CycloneSSH

CycloneSSH is a SSHv2 library dedicated to embedded applications. It can be used to operate network services such as remote shell and file transfer over an unsecured network. The authentication layer of SSH uses public-key cryptography to authenticate the remote machine. The transport layer of SSH provides confidentiality and integrity of data exchanged between the client and server.

Main Features

- SSH version 2.0 implementation
- Client and server modes of operation
- Password and public key user authentication methods
- Key exchange using Diffie-Hellman, ECDH, Curve25519 and Curve448 algorithms
- RSA, DSA, ECDSA, Ed25519 and Ed448 host key algorithms
- 3DES, AES, Camellia and Chacha20Poly1305 encryption algorithms
- Legacy support for RC4, IDEA and Blowfish encryption algorithms
- CBC, CTR and GCM encryption modes
- HMAC using SHA-1, SHA-256 or SHA512
- Elliptic Curve Cryptography (ECC) supported
- Flexible memory footprint. Built-time configuration to embed only the necessary features
- Portable architecture (no processor dependencies)
- The library is distributed as a full ANSI C and highly maintainable source code

Coming soon: SFTP and SCP application protocols

Supported Processors

- ARM7TDMI / ARM926EJ-S
- ARM Cortex-M3
- ARM Cortex-M4
- ARM Cortex-M7
- ARM Cortex-R4
- ARM Cortex-A5
- ARM Cortex-A8
- ARM Cortex-A9
- RISC-V
- MIPS M4K
- MIPS microAptiv
- PowerPC e200
- Coldfire V2
- RX600
- AVR32
- Xtensa LX6

Supported Compilers / Toolchains

- GNU GCC / Makefile
- Atollic TrueSTUDIO
- IAR Embedded Workbench
- Keil MDK-ARM
- Microsoft Visual Studio
- Segger Embedded Studio
- AC6 System Workbench for STM32 (SW4STM32)
- Atmel Studio
- Infineon DAVE
- Microchip MPLAB X
- NXP MCUXpresso
- Renesas e2Studio
- ST STM32CubeIDE
- TI Code Composer Studio (CSS)

Supported Operating Systems

- Amazon FreeRTOS
- ChibiOS/RT
- CMSIS-RTOS
- CMSIS-RTOS2 (RTX v5 and FreeRTOS)
- Keil RTX
- Micrium µC/OS-II
- Micrium µC/OS-III
- Segger embOS
- SYS/BIOS (TI-RTOS)
- Bare Metal programming (without RTOS)

Key Exchange Algorithms

- diffie-hellman-group1-sha1(w)
- diffie-hellman-group14-sha1(w)
- diffie-hellman-group14-sha256
- diffie-hellman-group15-sha512
- diffie-hellman-group16-sha512
- ecdh-sha2-nistp256
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521
- curve25519-sha256
- curve25519-sha256@libssh.org
- curve448-sha512

Host Key Algorithms

- ssh-dss(w)
- ssh-rsa(w)
- rsa-sha2-256
- rsa-sha2-512
- ecdsa-sha2-nistp256
- ecdsa-sha2-nistp384
- ecdsa-sha2-nistp521
- ssh-ed25519
- ssh-ed448

Encryption Algorithms

- arcfour128(†)
- arcfour256(†)
- idea-cbc(†)
- idea-ctr(†)
- blowfish-cbc(†)
- blowfish-ctr(†)
- 3des-cbc(w)
- 3des-ctr(w)
- aes128-cbc
- aes192-cbc
- aes256-cbc
- aes128-ctr
- aes192-ctr
- aes256-ctr
- camellia128-cbc
- camellia192-cbc
- camellia256-cbc
- camellia128-ctr
- camellia192-ctr
- camellia256-ctr
- seed-cbc@ssh.com
- seed-ctr@ssh.com
- aes128-gcm@openssh.com
- aes256-gcm@openssh.com
- chacha20-poly1305@openssh.com

MAC Algorithms

- hmac-md5(†)
- hmac-ripemd160@openssh.com(w)
- hmac-sha1(w)
- hmac-sha2-256
- hmac-sha2-512

Compression Algorithms

- none

(†) denotes insecure algorithms

(w) denotes weak algorithms

SSH Core

- RFC 4250: The Secure Shell (SSH) Protocol Assigned Numbers
- RFC 4251: The Secure Shell (SSH) Protocol Architecture
- RFC 4252: The Secure Shell (SSH) Authentication Protocol
- RFC 4253: The Secure Shell (SSH) Transport Layer Protocol
- RFC 4254: The Secure Shell (SSH) Connection Protocol

SSH Extensions

- RFC 3526: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
- RFC 4335: The Secure Shell (SSH) Session Channel Break Extension
- RFC 4344: The Secure Shell (SSH) Transport Layer Encryption Modes
- RFC 4345: Improved Arcfour Modes for the Secure Shell (SSH) Transport Layer Protocol
- RFC 4432: RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol
- RFC 4716: The Secure Shell (SSH) Public Key File Format
- RFC 5656: Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer
- RFC 6668: SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol
- RFC 8268: More Modular Exponentiation (MODP) Diffie-Hellman (DH) Key Exchange (KEX) Groups for Secure Shell (SSH)
- RFC 8270: Increase the Secure Shell Minimum Recommended Diffie-Hellman Modulus Size to 2048 Bits
- RFC 8332: Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol
- RFC 8709: Ed25519 and Ed448 Public Key Algorithms for the Secure Shell (SSH) Protocol
- RFC 8731: Secure Shell (SSH) Key Exchange Method Using Curve25519 and Curve448
- RFC 8758: Deprecating RC4 in Secure Shell (SSH)
- RFC draft: Camellia cipher for the Secure Shell Transport Layer Protocol
- RFC draft: The chacha20-poly1305@openssh.com Authenticated Encryption Cipher

SFTP

- RFC draft: SSH File Transfer Protocol (version 3)