FASP Security Model

Bulletproof security for business-critical digital assets

Key benefits

- Standard open-source encryption supports alternative ciphers, if needed
- LDAP, Active Directory user authentication
- Encryption in transit and at rest helps ensure maximum security of business-critical digital assets
- Data integrity verification guards against man-in-the-middle, re-play, and UDP denial-of-service attacks

All IBM Aspera® products have complete built-in security for data transfers using the standard open-source OpenSSL toolkit. The Open SSL cryptographic libraries and the standard secure shell (SSH) are used unmodified in order to take full advantage of the standard. Aspera's products have been approved by the US Department of Commerce for export as a mass-market encryption product with >64 bit encryption. The security model consists of session encryption (to establish a secure channel for exchanging a random persession key for data encryption), secure authentication of the transfer endpoints, on-the-fly data encryption, and integrity verification for each transmitted data block. The transfer preserves the native file system access control attributes between any of the supported operating systems.

Session encryption

Each transfer job begins by establishing a secure, encrypted session between the endpoints, using the standard secure shell (SSH). SSH is invoked with its default symmetric cipher option for session encryption, 3DES (128 bits). SSH supports other ciphers for session encryption (e.g., 128 bit AES, Blowfish, CAST128, Arcfour, 192 bit AES, or 256 bit AES) and command line invocations of Aspera scp may request these alternative ciphers if supported by the peer ssh server. The particular algorithm used to negotiate the session encryption key depends upon whether SSHv-2 or SSHv-1 is used. SSH-v2 is the default for the sshd service built into Linux, Solaris and Mac OS X, and included with the Aspera distribution for MS Windows. However, Aspera scp can be run with SSH-v1 as a command line option (and also works with other commercial implementations of ssh). SSH-v2 uses a Diffie-Hellman key agreement to negotiate the session encryption key. In SSH-v1, each host has a host-specific RSA key (normally 1024 bits) and dynamically generates a new server RSA key (normally 768 bits) each time the ssh daemon starts up.

This key is normally regenerated every hour if it has been used, and is never stored on disk. When an ssh client connects, the daemon responds with its public host and server keys, and the client and server negotiate the session encryption key.

Authentication

Once the secure session channel is established, the transfer endpoints authenticate using one of the secure authentication mechanisms in ssh: interactive password or public-key. For public key authentication, the private keys are stored encrypted on disk using a secure, private passphrase and authentication is done using RSA only (SSH-v1) or RSA/DSA (SSH-v2) public key exchange. The ssh-keygen program is distributed with the Windows version of Aspera scp for generating DSA and RSA keys. The default key length is 1024 bits although the user may request longer key lengths.

Data encryption

Once SSH authentication has completed, the FASP® transfer session performs a three-way handshake during which the remote endpoint generates a random AES 128-bit per-session key for data encryption, and a random 128-bit key for computing an MD5 checksum, and sends these keys to the initiator over the secure ssh channel. A new encryption and MAC key is generated on each FASP transfer session, and the keys are never stored on disk.

FASP uses 128-bit AES encryption in which the key is re-initialized throughout the duration of the transfer using a standard CFB (Cipher Feedback) mode with a unique secret nonce (or “initialization vector”) for each block. CFB protects against all standard attacks based on sampling of encrypted data during long-running transfers.

The FASP source code includes support for ciphers in addition to 128-bit AES, and can be extended with other openssl ciphers such as AES 192. At this time, FASP does not expose command-line or GUI options for the end user to select a cipher other than AES 128, but could if needed, as the cipher code is modular.

Data integrity verification

An MD5 cryptographic hash function (128 bits) is applied to each encrypted datagram before transmission on the network. The resulting message digest is appended to the secure datagram and verified at the receiver for data integrity (to prevent man-in-the-middle, re-play, and UDP denial-of-service attacks).

Firewall considerations

Aspera server runs one SSH server on a configurable TCP port (22 by default; 33001 is often used). The firewall on the server side must allow this one TCP port to reach the Aspera server. No servers are listening on UDP ports. When a transfer is initiated by an Aspera client, it opens an SSH session to the SSH server on the designated TCP port and negotiates the UDP port (33001 by default) over which the data will travel. To allow the UDP session to start, the firewall on the Aspera server side must allow port UDP 33001 to reach the Aspera server.

Concurrent transfers considerations

Concurrent transfers on Aspera servers with multiple concurrent clients will:

- Share the same UDP port on UNIX
- Require a range of UDP ports (e.g., 33001-33100) to be allowed on Windows because the operating system does not allow Aspera’s FASP protocol to reuse the same UDP port for multiple connections. Incoming client connections will auto-increment to use the next available port in the range

In the case of point to point deployments of Aspera products, the end-points accepting incoming connections act as servers, and, therefore, their firewalls must allow TCP port 22 and UDP port 33001 (both configurable) to access the Aspera machine.

Client/server installations

Server side firewall must allow inbound connections to the server on the TCP port and on the UDP port. For Windows servers only, allow a range of ports large enough to cover the number of potential concurrent clients (e.g., 33001 through 33020, for 20 concurrent transfers). This is needed because Windows does not allow UDP port sharing. Server side firewall must also allow outbound connections from the server on the TCP port and on the UDP port (or range of ports for Windows servers).

On the client side, typical consumer and business firewalls allow direct outbound connections from client computers on TCP and UDP. There is no configuration required for Aspera transfers in this case. In cases where corporate firewalls disallow direct outbound connections (typically using proxy servers for web browsing), allow outbound connections from the Aspera client on the TCP port and on the UDP port.

Point to point installations

Consider two Aspera computers: A and B. A initiates the transfer (we call A client) and B accepts an incoming connection (we call B server). The client and server designations are given by the computer initiating the Aspera transfers, regardless of the direction of the transfer (upload or download).

On the client side (computer A), typical consumer and business firewalls allow direct outbound connections from client computers on TCP and UDP. There is no configuration required for Aspera transfers in this case.
In cases where corporate firewalls disallow direct outbound connections (typically using proxy servers for web browsing):

- Allow outbound connections from the Aspera client on the TCP port and on the UDP port
- Allow either:
  - inbound UDP traffic responding to the outbound UDP (this is default on most firewalls) or.
  - inbound UDP traffic on port 33001 (on non-standard firewall configurations)

On the server side (computer B), allow inbound connections from A on the TCP port and allow inbound and outbound UDP connections to B on the UDP port.

For A and B to act as both client and servers, both computers' firewalls must allow outbound and inbound connections to/from the peer on the TCP port, and allow outbound and inbound UDP connections to/from the peer on the UDP port.

Key features

- Built-in transfer security that uses standard open-source OpenSSL toolkit
- Secure, encrypted sessions using standard secure shell (SSH).
- User/endpoint authentication with Native File System Access Control support across all operating systems
- Data encryption in transit and at rest with AES-128 cryptography
- Data integrity verification for each transmitted block

Supported operating systems

- Windows 2000/XP/2003/2008, Windows Vista, Windows 7
- Mac OS version 10.4 and higher
- Linux
- Solaris
- Isilon OneFS

Firewall configuration summary

- Aspera transfers use one TCP port for session initialization and control, and one UDP port for data transfer
- Concurrent transfers on Windows require a range of UDP ports because Windows does not allow the use of one port for multiple connections

About IBM Aspera

IBM Aspera offers next-generation transport technologies that move the world's data at maximum speed regardless of file size, transfer distance and network conditions. Based on its patented, Emmy® award-winning FASP® protocol, Aspera software fully utilizes existing infrastructures to deliver the fastest, most predictable file-transfer experience. Aspera's core technology delivers unprecedented control over bandwidth, complete security and uncompromising reliability. Organizations across a variety of industries on six continents rely on Aspera software for the business-critical transport of their digital assets.

For more information

On IBM Aspera solutions, please visit us at https://www.ibm.com/cloud/high-speed-data-transfer or contact aspera-sales@ibm.com.

IBM Corporation
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
May 2018

Please Recycle

KUD12391-USEN-02