

# SSHield

## Embedded Secure Shell (SSH / SECSH) Server and Client



SSHield is an embedded Secure Shell (IETF SECSH, formerly known as SSH) implementation with a full-featured suite of secure applications that are interoperable with all popular desktop,

server and embedded SSH implementations. SSHield enables secure communication over a public or insecure network using popular encryption and authentication

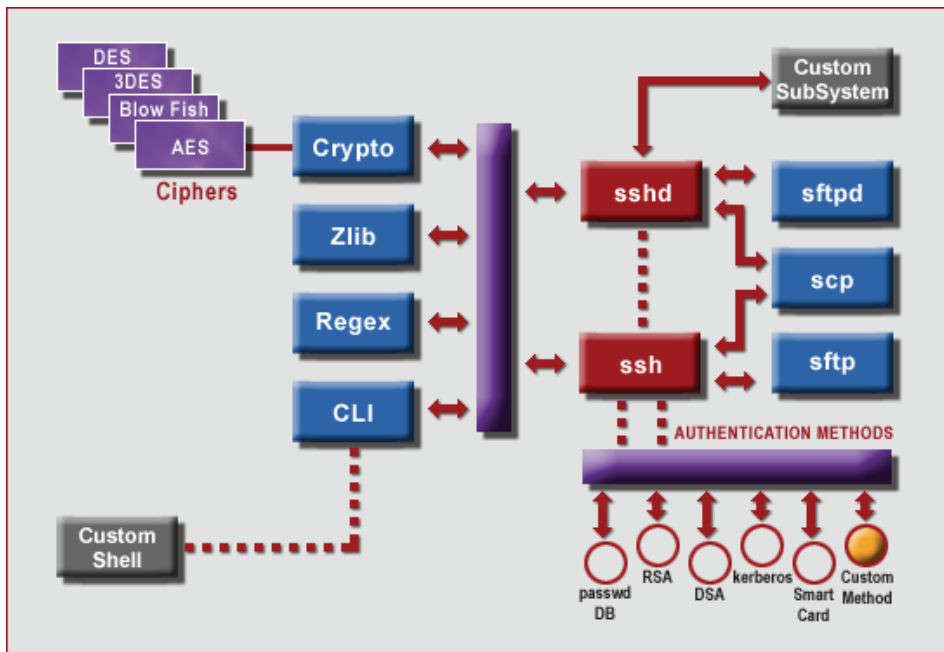


Fig 1: SSHield Component Architecture

### Special Features

- ❖ Includes server and client components for the SSH protocol as well as subsystems for SFTP and SCP
- ❖ Wide choice for encryption algorithms including AES (Rijndael), DES, 3DES, Blowfish, Twofish, CAST or Arcfour
- ❖ Overridable Pseudo Random Number Generator (PRNG)
- ❖ FIPS-certified cryptographic algorithms and FIPS 140-2 certification
- ❖ Target based key generation
- ❖ Extended support for digital certificate authentication
- ❖ Multi-tasking support
- ❖ Enhanced memory management & partition support
- ❖ Native support for VxWorks 5.3, 5.4.x, 5.5.x, and AE 1.x, Linux, QNX, pSOS and other OSes.

### Features

- ❖ Provides SSH protocol client and server support with both SSHv1 and SSHv2.
- ❖ Includes sftp client and server as well as scp with flexible library-style APIs.
- ❖ Supports password authentication in addition to public-key user authentication.
- ❖ X.509 certificate support for authentication.
- ❖ Support for Kerberos authentication.
- ❖ Supports custom authentication mechanisms.
- ❖ Modular crypto to scale out unneeded ciphers and hashes.
- ❖ APIs for target-based key generation.
- ❖ Data compression support.
- ❖ Port Forwarding for legacy applications and X11 Forwarding.
- ❖ Abstracted file IO system.
- ❖ Works with standard SecureShell client implementations on other platforms.
- ❖ Support for CPU types of either endian-ness including PowerPC, MIPS, X86, ARM/XScale.

techniques. It includes an SSH server and client, secure copy (scp), secure FTP client and server (sftp and sftpd), a built-in version of modular crypto libraries all of which can be scaled out when not in use. With advanced features such as X.509 digital certificate

support and Kerberos authentication, performance and memory optimizations for low-resource embedded environments. **SSHield** is an ideal fit for secure command-line management of any networked equipment and for securely transferring data and image files between field embedded devices and centralized servers.

## Cryptography Support

The **SSHield** implementation of the SSHv1 protocol uses RSA based authentication and encryption using public-key cryptography. **SSHield**'s SSHv2 protocol can use either RSA and DSA based authentication and provides additional methods for encryption. **SSHield** supports the following encryption ciphers and is further capable of supporting others from the included included crypto library or new ones as they are developed:

- ❖ AES
- ❖ 3DES
- ❖ CAST128
- ❖ Arcfour

**SSHield** also provides hmac-sha1 and hmac-md5 hashing methods for message integrity protection. **SSHield**'s included crypto library contains APIs to support popular hardware accelerators and dynamic embedded target-based key generation. Further, the cryptographic functionality, including the use of **X.509** certificates, is completely modular allowing for scaling out of unused ciphers for deeply scaled down memory footprints when **SSHield** is used.

## Authentication Support

Besides supporting public-key, **X.509**, and password based authentication out of the box, **SSHield** also includes hooks for customizing the authentication to plug in to various authentication standards such as **RADIUS**, **Kerberos**, or other proprietary authentication schemes including hardware tokens and biometric-based methods. Pre-tested integration with **TeamF1's AuthAgent Kerberos** as an optional authentication method allows for enterprise use of **SSHield**-enabled embedded devices in environments such as **UNIX® Kerberos**

- ❖ Available in full-source format.
- ❖ Configurable choice of encryption and authentication methods.
- ❖ Overridable PRNG functionality.
- ❖ Hooks to use configurable data-sources in lieu of file-systems.
- ❖ Configurability for proprietary external authentication mechanisms.
- ❖ Customizable hardware assist functionality.
- ❖ Complete scalability of unwanted components.

realms and **Microsoft® Active Directory** controlled networks.

## Port Forwarding

**SSHield**'s port forwarding is a powerful generic tunneling feature that allows the transparent and secure forwarding of TCP connections from one network node to another. Using this powerful mechanism, legacy insecure applications can be secured by redirecting traffic through the encrypted tunnel provided by **SSHield**. Security of the forwarded ports at the remote end can be further augmented by complementing the network security features of **SSHield** with a packet filtering firewall, such as **TeamF1's FireFly**, which gives fine-grained control over the accessibility of application ports from the public network, while simultaneously allowing full access from within the tunneling capabilities of **SSHield**. Where exposure of these ports is not as big a concern, **SSHield** contains built-in IP-level blocking facilities to restrict outside connections that originate from specific IP addresses.

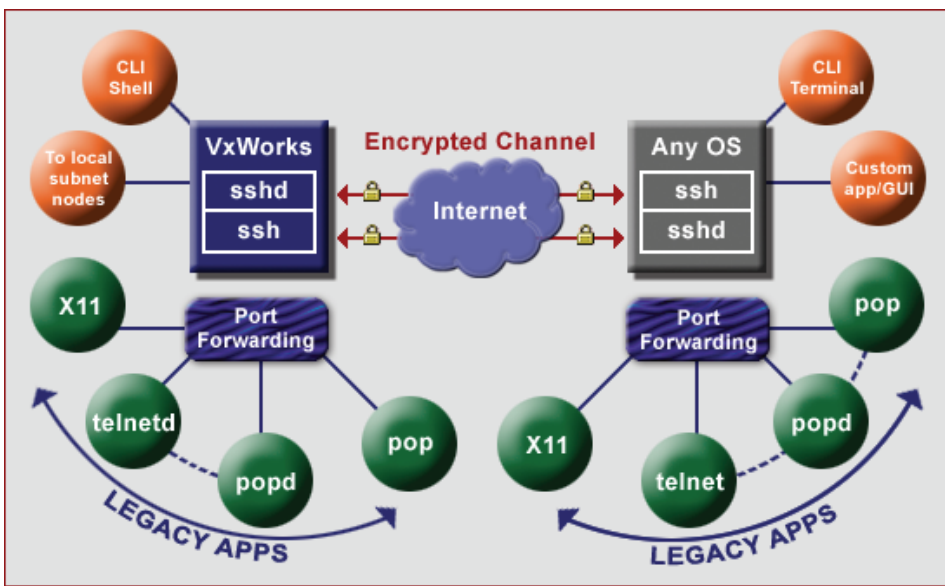


Fig 2: SSHield Applications

## Secure File Transfer

SSHield's flexible APIs to access the functionality of SFTP secure ftp (client and server) as well as SCP secure copy enables the use of secure file transfer functionality in embedded applications without tedious command line processing. An ftpLib style library API augments the standard standalone sftp/scp command usage and allows full access to the secure file transfer subsystems of the SSH protocol.

## Securing CLIs

For applications needing a new CLI layer, SSHield includes a utility function library to generate commands and hook them up to internal application management functionality with ease. For applications that need to secure an existing CLI, the CLI utility library can be scaled out easily to reduce resource requirements. SSHield also integrates well with existing CLI (command line interface) based device management frameworks that may already be in place. It has pre-defined interfaces for common management backplanes such as Rapid Control® CLI and WIND® Manage for CLI allowing for drop-in integration with these products, and can work with other CLI libraries including proprietary ones.

## Flexible IO

SSHield includes an optional abstract IO system to enable maximum flexibility for embedded devices that may not have a traditional file system, and yet require the use of secure file transfer capabilities, as well as to store and access keys from non file-system storage media. This, coupled with the ability to dynamically generate keys on the embedded device, greatly facilitates key management functions that may be needed by an embedded application.

## Interoperability

SSHield is compliant with the IETF definition of the SECSH protocol and is interoperable with freely available and commercial implementations of this protocol. It has been extensively validated against various SSH client & servers, SFTP clients & servers, and SCP implementations on embedded and non-embedded platforms including those on Windows®, Solaris®, UNIX®, and Linux. SSHield-enabled connected embedded devices can easily work with other SECSH implementations on a local network or across the Internet.

## Management Framework

SSHield provides API routines to

## Also Available

- ❖ **SSLimSecure SSL**  
*Secure Socket Layer*
- ❖ **V-IPSecure IPsec & IKE**  
*Network Layer Security*
- ❖ **FireFly**  
*IP-Filtering Firewall*
- ❖ **AuthAgent Kerberos / RADIUS**  
*Kerberos / RADIUS authentication*

## Custom Solutions

TeamF1's professional services team is available to help you put together customized implementations of SSHield solutions. They can also help with FIPS 140-2 certification of SSHield-enabled systems.

administer a database of permitted RSA and DSA keys, and to configure SSHield server options. Password authentication is managed by a table-driven mechanism, which can be manipulated programmatically as well. External authentication mechanisms such as those using smart cards, RADIUS, Kerberos and other custom methods are easily incorporated into the Secure Shell framework using configurable call-outs. Similar flexible hooks are provided for user-configurable data sources used by SFTP services instead of direct accesses to the file-system.



Fig 3: SSHield Port Forwarding

## A Quick Comparison of Network Security Technologies

Requirement	SSHield SSH	SSLimSecure SSL	V-IPSecure IPsec / IKE
Area of Security	Application	Transport/Session	Network
OSI Model Layer	L7	L4 / L5	L3
Secures legacy applications without modifications	✓	✗	✓
Typical area of usage	Management	Mgmt/Control	Data Path
Secures UDP (voice, NFS etc.)	✗	✗	✓
Standardized authentication mechanisms	✓	✓	✗
Extensible with authentication protocols	✓	✗	✓*
Supports User Authentication	✓	✓	Add-on
Suitable for sophisticated security policies	★★★	★	★★
Cross-platform interoperability	★★★	★★★	★★
Compatible with NAT & Firewalls	✓	✓	Limited
VPN capabilities	★	★★★	★★★
Per-connection/user end-to-end traffic flow confidentiality	✓	✓	✗
Protection against DoS attacks and traffic analysis	★★	★★★	★★★
Ease of provisioning	★★★	★★★	★★
Fine grained programmatic control	★★	★★★	★★★

\* Protocol does not provide standard interface for extending authentication mechanisms.

This product includes the OpenSSH software developed primarily by the OpenBSD Project and is a derivative version of the same.

This product includes cryptographic software from the OpenSSL cryptographic library written by Eric Young (eay@cryptsoft.com) and Tim J. Hudson (tjh@cryptsoft.com).

Email: [sales@TeamF1.com](mailto:sales@TeamF1.com)  
 Web: [www.TeamF1.com](http://www.TeamF1.com)  
 Ph: 510-505-9931 ext. 5  
 Fax: (510) 505-9941



© 2011 TeamF1, Inc.