



# SSHield

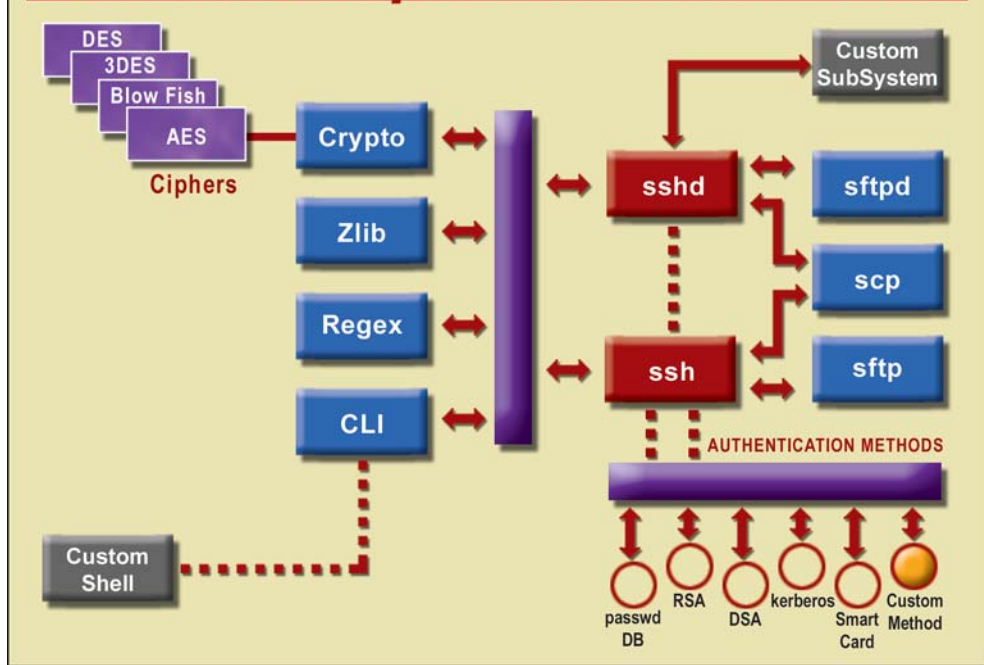
for pSOS+

Secure Shell

SSHield is a robust, standards based, small-footprint Secure Shell (IETF SECSH) implementation for pSOS® 2.x and 3.x. SSHield integrates the core server and client components needed to

implement a secure communication channel over insecure networks. Its unique, advanced features include a suite of secure applications such as secure copy (scp), secure FTP (sftp and

## SSHield Component Architecture



## pSOSystem Edition Features

- ❖ Designed exclusively for pSOS+
- ❖ Requires no special pSOS+ source modifications
- ❖ Integrated with pSOS+ and pRISM+ build & install methodology
- ❖ Easy Integration with BSPs
- ❖ Enhanced memory management & region support
- ❖ Multi-tasking support
- ❖ IO layer is built on top of standard pNA socket interface
- ❖ CLI integrated with pSH+
- ❖ Integrated with netutils library of pNA
- ❖ Native support for pSOSystem 2.x and 3.x

## 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 external 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.

sftpd), a built-in version of modular crypto libraries, support for X.509 digital certificates, and Kerberos authentication. It features enhanced memory management and optional CLI utility functions to get your secure application up & running in little time.

Given its ability to scale out optional features, SSHield is ideally suited for use in low-resource embedded environments.

### 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 crypto library or new ones as they are developed:

- ❖ AES (FIPS certified)
- ❖ 3DES (FIPS certified)
- ❖ CAST128
- ❖ Arcfour

SSHield also provides hmac-sha1 (FIPS certified) 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 and can be used by multiple security protocols, such as TeamF1's SSecure family of products, without any additional overhead.

### 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 realms and Microsoft® Active Directory and IAS controlled networks.

### Customization Flexibility

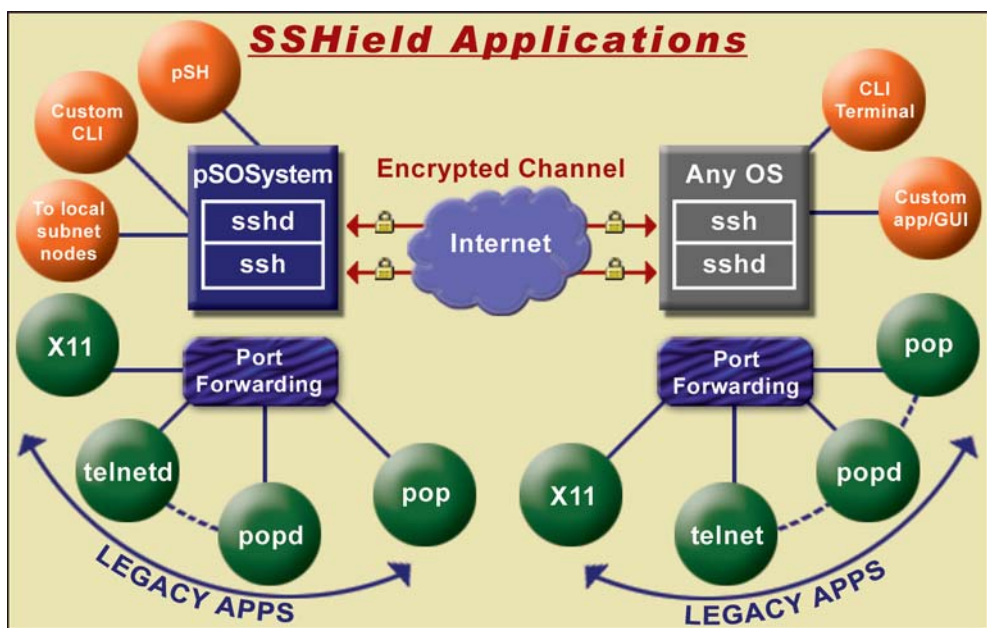
- ❖ Available in full-source format
- ❖ Configurable choice of encryption and authentication methods
- ❖ Overridable PRNG functionality
- ❖ Hooks to use configurable data-sources in lieu of file-systems
- ❖ Configurability for proprietary external authentication mechanisms
- ❖ Customizable hardware assist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

### 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.

### 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. A library-style 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 can work with any existing 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 administer a database of permitted RSA

### Also Available

- ❖ **SSLimSecure SSL**  
*Secure Socket Layer*
- ❖ **V-IPSecure IPsec & IKE**  
*Network Layer Security*
- ❖ **FireFly**  
*IP-Filtering Firewall*
- ❖ **ClassHopper**  
*Alternate Queuing Disciplines*

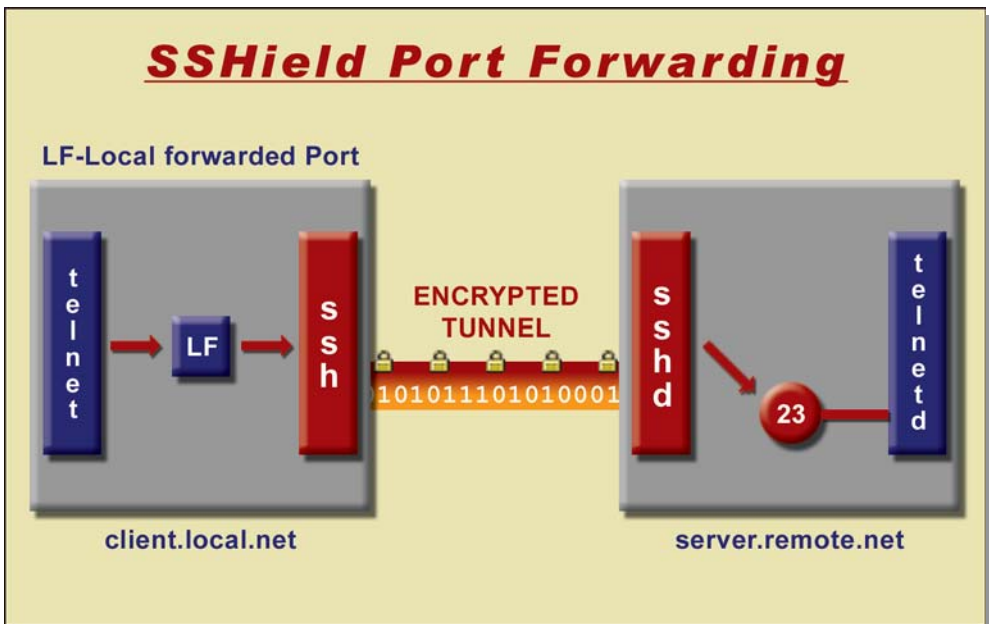
### Custom Solutions

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

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 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.

### Built for pSOSystem

**SSHield** is a drop-in component for *pSOS 2.x* and *3.x* and is seamlessly integrated in as an application component. It has been extensively validated on a variety of CPU architectures, and this minimizes development and integration efforts. **SSHield** is especially optimized for *pSOS+* with support for multi-tasking, memory regions & abstractions that are lean, yet fast. **SSHield** enables secure transactions in embedded network applications with the fewest changes.



## A Quick Comparison of Network Security Technologies

Requirement	SSHield SSH	SSLimSecure SSL	V-IPSecure IPsec / IKE
	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



© 2002-2004 TeamF1, Inc.