MiamiSSL

COLLABORATORS

	TITLE : MiamiSSL	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

MiamiSSL

1.1 MiamiSSL.guide

```
                MiamiSSL
********

   This is the documentation for MiamiSSL V2.11, an Secure Socket Layer
system for Miami. Copyright (C) 1996-1999 Nordic Global Inc.  All
rights reserved. Program and documentation by Holger Kruse.



                Disclaimer
                                        Legal information

                Usage / Copying
                                  Usage and copying conditions

                Requirements
                                     Required hardware and software

                Installation
                                     How to install MiamiSSL

                Use with web browsers
                                Use of MiamiSSL with web browsers

                Compatibility
                                    Compatibility with SSL servers

                Environment variables
                                Environment variables

                Utility Programs
                                   Other programs for MiamiSSL

                History
                                           History of MiamiSSL

                Support
                                           How to get help or updates
```

Acknowledgements

Acknowledgements

1.2 MiamiSSL.guide/NODE_DISCLAIMER

Disclaimer
****

MiamiSSL IS SUPPOSED TO BE AN SSL/TLS PACKAGE FOR Miami THAT ALLOW WEB BROWSERS AND OTHER PROGRAMS TO ESTABLISH SECURE, ENCRYPTED CONNECTIONS TO SSL-ENABLED WEB SERVERS.

EVEN THOUGH EVERY EFFORT HAS BEEN MADE TO MAKE MiamiSSL AS COMPATIBLE TO THE SSL/TLS STANDARDS AS POSSIBLE, I CANNOT RULE OUT THE POSSIBILITY THAT MiamiSSL HAS BUGS THAT HAVE HARMFUL SIDE EFFECTS ON YOUR SYSTEM OR ON OTHER MACHINES CONNECTED TO YOUR AMIGA.

I HEREBY REJECT ANY LIABILITY OR RESPONSIBILITY FOR THESE OR ANY OTHER CONSEQUENCES FROM THE USE OF MiamiSSL WHATSOEVER. THIS INCLUDES, BUT IS NOT LIMITED TO, DAMAGE TO YOUR EQUIPMENT, TO YOUR DATA, TO OTHER MACHINES YOUR AMIGA IS CONNECTED TO, ANY EQUIPMENT CONNECTED TO THOSE HOSTS, PERSONAL INJURIES, FINANCIAL LOSS OR ANY OTHER KINDS OF SIDE EFFECTS.

MiamiSSL IS PROVIDED AS-IS. THIS MEANS I DO NOT GUARANTEE THAT MiamiSSL IS FIT FOR ANY SPECIFIC PURPOSE AND I DO NOT GUARANTEE ANY BUG FIXES, UPDATES OR HELP DURING ERROR RECOVERY.

MiamiSSL CONTAINS ALGORITHMS THAT MAY BE ILLEGAL TO USE IN CERTAIN COUNTRIES, STATES, COUNTIES OR DISTRICTS BECAUSE OF LOCAL LEGISLATION. IT IS THE USER'S RESPONSIBILITY TO DETERMINE THE LEGALITY OF SUCH ALGORITHMS BEFORE USING THEM. THE FACT THAT MiamiSSL CONTAINS A CERTAIN ALGORITHM DOES NOT NECESSARILY MEAN THAT IT IS LEGAL TO USE THAT ALGORITHM IN ALL PARTS OF THE WORLD. I HEREBY REJECT ANY LIABILITY OR RESPONSIBILITY FOR ANY LEGAL CONSEQUENCES RESULTING FROM YOUR USE OF ANY ALGORITHMS IN MiamiSSL.

1.3 MiamiSSL.guide/NODE_CONDITIONS

Usage / Copying
****

MiamiSSL is a freeware add-on to Miami. It requires a legally registered version of Miami to be used.

The MiamiSSL binary or the binaries of any of the utility programs may not be modified or patched in any way (not even for personal use), except in ways explicitly approved by me for software updates. Using

patched or modified binaries is considered an act of software piracy !

MiamiSSL binaries may only be used for the purpose intended, i.e. to be executed on Amiga computers by AmigaOS, in combination with a legally registered version of Miami or MiamiDx. Reassembling, reverse-engineering, or translating binaries is expressly prohibited.

The documentation and program texts of MiamiSSL are subject to the same copyright as the program itself. This means neither documentation nor program texts may be modified or translated in any way.

To avoid any misunderstanding: YOU MAY NOT translate and distribute MiamiSSL program texts or documentation, unless I officially appoint you as a MiamiSSL translator. Unauthorized translations of program texts or documentation are illegal, violate my copyright, and will be deleted from public software sites.

If you want to distribute the MiamiSSL archive the following conditions apply:

* The sales price must not be higher than the cost of the empty disks required for the MiamiSSL files plus a nominal copying fee plus costs for shipping. The total price must not be higher than 10 US$ or 15 DM or the equivalent in any other currency.

* If the MiamiSSL archive is to be distributed as part of a CD-ROM collection of public domain and/or shareware programs, then the retail price of the CD-ROM may not exceed 20 US$, 30 DM or the equivalent in any other currency.

* All parts of the program and the documentation must be complete. The distribution of single parts or incomplete subsets of the original distribution is not allowed. The distribution of keyfiles is not allowed.

* MiamiSSL or parts of it may usually not be sold in combination with or as part of commercial software. Separate licensing conditions for commercial resale are available from kruse@nordicglobal.com upon request. However, unless and until you receive my explicit written approval, do not assume that you may distribute MiamiSSL or parts of it in combination or as part of commercial software.

* Program and documentation may not be changed in any way. Exception (this means: acceptable) is the use of archivers such as LHA as long as it remains possible to retrieve the original program/data.

Only the MiamiSSL main archive may be distributed at all, as described above. The encryption libraries (miamisslintl.library or miamisslusa.library) may not be distributed at all, neither in their patched form, nor in their original form (with the "orig" prefix to the file name). Encryption libraries may only be ordered and/or downloaded for personal use, on Amigas owned by the person or organization ordering or downloading the library.

VaporWare has explicit permission to distribute the original version

of miamisslintl.library via their web site, outside of the USA and
Canada.

1.4 MiamiSSL.guide/NODE_REQUIREMENTS

```
                Requirements
************
```

To use MiamiSSL you need:

* an Amiga running OS 2.04 or higher

* a registered version of Miami 2.1 or higher, or a registered version of MiamiDx.

* Miami keyfiles version 2 or higher. If you registered an old version of Miami then you may still have older version 1 keyfiles, and you need to upgrade. The upgrade from version 1 to version 2 is free of charge. Please see
  Installation
  for more details.

* The MiamiSSL main archive.

* One of the MiamiSSL encryption libraries. Please see
  Installation
  for information how to get them.

* reqtools.library version 38 or higher. You can get a current version of reqtools.library from Aminet.

* Software supporting MiamiSSL. At the moment Voyager-NG, IBrowse and AWeb-II support MiamiSSL in their latest release versions. Some other software that requires data encryption also makes use of MiamiSSL, e.g. the MS-CHAP password exchange algorithm in Miami/MiamiDx, and MiamiSecureShell, part of MiamiDx.

1.5 MiamiSSL.guide/NODE_INSTALLATION

```
                Installation
************
```

Here is a step-by-step overview of the installation:

* Ensure that you have reqtools.library V38 or higher installed. If you don't have it then get the ReqTools archive from Aminet (util/libs/ReqToolsUsr.lha).

* Make sure you are using Miami 2.1 or higher or Miami Deluxe. If

you have an older version of Miami installed then download Miami 2.1 or higher and install it "over" your current Miami installation.

* If you have not already done so: register Miami or Miami Deluxe. Your keyfiles have to be installed and working before you can install MiamiSSL.

* If you are using Miami 1.x or 2.x: In the "About" requester of Miami check if your keyfiles are already "Keyfile version 2" or higher (near the bottom of the requester). If you still have "Keyfile version 1", choose the upgrade "Upgrade keyfile version 1->2" in MiamiRegister, and wait until your new keyfiles have been installed. If you are using Miami 3.x or directly registered Miami Deluxe then your keyfiles will always be suitable for MiamiSSL, and you do not need to upgrade.

* Next you need to get one of the two MiamiSSL encryption library. For legal reasons these libraries are NOT included in the MiamiSSL main archive, but have to be downloaded or ordered separately. It is important that you order the CORRECT library. It is illegal for US/Canadian users to use the international library, and it is also illegal for users outside the US/Canada to order the US library.

  - Users in the USA/Canada: Using MiamiRegister, order the US encryption library. In MiamiRegister V2 this is done by selecting the fifth step in MiamiRegister. In MiamiRegister V3 select the third step ("Order files that are not freely available"). Wait for my email response, and install the library as described in the email.

  - Users outside the USA/Canada: Download origmiamisslintl.library, currently from www.vapor.com, and install the library as described in the archive.

  DO NOT rename any libraries, and DO NOT install libraries in "LIBS:". DO NOT ask me to send you the library by email. I am prohibited from doing so by law, i.e. you MUST get the library yourself, using the procedure described above. The library you will receive always has the version number 1.0. It will be patched to the latest version when you install MiamiSSL. There is no need to order a new library every time you install a new version of MiamiSSL.

* Download and unarchive the main MiamiSSL archive, and use the included installation script to install MiamiSSL.

* Read the section
  Compatibility
  , in particular the sections on

  Enabling ArcFour
  and
  Enabling IDEA
  . In a standard MiamiSSL installation some supported ciphers are disabled, because of legal restrictions. That may affect your ability to connect to some web

servers, especially in the US version. Depending on where you live and which licenses you own you MAY be allowed to use more ciphers, and the
Compatibility
chapter describes which ciphers are restricted in which version, under which circumstances you may enable them, and how to do that.

* Get your favorite web browser and install it. At the moment recent versions of VoyagerNG, IBrowse and AWeb-II have support for MiamiSSL.

* Depending on the browser you may have to perform some additional steps. Please consult the documentation of your web browser. For instance for IBrowse you need to make sure that the correct "https.library" is installed. Then set the SSL mode to "MiamiSSL" in the IBrowse configuration window. Voyager and AWeb-II should work with MiamiSSL "out of the box", without any changes. If you have previously used VoyagerNG with its own SSL implementation and now want to switch to MiamiSSL then delete or rename the file "Plugins/voyager_ssleay.vlib" in the VoyagerNG installation path. There is also some information in
Use with web browsers
.

1.6 MiamiSSL.guide/NODE_BROWSERS

Use with web browsers
****

At the moment the following web browsers support MiamiSSL:

VoyagerNG
VoyagerNG automatically detects and uses MiamiSSL for all URLs starting with "https://". The only thing you need to do is: if you installed VoyagerNG's own SSL encryption module "Plugins/voyager_ssleay.vlib" in the VoyagerNG installation path then you need to delete or rename that file. Otherwise VoyagerNG will use its own SSL implementation instead of MiamiSSL.

IBrowse
Not all versions of IBrowse correctly support MiamiSSL. Some versions do, but then some versions of the installation script do not install the correct libraries. I recommend you get the latest version of IBrowse and reinstall it "over" your current installation. Make sure you choose "MiamiSSL" during the installation, when Installer asks you which SSL implementation you want to use. After installing IBrowse open the IBrowse Network Preferences window, select the "Security" tab, and set the SSL module to "MiamiSSL". Then save the settings. After that you should be able to access "https://" URLs through MiamiSSL. If it still does not work then one of the IBrowse support libraries (most likely https.library) might have the wrong version. In that

case please contact IBrowse tech support for more help.

AWeb-II
As far as I know no special configuration of AWeb-II is necessary. It automatically uses MiamiSSL for "https://" URLs. You do, however, need the full commercial version of AWeb-II. The demo version does not support "https://" URLs at all.

1.7 MiamiSSL.guide/NODE_COMPATIBILITY

Compatibility
****

To answer the most frequently asked questions first:

Does MiamiSSL support 128-bit encryption ?
Yes, MiamiSSL DOES support 128-bit encryption, in its current version and in all previous versions, just like all other currently distributed SSL implementations for AmigaOS. If a web site you connect to complains about lack of 128-bit support in the browser then that message is bogus. The message does indicate a compatibility problem, but one of a different kind. It could be related to the SSL version, Java, JavaScript, browser names or something else.

Is MiamiSSL compatible with SGC (Server Gated Cryptography) ?
Yes, it is. On the client side SGC is really no different from normal SSL connections, except that SGC-compliant browsers released for PCs are intentionally crippled when used with non-SGC web servers. Amiga browsers with MiamiSSL always operate at the best possible encryption strength, regardless of whether the web server uses SGC, domestic SSL protocols at full strength or only export ciphers.

Does MiamiSSL support SSLv3 ?
Yes, in its current version MiamiSSL supports the SSLv2, SSLv3 and TLSv1 standards.

Older AmigaOS SSL implementations only supported SSLv2, which MAY have been the cause of some problems, including unexpected disconnects and bogus messages complaining about lack of 128-bit support. MiamiSSL SHOULD be compatible with all current SSL servers if all Ciphers are enabled and environment variables are set correctly. Please see

Unexpected disconnects
if you are unable to connect to a web server, and get an error message such as "Connection reset by peer" or "No common cipher found".

MiamiSSL currently support the ciphers and algorithms listed below. Many of them are supported in many different modes and combinations. To get a full list of supported cipher combinations in your installation type "Miami:MiamiSSLCiphers -v". See
MiamiSSLCiphers

for more
information.

Key exchange algorithms
Key exchange algorithms

Authentication algorithms
Authentication algorithms

Encryption algorithms
Encryption algorithms

MAC algorithms
MAC algorithms

Unexpected Disconnects
Unexpected Disconnects

Enabling ARCFOUR
Enabling ARCFOUR

Enabling IDEA
Enabling IDEA

1.8 MiamiSSL.guide/NODE_KEYEXCHANGE

Key exchange algorithms
==========

Before any encrypted data can be exchanged between client and server, both sides have to agree on a session key. That requires a secure key exchange. Several algorithms can be used for that.

DH
DH is a very fast, patent-free (patent expired) algorithm to exchange session keys. It is only supported by SSLv3 and TLSv1 though, not by SSLv2, and, unlike RSA, it does not provide any authentication.

RSA
RSA is the most widely implemented key exchange algorithm. It is always used with SSLv2, and can also be used with SSLv3 and TLSv1. RSA is patented in the US by RSADSI, and the US version of MiamiSSL uses a royalty-free non-commercial license based on the RSAref implementation. RSA not only exchanges keys, but can also be used for authentication.

For "export grade" ciphers the size of temporary DH/RSA keys used for the key exchange is limited to 512 bits, because of US Federal Regulations. For the US version of MiamiSSL and domestic "full strength" encryption the size of temporary RSA keys used for the key

exchange is limited to 1024 bits, because of restrictions on the use of
RSAref.

1.9 MiamiSSL.guide/NODE_AUTHENTICATION

```
Authentication algorithms
=========================

   Authentication algorithms are used to authenticate one end of the
connection to the other end. Usually only the server is authenticated
to the client, not the other way around.  Authentication is important
in order to prevent "man-in-the-middle" attacks, i.e. to ensure that
transparent web proxies cannot intercept sensitive, encrypted data
(such as credit card information) on the fly.

DSS
     DSS is the official standard for digital signatures, i.e.
     authentication.  It is only supported by SSLv3 and TLSv1 though,
     not by SSLv2, and, unlike RSA, cannot be used for key exchange.
     DSS is assumed to be patent-free by most Internet organizations,
     but the patent status is not completely clear.

RSA
     RSA is the most widely implemented authentication algorithm. It is
     always used with SSLv2, and can also be used with SSLv3 and TLSv1.
     RSA is patented in the US by RSADSI, and the US version of MiamiSSL
     uses a royalty-free non-commercial license based on the RSAref
     implementation. RSA not only provides authentication, but can also
     be used for key exchange.
```

1.10 MiamiSSL.guide/NODE_ENCRYPTION

```
                Encryption algorithms
=====================

   One of the following encryption algorithms is used for the actual
transfer or encrypted information such as web pages. The session key
for the algorithm is negotiated earlier, by
                Key exchange algorithms
                .

BLOWFISH
     BLOWFISH is a very fast, unpatented encryption algorithm with a
     key size of 128 bits. It is not currently supported by SSLv2,
     SSLv3 or TLSv1 though. It is only included in MiamiSSL because
     MiamiSecureShell can use it for SecureShell connections. Future
     versions of the TLSv1 standard are expected to get support for
     BLOWFISH.

DES
```

DES is the most widely used general-purpose data encryption algorithm, and an official US Government standard. It is patent-free (patent expired), rather slow if implemented in software, and uses a key size of 56 bits. A stronger variation of DES is 3DES, which triples the key size to 168 bits. Some software sometimes (incorrectly) reports those key sizes to be 64 and 192 bits, respectively. An export version of DES exists with a key size of 40 bits. DES and 3DES are supported by SSLv2, SSLv3 and TLSv1. DES is also used by the MS-CHAP password exchange algorithm.

IDEA

IDEA is a rather fast encryption algorithm with a key size of 128 bits. It is patented, with patents in many countries being held by ASCOM. IDEA is by default disabled in MiamiSSL because of the patent. For information on how to obtain a license so you can enable IDEA please see

Enabling IDEA

. IDEA is supported by SSLv2, SSLv3 and TLSv1.

ARCTWO

ARCTWO is a rather fast encryption algorithm with a key size of up to 128 bits. It is patent-free and has been in the public domain for a while. It is often refered to by its original name RC2(tm), a trademark in the US, held by RSADSI. ARCTWO is supported in full strength by SSLv2 and in export strength (40 bits) by SSLv3 and TLSv1.

ARCFOUR

ARCFOUR is a fast encryption algorithm with a key size of up to 128 bits. It is very widely used, patent-free and in the public domain. ARCFOUR is assumed to be compatible with the commercial cipher RC4(tm), a trademark by RSADSI. RSADSI considers the details of RC4 to be a trade secret, i.e. RC4 may not be used in the US without a (prohibitively expensive) license. ARCFOUR is supported by SSLv2, SSLv3 and TLSv1, in strengths between 40 and 128 bits. For a long time RC4 was the only cipher supported by export versions of web servers, so web browsers not supporting RC4 (or a compatible cipher such as ARCFOUR) could not connect to those servers. Many of those web servers which only support RC4 are still around. ARCFOUR is always enabled in the international version of MiamiSSL, but disabled in the US version of MiamiSSL by default, because in the US the legal status of ARCFOUR in relation to RC4 is not completely clear and may depend on the exact location where ARCFOUR is being used. If you want to use ARCFOUR with the US version of MiamiSSL then you need to enable it first. For more information on that see

Enabling ARCFOUR

.

1.11 MiamiSSL.guide/NODE_MAC

```
MAC algorithms
==============

   MAC (Message Authentication Code) algorithms ensure that data sent
from one end to the other end is not modified by attackers in transit.
These algorithms usually consist of a cryptographically strong hash
function, combined with a secret key. The following hash functions are
supported by MiamiSSL:

MD5
     MD5 generates a 128-bit hash code. It has been around for a long
     time and is generally considered secure. However recently some
     potential vulnerabilities have been discovered that, so far,
     cannot be effectively exploited though. Anyway those concerns are
     enough reason to not use MD5 any more for newer protocols.  MD5 is
     supported by SSLv2, SSLv3 and TLSv1.

SHA1
     SHA1 is a slightly modified version of SHA, a relatively new hash
     algorithm, and generates 160-bit hash codes. It is considered more
     secure than MD5 and has no known vulnerabilities. SHA1 is
     supported by SSLv3 and TLSv1.
```

1.12 MiamiSSL.guide/NODE_DISCONNECTS

```
                Unexpected Disconnects
======================

   It is possible that you get disconnected from a web server when
trying to access a secure web page. Usually your web browser will show
an error message such as "Connection reset by peer" or "No shared SSL
ciphers" in that situation. Sometimes you may also get a bogus error
message about lack of 128-bit support. The most likely causes for such
disconnects are:

   * You are using the US version of MiamiSSL, the web server you are
     connecting to only supports RC4, and you do not have ARCFOUR
     enabled in MiamiSSL. Please see
                Enabling ARCFOUR
                 for information
     on how to enable ARCFOUR in the US version of MiamiSSL. Enabling
     ARCFOUR is subject to legal restrictions and may not be available
     to you. If you do not fulfill the requirements for legal use of
     ARCFOUR then you have no way of accessing that web server with
     AmigaOS, sorry.

   * The web server you are connecting to does not handle the automatic
     protocol negotiation in TLS. There are two solutions to this:

    If you are using Miami 3.2b or earlier, or Miami Deluxe 0.8h or earlier
          Set the environment variable "ENV:Miami/SSLVERSION3" to "yes".
          This restricts MiamiSSL V2 to use SSLv3 only, i.e. inhibits
          the TLS negotiation. With that setting you can connect to
          some broken servers that do not handle TLS fallback correctly,
```

```
        but you will not be able to connect to servers that require
        SSLv2 or TLS. In order to connect to those servers you need
        to delete the environment variable again.

        Instead of setting "ENV:Miami/SSLVERSION3" to "y" you can
        also try setting ""ENV:Miami/SSLVERSION2" to "y". This
        restricts MiamiSSL to SSLv2 only, and disabled SSLv3 and TLS.

        Setting one of these two options should allow you to connect
        to any web server.

    If you are using a Miami version higher than 3.2b or a Miami Deluxe version ↩
       higher than 0.8h
        These versions support an automatic reconnect and fallback to
        v3 and v2 at the socket level, i.e. after a failed TLS
        connection attempt they allow MiamiSSL to connect to the same
        server again, in SSL v3 and v2 modes, without changing any
        environment variables or bothering the web browser or user
        about this. You SHOULD NOT need to set the SSLVERSION2 or
        SSLVERSION3 variable with these protocol stacks. Connections
        to v2 servers, v3 servers and broken servers should all happen
        automatically.

        Note: At the time of this writing these newer Miami and Miami
        Deluxe versions were not released yet, so please don't ask
        for them. Once they are released they will be made available
        on "www.nordicglobal.com".
```

1.13 MiamiSSL.guide/NODE_ENABLEARCFOUR

```
                Enabling ARCFOUR
================

   Enabling ARCFOUR is only an issue for users of the US version of
MiamiSSL. In the international version that cipher is always enabled,
so users of the international version can skip this part of the
documentation.

   The company RSADSI has been shipping the commercial cipher RC4(tm),
trademark by RSADSI, with their commercial BSAFE SSL toolkit for a long
time. RC4 also has been one of the few ciphers for which the US
Government has granted an export permit under "fasttrack" conditions.
That has made RC4 VERY popular among web servers, in particular export
versions of web servers. Many of these web servers ONLY support RC4, no
other cipher, allowing only web browsers that support RC4 to connect to
those web servers.

   The problem with that is that RC4 is not a "free" protocol. In the
US RC4 is considered a "trade secret" by RSADSI, making it illegal to
use software that supports RC4 in the US unless one obtains a license
for RC4. That license is so prohibitively expensive that the small
Amiga market does not create enough revenues to raise the money for
such a license. Because of that no AmigaOS SSL implementation currently
```

supports RC4.

In 1994 an algorithm that is supposedly compatible with RC4 was posted on Usenet. That algorithm was later published in many places (textbooks, magazines etc.) under the name "ARCFOUR", without patent or copyright. Experience shows that the algorithm can indeed be used as a compatible plug-in replacement for RC4. This algorithm is supported by all versions of MiamiSSL. In the international version it is always enabled. In the US version it is disabled by default, but can be enabled manually.

Whether ARCFOUR is legal to use in the US has been subject to considerable debate. The key point is whether the publication of ARCFOUR invalidated the "trade secret" claim by RSADSI or not. Opinions on this differ. One of the points of contention is whether the amount of algorithmic reverse-engineering necessary to come up with ARCFOUR is a legal method of "gaining access" to a trade secret. Apparently in some US states that method is considered legal, and publishing an algorithm obtained that way effectively defeats the trade secret, whereas in other states that method is illegal, and the original algorithm remains under trade secret status afterwards, which would make the use of ARCFOUR illegal.

I recommend you try to get along without ARCFOUR, if possible. If you absolutely need ARCFOUR then I strongly recommend you get professional legal advice on this matter first, taking into account the laws in your area.

In any case, if you DO decide to enable ARCFOUR then you bear the full responsibility for this. You have hereby been duly informed that Nordic Global Inc. considers the use of ARCFOUR in the US legally questionable, and is advising against it. Nordic Global Inc. will not be responsible or liable for the consequences of such actions by end users, and by enabling ARCFOUR in MiamiSSL you agree to hold Nordic Global Inc. harmless of any claims resulting from your use of ARCFOUR.

To enable ARCFOUR you need the program "MiamiSSLUSAArcFour". Please see

MiamiSSLUSAARcFour
for more information.

1.14 MiamiSSL.guide/NODE_ENABLEIDEA

Enabling IDEA
==========

IDEA is a rather fast 128-bit encryption algorithm, considerably faster than 3DES, with a similar level of security. It is not widely used in web servers, but some web servers do support it. You never absolutely need IDEA to access any web site (because most PC browsers do not support it either), but if you often connect to web servers that do not support ARCTWO or ARCFOUR, but only 3DES and IDEA, then you may be able to speed up your connection by enabling IDEA, because then the

slower 3DES does not have to be used.

IDEA is patented in many countries by the company "ASCOM" and is included in MiamiSSL in compliance with paragraph 5 of ASCOM's licensing policy (http://www.ascom.ch/infosec/idea/policy.html), allowing non-commercial software products to be distributed with IDEA without a license, subject to licensing by the end user. We are hereby providing users with the necessary information regarding the licensing status, as required by paragraph 5 of ASCOM's licensing policy.

This means MiamiSSL does NOT contain an IDEA license. YOU need to obtain one if you live in one of the countries where IDEA is patented and want to use IDEA with MiamiSSL.

Currently ASCOM offers non-commercial licenses for free and commercial licenses under different conditions. A commercial "end user license" currently costs US$ 15.00. If you want to use IDEA then you should check with ASCOM (http://www.ascom.ch/infosec) to find out if you need a commercial license. If you do then you need to buy one from ASCOM before using IDEA in MiamiSSL. You can buy a license directly on ASCOM's web site.

In any case, you have hereby been duly informed by Nordic Global Inc. that IDEA is shipping in MiamiSSL without a license, and that if you decide to enable IDEA it is YOUR responsibility to obtain the required license from the patent holder for your own personal use. Nordic Global Inc. will not be responsible or liable for the consequences of your use of IDEA, and by enabling IDEA in MiamiSSL you agree to hold Nordic Global Inc. harmless of any claims resulting from your use of IDEA.

To enable IDEA you need the program "MiamiSSLIDEA". Please see

MiamiSSLIDEA
for more information.

1.15 MiamiSSL.guide/NODE_ENVVARS

Environment variables
****

MiamiSSL uses the following environment variables:

MIAMI/SSLLIB
This variable points to the encryption library MiamiSSL uses. It is usually set to either "Miami:Libs/miamisslintl.library" or "Miami:Libs/miamisslusa.library". The variable is set automatically during installation and should not be modified.

MIAMI/SSLVERSION2
If this variable is set to "y" then MiamiSSL only uses SSLv2 for connection, instead of going through the usual SSLv2/SSLv3/TLSv1 negotiation. See

```
                Unexpected Disconnects
                 for more information when
     to set this variable.

MIAMI/SSLVERSION3
     If this variable is set to "y" then MiamiSSL only uses SSLv3 for
     connection, instead of going through the usual SSLv2/SSLv3/TLSv1
     negotiation. See
                Unexpected Disconnects
                 for more information when
     to set this variable.
```

1.16 MiamiSSL.guide/NODE_UTILITY

```
                Utility programs
****************

   The following utility programs are included in the MiamiSSL archive
and installed in the "Miami:" directory during the installation of
MiamiSSL.



                MiamiSSLCiphers
                                              MiamiSSLCiphers

                MiamiSSLClient
                                               MiamiSSLClient

                MiamiSSLIDEA
                                                 MiamiSSLIDEA

                MiamiSSLUSAArcFour
                                           MiamiSSLUSAArcFour

                MiamiSSLVersion
                                              MiamiSSLVersion
```

1.17 MiamiSSL.guide/NODE_MIAMISSLCIPHERS

```
MiamiSSLCiphers
===============

   MiamiSSLCiphers displays all ciphers available with the current
setup of MiamiSSL. Available options:

-v
     Verbose output. Display detailed cipher explanation instead of
```

```
     brief, ':'-separated list of ciphers.

-ssl2
     Display only ciphers available in SSLv2 mode.

-ssl3
     Display only ciphers available in SSLv3 mode.
```

1.18 MiamiSSL.guide/NODE_MIAMISSLCLIENT

```
MiamiSSLClient
==============

   MiamiSSLClient is a utility that allows you to establish connections
to SSL servers (e.g. secure web servers), and monitor the connection
establishment phase and all session parameters, e.g. encryption
strength, certificates, ciphers, SSL versions etc.

   MiamiSSLClient first establishes a connection and displays all
related parameters, and then waits for additional data from the server,
until a timeout is reached (usually 300 seconds). You can interrupt
MiamiSSLClient earlier by pressing Ctrl-C.

   Available options:

-host (hostname)
     Host to connect to. The default is "localhost".

-port (portnumber)
     Port number to connect to. The default is 443 (https).

-connect (host:port)
     Combination of "-host" and "-port" options

-verify (depth)
     Enables peer certificate verification and sets the maximum depth
     of certificate chains.

-cert (filename)
     Filename of client side certificate. PEM format assumed.

-key (filename)
     Filename of private key for client side certificate. Only
     necessary if the client side certificate does not already contain
     the private key. PEM format assumed.

-CApath (pathname)
     Pathname of certificate repository. PEM format assumed.

-CAfile (filename)
     Filename of certificate repository. PEM format assumed.

-reconnect
     Try to disconnect and reconnect five times with the same session
```

```
     ID. For debugging purposes. Does not work with all servers.

-showcerts
     Show all certificates in the chain.

-debug
     Show additional debugging information, such as packet hex dumps.

-state
     Show all internal SSL states.

-quiet
     Suppress most output.

-ssl2/-ssl3/-tls1
     Only use the specified SSL/TLS version.

-no_tls1/-no_ssl2/-no_ssl3
     Reject negotiation of the specified SSL/TLS version.

-bugs
     Enable all workarounds for bugs in SSL servers. Try this option if
     you get an error message otherwise.

-cipher
     Specify the cipher to use, e.g. "DES-CBC3-MD5".
```

1.19 MiamiSSL.guide/NODE_MIAMISSLIDEA

```
                MiamiSSLIDEA
============

   Before using this program please read the important background
information in
                Enabling IDEA
                .

   MiamiSSLIDEA enables or disables the IDEA cipher in the currently
installed MiamiSSL encryption library. The program patches the
encryption library, i.e. the change is permanent and does not have to be
repeated after rebooting. However each time you upgrade MiamiSSL to a
new version IDEA is automatically disabled again.

   To enable or disable the IDEA cipher just run MiamiSSLIDEA in a
shell window and follow the dialog.
```

1.20 MiamiSSL.guide/NODE_MIAMISSLUSAARCFOUR

```
                MiamiSSLUSAArcFour
==================

   Before using this program please read the important background
information in
                Enabling ARCFOUR
                .

   MiamiSSLArcFour enables or disables the ARCFOUR cipher in the US
version of the MiamiSSL encryption library, if it is currently
installed. If you are using the international version of the MiamiSSL
encryption library then this program has no use for you.

   The program patches the encryption library, i.e. the change is
permanent and does not have to be repeated after rebooting.  However
each time you upgrade MiamiSSL to a new version ARCFOUR is
automatically disabled again.

   To enable or disable the ARCFOUR cipher just run MiamiSSLUSAArcFour
in a shell window and follow the dialog.
```

1.21 MiamiSSL.guide/NODE_MIAMISSLVERSION

```
MiamiSSLVersion
===============

   MiamiSSLVersion displays version information about the currently
installed version of MiamiSSL. Available options:

-v
     Display version information (default).

-b
     Display build date.

-o
     Display build options (currently no effect).

-f
     Display build flags (currently no effect).

-a
     Display all of the above.
```

1.22 MiamiSSL.guide/NODE_HISTORY

```
History
*******
```

```
2.11
        * Bug fix: "MiamiSSLCiphers ?" would busy-loop.

        * Bug fix: Certificate verification did not work, because the
          hash function changed from SSLeay to OpenSSL, and MiamiSSL 2.0
          still used the old codes. New certificates are now installed
          in "Miami:SSL/v2certs".

        * Updated to latest OpenSSL 0.9.3 snapshot.

        * Added ARCTWO cipher back into US version.

        * Added option to enable ARCFOUR cipher in the US version.

        * Added option to enable IDEA cipher in all versions.

        * Added utility programs MiamiSSLUSAArcFour, MiamiSSLIDEA and
          MiamiSSLClient.

        * Added AmigaGuide documentation.

        * Improved version negotiation fallback to also allow fallbacks
          to SSLv3 for compatibility with a wider range of non-compliant
          servers.

2.0
        * Migrated sources to OpenSSL 0.9.3 snapshot.

        * Added support for SSLv2 and SSLv3

1.5
        * Added BLOWFISH cipher for MiamiSecureShell.

1.4
        * Extended API to support functions needed by MiamiSecureShell.

1.3
     Minor changes only.

1.2
        * Extended API to support functions needed for MS-CHAP.

1.1
     Minor changes only.

1.0
     Initial release.
```

1.23 MiamiSSL.guide/NODE_SUPPORT

```
Support
*******
```

```
   There are several ways to get technical support, updates etc.:

email
     kruse@nordicglobal.com

snail mail
     Nordic Global Inc.
     Attn: Holger Kruse
     PO Box 780248
     Orlando FL 32878-0248
     USA

WWW
     http://www.nordicglobal.com

mailing lists
     send "SUBSCRIBE miami-talk-ml", "SUBSCRIBE miami-announce-ml",
     "SUBSCRIBE miamidx-talk-ml" or "SUBSCRIBE miamidx-announce-ml" in
     the body of a mail to "listar@nordicglobal.com". These are the
     mailing lists for Miami and Miami Deluxe. They are also used to
     discuss MiamiSSL.
```

1.24 MiamiSSL.guide/NODE_ACKNOWLEDGEMENTS

```
Acknowledgements
****************

   My sincere thanks go to

   * James Cooper, Steve Krueger and Doug Walker for the SAS/C
     development system and their great support.

   * Tim Hudson and Eric Young for SSLeay.

   * The OpenSSL team for their continued work on OpenSSL, derived from
     SSLeay.
```