21">
<!-- Any parameters listed in section 5. can be included here -->

<!-- These parameters are special for the applet, most have an equivalent -->
<!-- command-line option when run as a stand-alone client -->
<param name=sepframe value="false"><!-- wheter to run in a separate frame or not -->
<param name=verbose value="true"><!-- output verbose debug-info to java-console -->
<param name=debug value="true"><!-- give more debug-info to java-console -->
<param name=quiet value="true"><!-- quiet mode, don't query for server/username if given -->
<param name=cmdsh value="true"><!-- enable/disable local command-shell -->
<param name=menus value="pop2"><!-- enable/disable pulldown or popup menus -->
<param name=autoprops value="both"><!-- enable/disable automatic save/load of settings -->
<param name=propsfile value="c:\ssh\ourserver.mtp"><!-- file containing settings (properties) to load -->
<param name=commandline value="mc -x -c"><!-- complete commandline if running a single command
only -->
<param name=sshhome value="c:\ssh\"><!-- If authorized to access local files, this is home-dir -->
<param name=appletbg value="black"><!-- Color of unused space in Applet's Panel -->
</applet>
```

Any number of parameters may be given to the applet. All of the parameters have default values, so no parameters MUST be specified in this file.

An applet may be run in basically three ways: with an applet-enabled browser, with a java plugin installed in a browser, or with a stand-alone applet viewer. All three ways are supported by AppGate MindTerm. However, the html-code for running an applet using a java plugin is not the same as for running it with an applet viewer or an applet-enabled browser.

Normally, applets are restricted to run within the “Java Sandbox” for security reasons. This puts some restrictions on what it can and cannot do. Basically, when run as an applet, AppGate MindTerm can only provide a login shell to the same

IP address that served the applet. In many cases, this can manually be extended so that it can access local files and provide SSH tunnels, etc.

The applet can be given a command to run on the server when connected, using the ‘commandline’ parameter, followed by the command to run.

Another way to run the applet without these restrictions is to use a cryptographically signed applet. In this case, the applet will function mostly like a normal stand-alone program.

8 Using FTP Tunnels

To use the FTP tunneling feature, a local tunnel that uses the FTP plug-in must be defined. AppGate MindTerm must then be connected to the tunnel using an FTP client that can be set to use "passive mode" transfers (most have this capability). One way to do this is to go to the (Tunnels -> Basic...) dialog and add a new tunnel with ‘protocol’ set to ftp, this automatically sets the remote port to 21, which is the standard FTP port on a Unix server. The local port is set to an arbitrary unused local port. The remote host is the address of the FTP server (as it is addressed from the SSH server). When the SSH server is connected, almost any FTP client may access the FTP server. For example, in WS_Ftp on Windows:

1. Define a new "site" with address localhost (or the address used for localhost. See 4.3.1 and 4.4.).
2. Go to "Site properties".
3. In "folder" advanced set "Remote Port:" to local port selected in AppGate MindTerm.
4. Enable "Passive transfers".

When WS_Ftp connects to this new site, it connects through the SSH tunnel in AppGate MindTerm. Therefore, the FTP server need not be reachable, for instance, if it is behind a firewall. To set up more than one FTP server behind the same SSH server, repeat the same procedure, selecting different local ports for each new server (in both AppGate MindTerm and WS_Ftp).

This can also be done using the ‘FTP to SFTP Bridge’ feature. From the FTP client’s perspective, connecting to a proxied FTP server and being “bridged” to an SFTP server is virtually the same. See 4.1.10 'FTP to SFTP Bridge...' on page 11.

When using SSH1, only passive-mode FTP clients can be used. In SSH2, both passive and non-passive (i.e. using PORT commands) can be used. However, due to restrictions and/or bugs in certain servers, it may be that only passive-mode works in some cases.

9 AppGate MindTerm Notes

All comments and bug reports should be sent to:

`mindterm@appgate.com`

Appendix A Connection Parameters

Name	Description
alive	Connection keep-alive interval in seconds (0 means none).
auth-method	Method of authentication, or if ‘custom list...’ is selected, a comma-separated list of methods to try in the order they are given. The options are: password, publickey, tis, secureid, cryptocard, and keyboard-interactive.
compression	Compression level (0=none, 1=fast, 9=slow/best)
display	The local X11 display to forward X11 connections to (i.e. <host>:<screen>).
exit-on-logout	Exit MindTerm after logout (true/false)
force-pty	Force allocation of PTY. This is necessary to enable when executing a single command on the SSH server that requires a non-dumb terminal.
local-bind	Default local address to bind to for forwards (see 4.4).
port	Port on server to connect to.
private-key	Name of file containing private key when using publickey as an authentication method.
real-server	Real IP address of SSH server if it is behind a firewall (used when ‘portftp’ is enabled)
server	Name (or IP address) of server to connect to.
ssh1-cipher	Name of block cipher to use in SSH1 (blowfish-cbc/3des-cbc/idea-cbc), or if ‘none’ is selected, no encryption. NOTE: ‘No encryption’ is normally not supported by the SSH server.
username	User name to login as on SSH server.
x11-forward	Allow X11 connections to be forwarded (true/false)

Name	Description (values in parentheses when applicable)
server	Name of server to connect to.
real-server	Real address of sshd if it is behind a firewall; only used with protocol plug-ins.
local-bind	Address to use as localhost for forwards.
port	Port on server to connect to (0-65535).
username	User name to login as.
ssh1-cipher	Name of block cipher to use in ssh1 (blowfish-cbc/3des-cbc/idea-cbc)
auth-method	Method of authentication, either single or comma separated (password/publickey/tis/secureid/cryptocard/keyboard-interactive).
private-key	Name of file containing private key when using public key authentication.
display	Display definition (i.e. <host>:<screen>)
alive	Connection keep-alive interval in seconds (0-600, 0 means none)
x11-forward	Indicates whether X11 display is forwarded or not (true/false)
password	Password for normal authentication (only saved if ‘Save Passwords’ is checked)
passphrase	Passphrase for public key keypair file (only saved if ‘Save Passwords’ is checked)
proxy-type	Type of proxy server to connect through (none/http/socks4/socks5)
proxy-port	Port on proxy server to connect through
proxy-user	Username if authentication is used on proxy server
proxy-password	Password if authentication is used on proxy server

Name	Description (values in parentheses when applicable)
sftpbridge-host	Interface to listen on in FTP to SFTP bridge (empty if disabled)
sftpbridge-port	Port to listen on in FTP to SFTP bridge
strict-hostid	Strict host key check; can only connect to known hosts
protocol	Preferred protocol (auto/ssh1/ssh2)
force-pty	Indicates whether to allocate a pty or not (true/false)

Appendix B SSH2 Parameters

Name	Description
kex-algorithms	Kex algorithms to use in preferred order (diffie-hellman-group1-sha1/diffie-hellman-group-exchange-sha1)
server-host-key-algorithms	Host key algorithms to accept in preferred order (ssh-rsa/ssh-dss)
enc-algorithms-cli2srv	Encryption algorithms client to server
enc-algorithms-srv2cli	Encryption algorithms server to client
mac-algorithms-cli2srv	Mac algorithms client to server
mac-algorithms-srv2cli	Mac algorithms server to client
comp-algorithms-cli2srv	Compression algorithms client to server
comp-algorithms-srv2cli	Compression algorithms server to client
package-version	Package version to send to server in protocol version

Appendix C Terminal Parameters

Name	Description (values in parentheses when applicable)
rev-video	Reverse video in terminal (true/false)
autowrap	Autowrap of line if output reaches edge of window (true/ false)
rev-autowrap	Reverse autowrap when exceeding left edge of window (true/false)
insert-mode	Toggles insert mode (true/false)
auto-linefeed	Enable auto-linefeed (true/false)
repos-input	Reposition scroll area to bottom on keyboard input (true/ false)
repos-output	Reposition scroll area to bottom on output to screen (true/ false)
local-pgkeys	Use PgUp, PgDn, Home, and End keys locally or escape them to the shell (true/false)
copy-crnl	Place <CR><NL> instead of <NL> at the end of lines in copy/paste (true/false)
visible-cursor	Toggles if cursor is visible or not (true/false)
ascii-line	Use ASCII line-draw characters instead of drawing (true/ false)
local-echo	Enable local echo (true/false)
visual-bell	Toggles audible and visual bell (true/false)
map-ctrl-space	Map <Ctrl>+<space> to <NUL> (true/false)
80x132-toggle	Toggle 80/132 columns (true/false)
80x132-enable	Enable 80/132 switching (true/false)
copy-select	Copy directly on mouse selection (true/false)

Name	Description (values in parentheses when applicable)
font-name	Name of font to use in terminal
font-size	Size of font to use in terminal
geometry	Geometry of terminal (‘<cols>x<rows>’)
term-type	Name of terminal to emulate (xterm, linux, scoansi, att6386, sun, aixterm, vt220, vt100, ansi, vt52, xterm-color, linux-lat, at386, vt320, vt102)
save-lines	Number of lines to save in the scrollback buffer (0-8192)
scrollbar	Scrollbar position (none/left/right)
bg-color	Background color (<name> or ‘<r>, <g>, <b>’)
fg-color	Foreground color (<name> or ‘<r>, <g>, <b>’)
cursor-color	Cursor color (<name> or ‘<r>, <g>, <b>’)
resize-gravity	Resize gravity, fixpoint of screen when resizing (top/bottom)
backspace-send	Character to send on BACKSPACE (BS/DEL)
delete-send	Character to send on DELETE (BS/DEL)
select-delim	Delimiter characters for click-selection (“<characters>”)
paste-button	Mouse button for paste (shift+left/middle/right)

Appendix D Ciphers, Macs, and Compressions Supported

MindTerm-supported ciphers:

- aes128-cbc
- blowfish-cbc
- twofish128-cbc
- aes192-cbc
- aes-256-cbc
- twofish-cbc
- cast128-cbc
- 3des-cbc
- idea-cbc
- arcfour

MindTerm-supported Macs (message authentication codes):

- hmac-md5
- hmac-sha1
- hmac-sha1-96
- hmac-md5-96
- hmac-ripemd160

MindTerm-supported compressions:

- none
- zlib