White Paper

DSS 5 Security Features

Date Written: 10/31/2014

Digital Sending Software 5

Security Features

Whitepaper

Table of Contents

Introduction

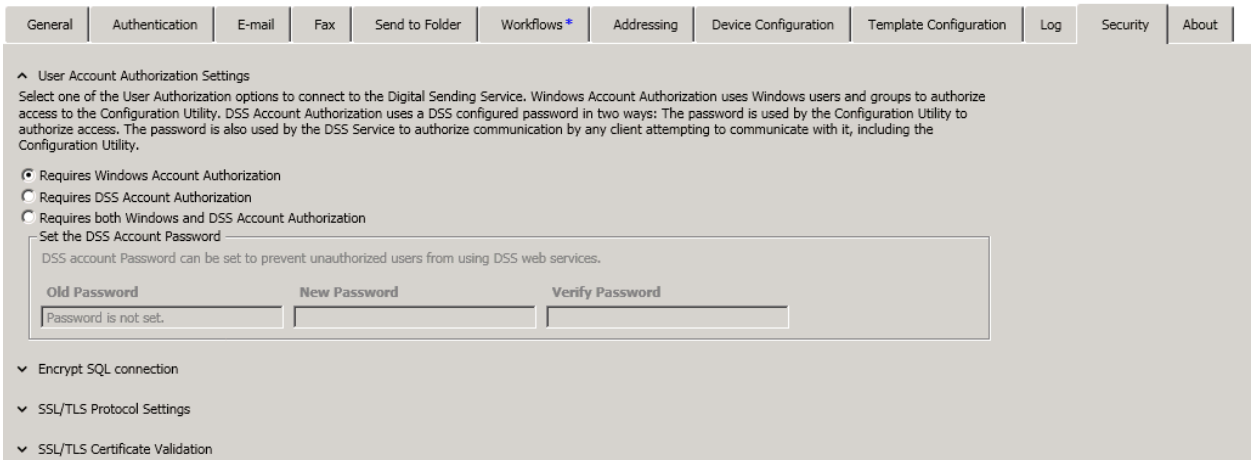
The purpose of this whitepaper is to give the reader a comprehensive view of data security mechanisms available with DSS 5.01.xx. Some of the security information is already available in the DSS System Administrator's Guide (SAG). This paper will refer to the SAG for topics that are currently documented there.

Many new features are not yet documented in the SAG. Those features are detailed in this paper. The intent is to move this information to the DSS SAG in the near future.

User Accounts and Passwords

Security to Run the Configuration Utility (CU) and Connect to the DSS Service

The DSS administrator can configure the type of security to use for opening the Configuration Utility and connecting to the DSS service. There are two mechanisms that can be used – Windows users and groups and using a DSS password. A screenshot of the UI in DSS is shown below.



Windows Account Authorization

Permission to start the Configuration Utility and connect to the DSS service can be controlled using Windows users and groups. This is the default security mechanism. Windows' account authorization controls access to the DSS service via the Configuration Utility, but does not provide access control to the DSS service via any other application than the Configuration Utility. For a full description of this capability see the System Administrator's guide, in the section "Security to start the Configuration Utility".

DSS Account Authorization

DSS allows the administrator to set a password for access to the DSS system service. When set, this password will be required by the DSS Configuration Utility as well as any other application that attempts

to access the DSS service. When DSS Account Authorization is enabled any user starting the Configuration Utility will be prompted for the configured password.

If there are 5 consecutive unsuccessful sign in attempts to the DSS service the service is locked from future sign in attempts for a period of time and a critical error email message is sent to the DSS administrator. The length of time the service is locked is set to 30 minutes by default and is configurable in the configuration file:

<install-folder>\Hewlett-Packard\HP Digital Sending Software\Filesystems\Product\Dss\Configuration\HP.Dss.App.Service.Config.xml.

Permissions required for a non-admin to run the CU with full functionality

Depending on the authorization settings chosen it is possible for users who are not Windows' administrators on the DSS server to run the Configuration Utility. But users who are not Windows' administrators will not have the system permissions needed to perform many of the tasks available via the Configuration Utility. Please see the System Administrator's Guide, the section entitled "Permissions needed to run DSS with full functionality", for details of granting permissions to non-Windows administrators to administer DSS with full functionality.

Device Credentials for FutureSmart Devices

FutureSmart devices can have passwords enabled by device administrators. When FutureSmart devices are password protected DSS must know the password in order to interact with the device. Please see the System Administrator's Guide, in the section "Device credentials for FutureSmart devices", for details on this functionality.

Data Encryption with Pre-FutureSmart Devices

When DSS interacts with pre-FutureSmart devices the data is encrypted before it is sent over the network. DSS and pre-FutureSmart devices use the Blowfish encryption algorithm with 128 bit encryption strength. There are no configuration options available for this encryption. DSS only uses Blowfish encryption when communicating with pre-FutureSmart devices.

Data Security Using SSL / TLS

Much of the data security in DSS is provided by utilizing the SSL / TLS protocols. This paper will give a brief overview of some basic SSL /TLS concepts and will then discuss specifics of how the protocols are used for communication between DSS components and between DSS and external entities including FutureSmart devices, SMTP servers and LDAP servers.

Brief Overview of SSL / TLS Protocols

There is a vast amount of information about the SSL / TLS protocols available on the web. This paper gives a brief, very high level, overview of some important SSL /TLS concepts, but the reader is referred to public information for a deeper understanding.

One good reference is a whitepaper created by the HP Jetdirect team. Jetdirect is the name of the network interface in HP printers and MFPs. The whitepaper gives a nice overview of SSL /TLS in general and a lot of specific information about configuring HP printers for use with SSL / TLS. The paper is titled "HP Jetdirect and SSL/TLS" and can be found here:
http://h20628.www2.hp.com/km-ext/kmcsdirect/emr_na-c01361514-2.pdf

The SSL / TLS protocols, when utilized, encrypt all the information being communicated between two endpoints on a communication channel. The encryption is done using a combination of symmetric and asymmetric cryptography. Asymmetric cryptography, which is more secure but much slower than symmetric cryptography, is used to generate symmetric cryptographic keys in a very secure fashion. The symmetric keys are then used for most of the communication channel encryption. In asymmetric key encryption the keys are distributed via certificates. The certificates are created by certificate authorities and an important part of the protocols involve trust in the authority that has created the certificates to be used in the communication.

Client and Server definitions

Whenever two entities, or endpoints, communicate using the SSL / TLS protocols one of the entities is the client and the other is the server. The client initiates the communication in order to communicate with the server. The communication will involve information flowing both ways between the client and server, but for any specific SSL / TLS session the client is the one that initiated the communication. When understanding what must be done at each endpoint to enable the SSL / TLS based communication it is important to know which endpoint is the client and which is the server.

DSS can be the SSL / TLS client or server depending on exactly what feature is being used. SSL / TLS sessions do not necessarily map 1:1 to larger tasks such as, for example, a FutureSmart device sending a job to the DSS server for processing. Large tasks like job processing can sometimes involve many SSL / TLS sessions. During some of these SSL/ TLS sessions the DSS server will be the SSL / TLS client and in others the DSS server can be the SSL /TLS server.

Asymmetric Cryptography

Asymmetric Cryptography uses key pairs that are mathematically linked. When data is encrypted using one of the keys of the pair it can only be decrypted using the other key in the pair. Using this mathematical concept to build a security structure involves an entity creating a key pair and making one of the keys of the pair publicly available to other entities while the second key of the pair is kept private. These are referred to as the public and private keys.

In the SSL / TLS protocols the server gives its public key to the client. The client uses this key to encrypt an encryption secret and sends the encrypted secret to the server. The server decrypts the secret with its private key. Using the secret the client and server can derive a symmetric encryption key and then encrypt data with the faster symmetric cryptography, but feel secure since the secret was shared using asymmetric cryptography.

Certificates and Certificate Authorities

For asymmetric cryptography public keys must be made available to clients. The keys are made available in certificates. Certificates contain the public key, information about the entity that holds the corresponding private key, and also information about the entity that created the certificate.

Certificate Authorities create certificates. When an entity creates its own certificate it is acting as its own certificate authority and the resulting certificate is called a self-signed certificate. When a third party creates a certificate for another entity the entity that creates the certificate is a Certificate Authority (CA). VeriSign is a well-known third party Certificate Authority. Whenever a certificate is created it is signed by the CA that created it, whether or not the CA is itself (self-signed) or a third party.

Certificate Authorities are rarely a single server or entity. In fact, many companies have set up one or more of their own public-key infrastructures to deal with public key distribution. Wikipedia defines a public-key infrastructure as: " A public-key infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store and revoke digital certificates". For the very simplified explanation of SSL /TLS in this whitepaper we will assume that the CA is just a simple single server, but, any instructions found in this paper referring to CAs or CA certificates may have to be modified to take into account the actual PKI infrastructure within which DSS is operating.

Certificate Authorities create their own self-signed certificates which contain the public key of the CA.

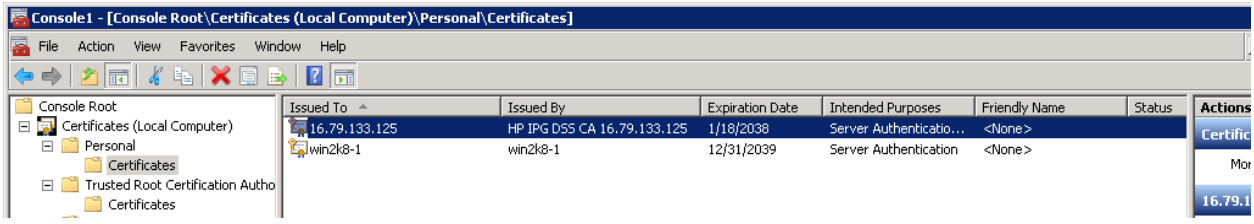
Since SSL / TLS clients get keys from servers an important aspect of the security structure is helping the client trust that the certificate it has been given is really coming from who they think it is, and not instead coming from a fake entity that is trying to break their security.

As an example, let's look at the DSS certificates that DSS creates for itself on a DSS server when it is installed.

To look at the certificates on a DSS server the Microsoft Management Console (MMC) is used with the certificates plug in. Follow these steps on the DSS server:

1. Start → Run → "mmc"
2. File → Add/Remove Snap-in
3. choose "Certificates" and press "Add" button
4. "Computer Account" → Next → "Local computer" → Finish
5. Press "OK" button
6. In tree view on left panel, expand "Certificates (Local Computer) / Personal / Certificates"
7. There should be a certificate with the name <IP address> in the "Issued To" column and HP IPG DSS CA <IP address> in the "Issued By" column

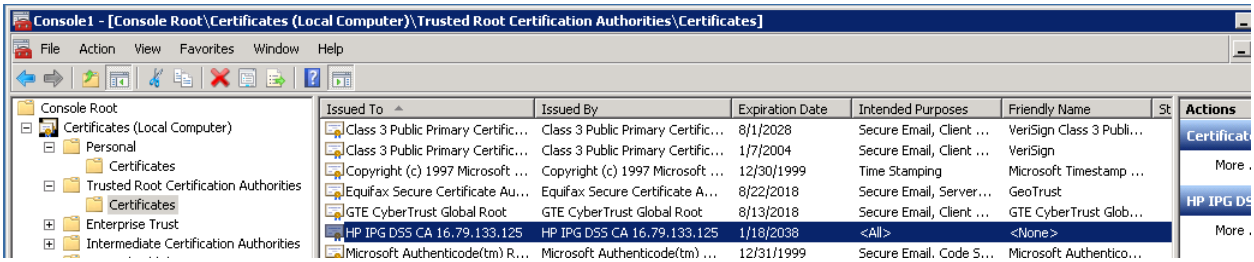
See the screenshot below.



The highlighted certificate is the DSS certificate that holds its public key. (A certificate is not created for the private key; the private key is not to be given out). DSS uses the IP address of the DSS server for

naming with certificates. Notice also that the certificate has been issued by HP IPG DSS CA <IP address>. The issuer is also DSS, but this is DSS acting in the role as a certificate authority.

Since DSS is also acting as a CA, let's look at the DSS certificate that is generated for DSS as a CA. This certificate is in the Trusted Root Certification Authorities Store.



The certificate in the Personal/Certificates store is distributed by DSS, to the client in an SSL / TLS session, when DSS is the SSL / TLS server in that session. The certificate in the Trusted Root Certification Authorities store is used to establish trust and is not automatically distributed.

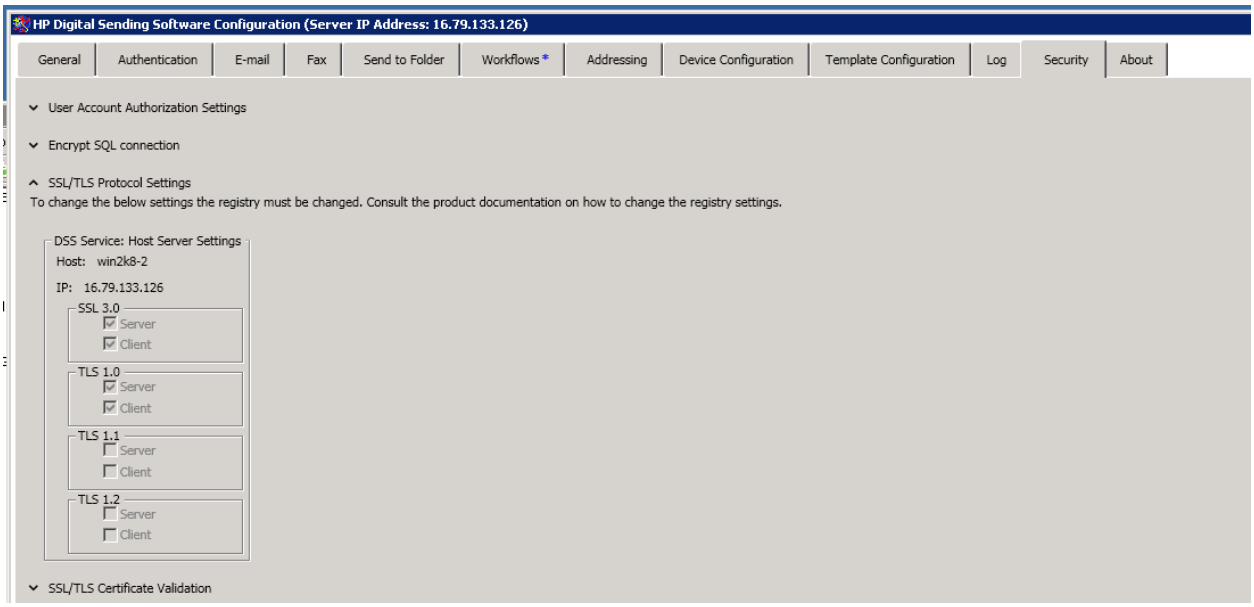
SSL/ TLS levels

SSL stands for Secure Socket layer. TLS stands for Transport Layer Security. SSL started with SLL 2.0 and there is also SSL 3.0. TLS is a later spec and considered an evolution of the same class of protocol. There are TLS versions 1.0, 1.1, and 1.2. As the protocols evolved they continue to improve and TLS 1.2 is now considered the most secure. SSL 2.0 is almost never used.

When a client contacts a server for an SSL / TLS session one of the first things they do is select what level of the protocol they will use to communicate. The level chosen is the highest (newest) that each endpoint supports.

Which levels of the protocol DSS can use is controlled by the underlying operating system. Whether or not a level is enabled is configured in the Windows registry. At the time this paper is being written the SSL 3.0 and TLS 1.0 levels are on by default and the other levels are not enabled. These defaults could change in the future. Changing the levels enabled on the OS can be done by direct registry editing or using one of several commercially available tools. There are many web articles that discuss how to enable and disable the various levels.

DSS does not provide a capability to change which levels of the protocol the underlying OS supports. If the DSS admin wants to change the current settings they should consult the available literature for how to do it. DSS does show which levels are enabled and disabled so the DSS admin can understand at which levels DSS can operate. This information is shown on the Security tab of the Configuration utility, as shown below. Note that the protocols are enabled or disabled independently for when the system is acting as a SSL / TLS client or server. Also note that when an administrator changes the available SSL / TLS protocols that can be used for a machine the changes apply to all applications running on that server, not just for DSS.



When using a remote Configuration Utility the settings will appear twice, once for the CU server and once for the DSS server.

IMPORTANT NOTE: At this time DSS is unable to function if only TLS 1.1 and /or TLS 1.2 are enabled and SSL 3.0 and TLS 1.0 are disabled. If the server is configured for only TLS 1.1 and/or 1.2 then the DSS service will fail to start. If this occurs, it can be remedied by once again enabled SSL 3.0 and / or TLS 1.0.

Server Certificate Validation

When an SSL / TLS client receives the server's certificate with the server's public key in it, the client has a decision to make. Should the client just trust that the certificate is OK or should it test to see that it can be trusted by validating the certificate?

Certificate Authority certificates are used for certificate validation. Each certificate is signed by the CA that created it. If an entity trusts that any certificates it might get that are signed by a particular CA, let's call it CA1, are good then the client should put the certificate for CA 1 in its trusted root certification authority's store. The flow would go something like this:

1. Entity 1, which will be the client in the SSL / TLS communication, trusts that any certificates it gets that are signed by CA1 are trustworthy.
2. Entity 1 obtains the certificate from CA1 and puts it into its own Trusted Root Certification Authorities store.
3. Entity 1 initiates an SSL / TLS session with entity 2, so Entity 1 is the client and Entity 2 is the server.
4. Entity 2 passes its public key certificate to entity 1, this certificate is signed by the CA named CA1.
5. Entity 1 gets entity 2's certificate and sees that it has been signed by CA1. Entity 1 does some checks to see if it can trust the certificate it received:
    a. A check to see if CA1's certificate is in its Trusted Root Certification Authorities store, which it is because it was put there in step 2 above.

b. A check to see if the name associated with the certificate exactly matches the name that entity1 used to contact entity2. For example, if entity 1 contacted entity 2 with its IP address, entity 2 would expect that IP address to be one of the names in entity 2's certificate. (Certificates have a primary name known as the common name or CN, and can also have alternate names stored in the SAN (Subject Alternative Name) part of the certificate.)
c. There are additional checks that won't be detailed here

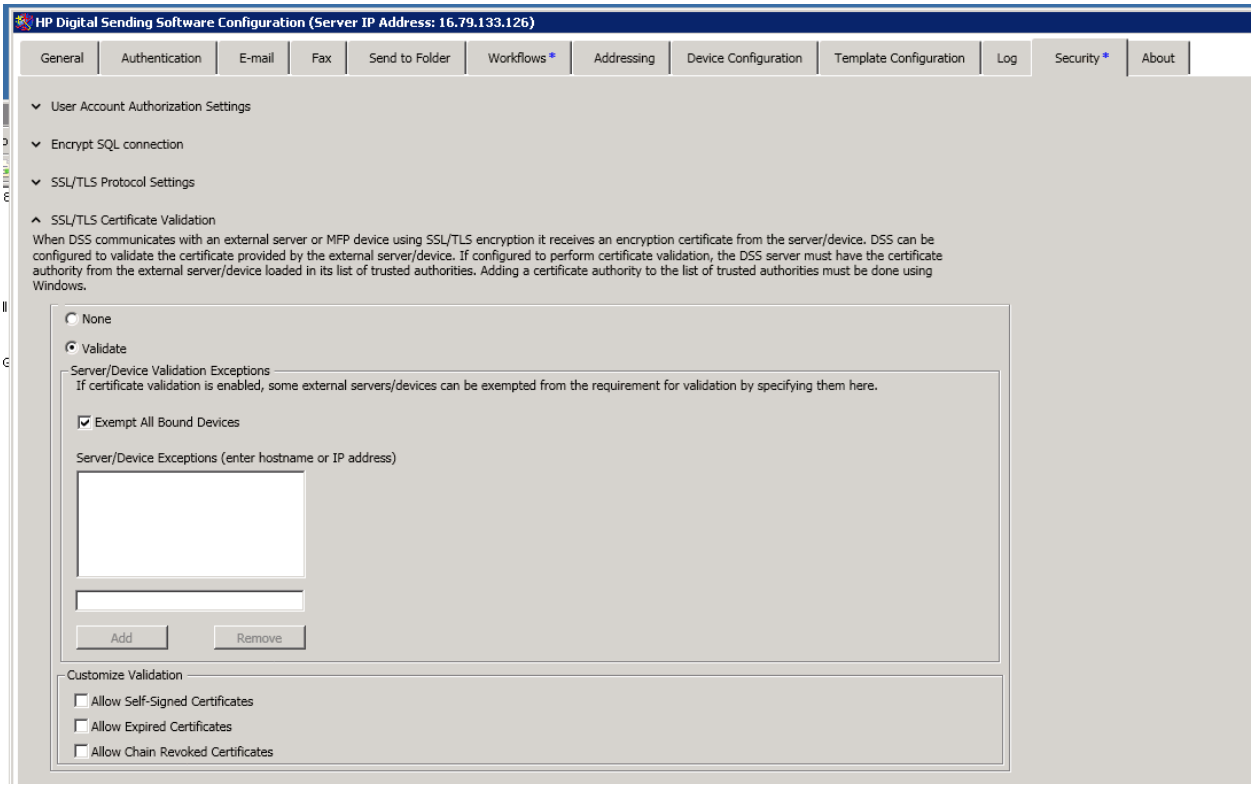
6. The SSL / TLS session continues because Entity 1 trusts the certificate it got from Entity 2.

Note that the above session only succeeds because entity 1 did some work **before** it started the communication with entity 2. When server certificate validation is on clients will only be able to communicate with servers when the server certificate is signed by a CA in the client's trusted root certification authorities store. Therefore, putting a CA certificate into the trusted root certification authorities store is a big deal and forms the backbone of trust that allows secure communication.

Certificates can be imported in Windows using the same mmc plug-in we used before to view the DSS certificates. Highlight the store into which you want to import a certificate then under the **Action** menu choose **All Tasks** and then **Import…** to start the import process.

Where you obtain a CA certificate to import into the Trusted Root Certification Authorities store depends on the CA. Some large public CAs have these publically available. If you need to obtain the CA certificate from a smaller entity you will have to find and follow the CA's documentation for obtaining their certificate.

In DSS server certificate validation is off by default. The UI to enable and configure server certificate validation is found on the Configuration Utility's Security tab.



When server certificate validation is enabled for DSS as a whole, it can be disabled for specific servers or devices by adding them to the Server / Device exception list.

When a Remote Configuration Utility (RCU) is in use it must be considered a server when server certificate validation is enabled. Even though most of the time the RCU will be the client in communications with the DSS service it is managing, there will be occasional communications from the DSS service back to the RCU in which the service is the client in the communication and the RCU is the server. Therefore, if an RCU is used when server certificate validation is enabled the DSS certificates from the RCU computer must be loaded on to the computer running the DSS service, or, the computer which runs the RCU must be added to the Server / Device Exceptions list shown in the user interface shown above.

Note that server certificate validation when communicating with devices can be exempted for all devices as a group. This is because configuring devices properly for server certificate validation with DSS can be a very time consuming thing which involves new certificates being created for, and loaded onto, each device and the CA which creates those certificates needs to be loaded on the DSS server. Only extremely security conscious customers will want to do this for their fleet of devices so a checkbox is provided to easily exempt all the devices at once instead of having to add them to the exempt list one at a time. More detail on how to properly configure a device and DSS for server certificate validation is discussed later in this document.

The UI shown above does not control server certificate validation for three communication channels:

- For SSL / TLS communication between the DSS service and the DSS database server certificate validation is always on. Communication with the database is discussed in more detail later in this paper

- For SSL / TLS communication between the DSS service and LDAP servers certificate validation is always on. Communication with LDAP servers is discussed in more detail later in this paper
- For SSL / TLS communication between the DSS Configuration Utility and the DSS service server certificate validation is controlled by a configuration file. This is discussed in more detail later in this document.

SSL / TLS communication with FutureSmart Devices

SSL / TLS encryption is used when DSS communicates with FutureSmart devices. By default server certificate validation is disabled for this communication channel.

FutureSmart devices have default, self-signed, certificates on them that can be used for SSL /TLS communication when server certificate validation is not enabled. However, these default certificates are not adequate to use when server certificate validation is enabled. When server certificate validation is enabled new certificates must be generated for each device and loaded on the device. Also, the certificate for the CA that created the device certificates will have to be loaded onto the DSS server in its Trusted Certificate Store in order for the process to work.

Please refer to the whitepaper "HP Jetdirect and SSL/TLS" for instructions on how to generate certificates for FutureSmart devices and load them onto the device. Instructions to access this whitepaper were given earlier in this document in the section *Brief Overview of SSL / TLS communication.*

When generating certificates for the device the names in the certificate are very important. Recall that certificates can have a primary name, the CN, and optionally may have additional names in the Alternative Subject Name section of the certificate. The following name requirements exist for the certificates to work properly with DSS:

- One of the names must be the IP address of the device. For this reason, if server certificate validation is enabled, devices must have an IP address that does not change.
- If a device is added to DSS by host name or fully qualified domain name (fqdn) then that name must appear in the certificate **exactly** as it was entered into DSS.

Server certificate validation for DSS <-> FutureSmart device communication can be enabled via the UI on the Configuration Utility's Security tab. Please see the Server Certificate Validation section of this paper for details.

SSL / TLS communication between the DSS CU and the DSS Service

All communication between the Configuration Utility and the DSS service uses SSL / TLS protocols. This is true when the CU and service are on the same server or on different servers.

Server certificate validation is off by default for DSS CU <-> DSS service communications. Server certificate validation for this communication channel is not controlled by the UI on the Configuration Utility's Security tab. Server certificate validation for this communication can be enabled / disabled in the configuration file:

<install-folder>\Hewlett-Packard\HP Digital Sending Software\Filesystems\Product\Dss\Configuration\HP.Dss.App.ConfigurationUtility.View.config.xml.

In this config file, under the <appSettings> section, is the item <add key="AcceptAllSSLCertificates" value="true" />. To enable the server certificate validation change 'true' to 'false', save the file and restart the DSS CU. This must be done on each server that runs the Configuration Utility, including the DSS server itself and any servers that run the CU remotely.

When server certificate validation is turned on for DSS CU <-> DSS service communication, and the CU is running on a different server than the service, then the DSS CA certificate from the server running the DSS service must be put into the Trusted Root Certification Authorities store on the server running the CU.

SSL / TLS communication with the SQL Server database

The SQL database used by DSS does not contain passwords that have been entered by the DSS administrator, but it may contain sensitive information such as email addresses and folder destinations for which the administrator may want to implement added security.

By default DSS does not use the SSL / TLS protocols when interacting with the SQL database. In order to enable SSL / TLS communication for this channel several things must be done:

1- Create and install a certificate for the SQL Server instance and configure SQL Server to know about and use the certificate
2- Put the SQL Server certificate's CA Authority certificate on the server that is running the DSS service (only necessary when using an external database).
3- Enable SSL / TLS communication for this channel in DSS

Before going deeper into the instructions there are two things it is important to understand.

- Server certificate validation is always on for communication with the SQL Server database
- FutureSmart devices directly access the DSS database for addressing information. Since the devices at this time are not enabled to use SSL / TLS communication with the database the SQL Server instance **must NOT be configured to require** SSL / TLS communication for all clients. Instead, the database will be configured to use SSL / TLS if the connecting client requests it and then DSS is configured to request an SSL /TLS communication.

Below are detailed instructions on configuring SQL server and DSS to use SSL / TLS security. Most of the instructions given below come from a Microsoft white paper that can be found at: http://support.microsoft.com/kb/316898 entitled *How to enable SSL encryption for an instance of SQL Server by using Microsoft Management Console.* They have been slightly edited to help tailor them to the specific situation where DSS is the client and SQL Server is the server. If the reader wants more information browsing to the MS whitepaper and following its available hyperlinks may be helpful.

Step 1 - Install a certificate on the SQL Server computer with Microsoft Management Console (MMC)

To use SSL encryption, you must install a certificate on the SQL Server computer. Follow these steps to install the certificate by using the Microsoft Management Console (MMC) snap-in.

1. How to configure the MMC Snap-in
    a. To open the MMC console, click **Start**, and then click **Run**. In the **Run** dialog box type:

       MMC
    b. On the **Console** menu, click **Add/Remove Snap-in...**.
    c. Click **Add**, and then click **Certificates**. Click **Add** again.
    d. You are prompted to open the snap-in for the current user account, the service account, or for the computer account. Select the **Computer Account**.
    e. Select **Local computer**, and then click **Finish**.
    f. Click **Close** in the **Add Standalone Snap-in** dialog box.
    g. Click **OK** in the **Add/Remove Snap-in** dialog box. Your installed certificates are located in the **Certificates** folder in the **Personal** container.
2. Use the MMC snap-in to install the certificate on the server:
. Click to select the **Personal** folder in the left-hand pane.
    a. Right-click in the right-hand pane, point to **All Tasks**, and then click **Request New Certificate...**.
    b. The **Certificate Request Wizard** dialog box opens. Click **Next**. Select **Certificate type is "computer"**.
    c. In the **Friendly Name** text box you can type a friendly name for the certificate or leave the text box blank, and then complete the wizard. After the wizard finishes, you will see the certificate in the folder with the fully qualified computer domain name.

Step 2 - Configure the DSS Server to trust the certificate being used by the SQL Server database. (This only needs to be done if the DSS Server and the server running SQL Server are different computers. This can only occur when DSS is configured to use an external database instead of its own default database.)

For the client to request the SSL encryption, the client computer must trust the server certificate and the certificate must already exist on the server. You have to use the MMC snap-in to export the Trusted Root Certification Authority used by the server certificate:

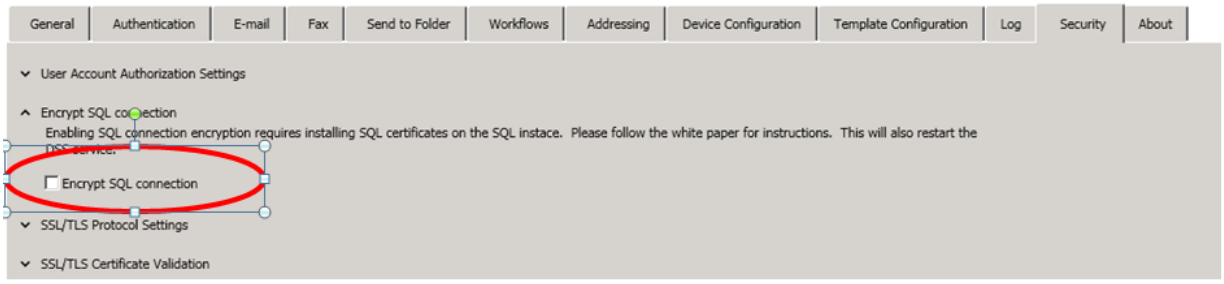
1. To export the server certificate's Trusted Root Certificate Authority (CA), follow these steps:
    a. Open MMC, and then locate your certificate in the **Personal** folder on the server running SQL Server.
    b. Right-click the certificate name, and then click **Open**.
    c. Review the **Certification Path** tab. Note the top most item.
    d. Navigate to the **Trusted Root Certification Authorities** folder, and then locate the Certificate Authority noted in step c..
    e. Right-click **CA**, point to **All Tasks**, and then click **Export**.
    f. Select all the defaults, and then save the exported file to your disk where the client computer can access the file.

2. Follow these steps to import the certificate on the DSS server:
. Navigate to the DSS Server computer by using the MMC snap-in, and then browse to the **Trusted Root Certification Authorities** folder.
    a. Right-click the **Trusted Root Certification Authorities** folder, point to **All Tasks**, and then click **Import**.
    b. Browse, and then select the certificate (.cer file) that you generated in step 1. Select the defaults to complete the remaining part of the wizard.

Step 3 - Configure DSS to use SSL / TLS when communicating with SQL SERVER

This step is done on the Security tab of the DSS Configuration Utility. Checking the checkbox shown in the screenshot below will enable the SSL / TLS communication. Remember, for this communication channel server certificate validation is always on.

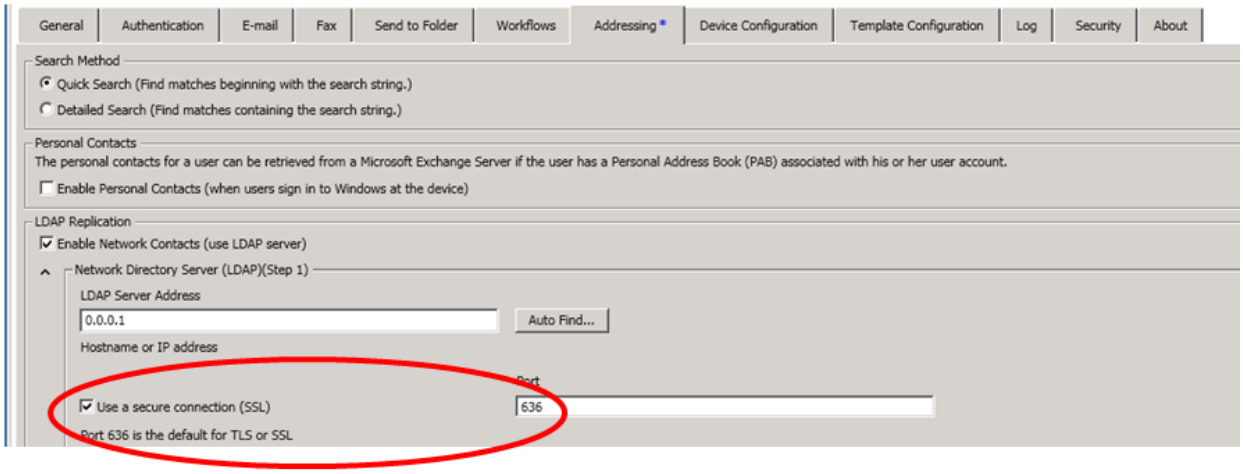
When the **Encrypt SQL connection** checkbox is checked DSS will:
1. Test the connection
2. If the test from step 1 fails the checkbox will be unchecked
3. If the test from step 1 passes, when the Apply button is pressed to save current settings, DSS will restart the CU and service



SSL/TLS communications with LDAP servers

DSS may communicate with LDAP servers for several reasons. These include authentication if LDAP authentication is the chosen method, and for addressing if the system is configured to do LDAP address replication.

In the DSS Configuration Utility, when a feature is enabled that uses an LDAP server, the use of the SSL / TLS protocols is enabled via a checkbox in the UI. Below is a screenshot of the UI to configure DSS for LDAP address replication.

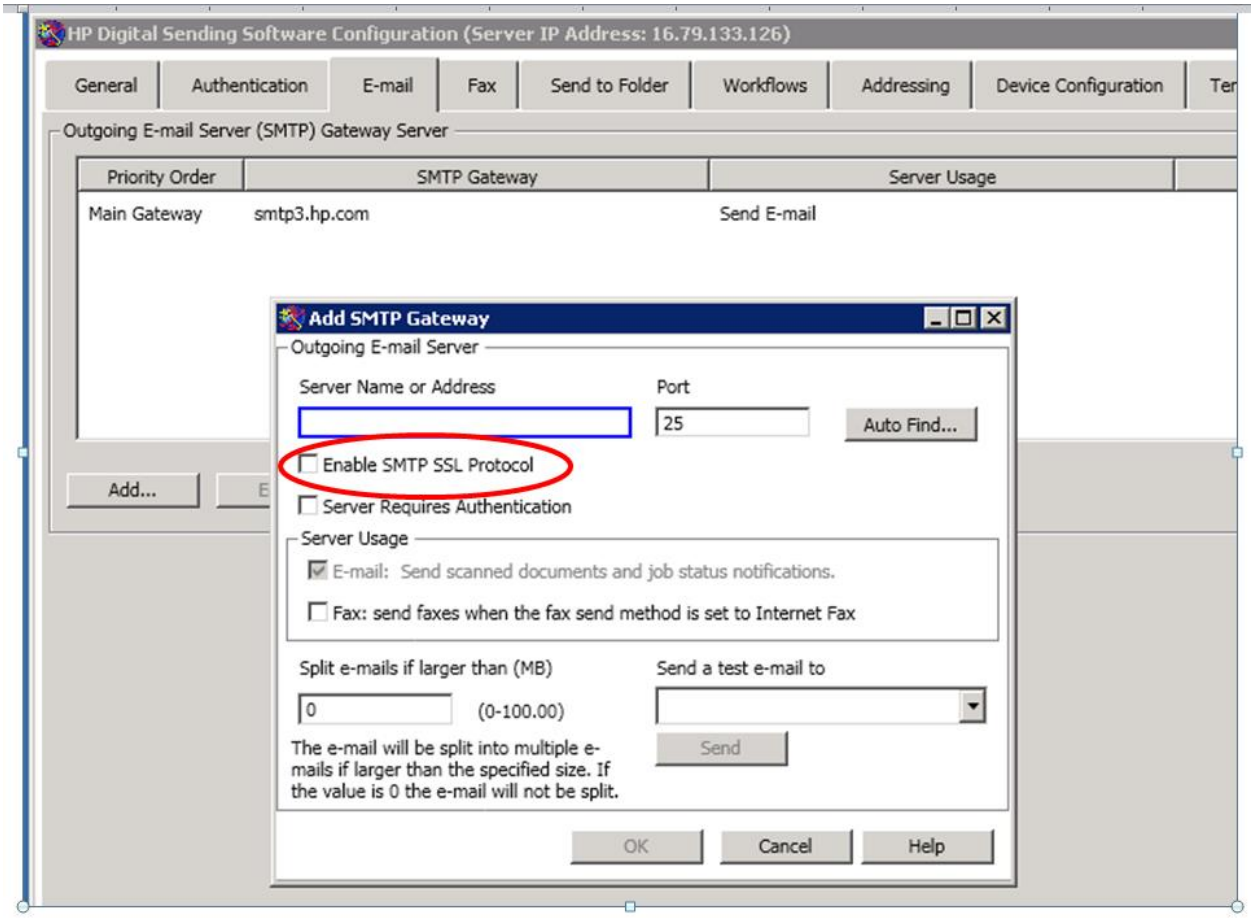


The checkbox shown above for the LDAP addressing UI is also available in the LDAP authentication UI (not pictured here).

When SSL / TLS communication with LDAP servers is enabled server certificate validation is always enabled. The UI on the security tab for exempting servers from server certificate validation does not apply to LDAP servers.

SSL / TLS communication with SMTP servers

SSL / TLS communication with SMTP servers is enabled in the UI, on the Configuration Utility's E-mail tab, when adding an SMTP server to the list of available servers for use. The SSL / TLS communication is enabled on a per server basis. The screenshot below shows the UI and the checkbox to check to enable SSL / TLS communication with an SMTP server.



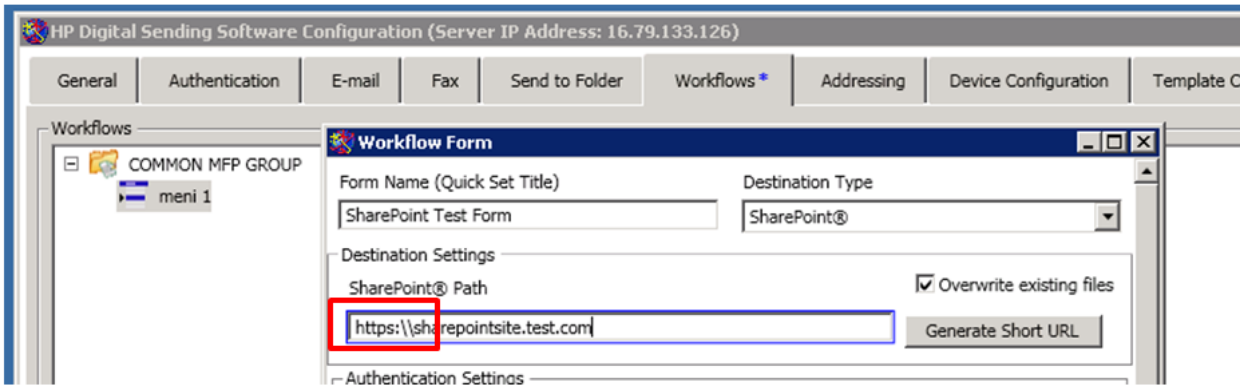
Server certificate validation for communication with SMTP servers is enabled in the CU UI on the Security tab shown earlier in this paper. If the DSS administrator wants server certificate validation on in general but wants to exclude communication with an SMTP server then the SMTP server should be added to the Server / Device exceptions list box.

There are other security mechanisms available for e-mail jobs that are sent from FutureSmart devices. These are e-mail encryption and e-mail signing. These security mechanisms work on the data of the e-mail message, while SSL / TLS works by encrypting the entire communication channel between the DSS server and the SMTP server. These three mechanisms can be used individually or in any combination with one another. More information on e-mail signing and encryption is given later in this document.

SSL / TLS communication with SharePoint sites

DSS 5's Send to Workflow functionality includes Microsoft SharePoint sites as a possible destination. When configuring a send to workflow form with a SharePoint destination SSL / TLS communication is enabled when the URL provided for the site starts with https. If the URL starts with http then SSL / TLS communication is not used

The screenshot below shows a workflow form being configured and a secure URL being provided for the SharePoint destination.



Server certificate validation for communication with SharePoint servers is enabled in the Configuration Utility UI on the Security tab shown earlier in this paper. If the DSS administrator wants server certificate validation on in general but wants to exclude communication with a SharePoint server then the SharePoint server should be added to the Server / Device exceptions list box.

Device Access of Address Book Information in the DSS Database

FutureSmart devices directly access the DSS database for addressing information. When a FutureSmart device is bound to DSS it is given the SQL connection string needed to directly access the DSS database. This connection string is passed to the device using the encryption scheme described earlier in this paper in the section “SSL/TLS Encryption with FutureSmart Devices”.

Security for the data exchange between FutureSmart devices and the DSS database is provided by the username and password that are part of the connection string. Once the connection is made the data is transferred without being encrypted. This data can include names, physical addresses, fax numbers, phone numbers and email addresses.

Pre-FutureSmart devices do not exchange data directly with the DSS database. Pre-FutureSmart devices request address information from the DSS service which in turn collects the information from its database and returns it to the device. This information exchanged is encrypted using the scheme described earlier in this paper in the section “Data encryption with pre-FutureSmart Devices”.

E-mail Signing and Encryption

In addition to SLL / TLS protocol security provided by DSS for communication to SMTP servers, there are two other security mechanisms available for e-mail jobs when the e-mail is sent from a FutureSmart device via DSS. These are e-mail encryption and e-mail signing. Email encryption and signing operate on the data of the e-mail message, while SSL / TLS protocols encrypt the entire communication channel between DSS and SMTP servers. These three security mechanisms can be used individually or in any combination with each other.

Pre-FutureSmart devices offer e-mail signing and encryption in their firmware, but these functions are not available when pre-FutureSmart devices send e-mail jobs via DSS. If the administrator wants to use e-mail signing and encryption from pre-FutureSmart devices that are managed by DSS the devices must be configured to send e-mail jobs directly from the device instead of via DSS.

E-mail encryption involves encrypting the email with a public key for each recipient. This means that the public key certificates for each recipient must have been made available before the e-mail is sent. Certificates used for encrypted e-mail are S/MIME type certificates. The FutureSmart device is responsible for obtaining the public key certificates of the recipients and passing them to DSS. For this to happen the device must already have been configured for LDAP based addressing and the public key certificates of the recipients must have been stored in the LDAP directory for each recipient. The LDAP attribute that holds the public key certificates is configured in the device when configuring email encryption on the device. Only recipients that exist in the LDAP server and have their public key certificates available in the server can receive encrypted emails. The e-mail is then encrypted with the recipient's public key and the recipient must decrypt the email with their private key when it is received. DSS encrypts e-mail using the AES 256 encryption algorithm.

The e-mail client will also need to be configured correctly to receive and read encrypted e-mails. Please see your e-mail system's documentation for specific instructions.

To summarize the steps that must be taken before an encrypted email can be sent from a FutureSmart device via DSS:

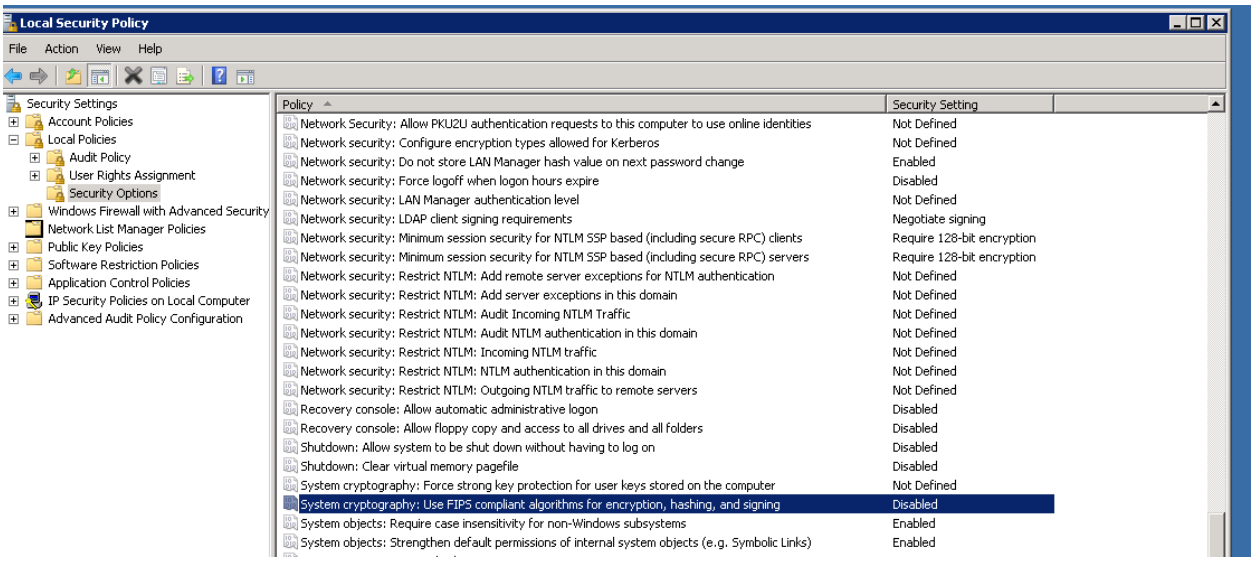
1- Load the public key certificate for all potential recipients into the LDAP directory for the recipient
2- Configure the FutureSmart device for LDAP based addressing
3- Configure the FutureSmart device for encrypted email, which includes providing the name of then LDAP attribute which stores the recipients' public key certificates
4- Configure recipient e-mail clients as necessary to read encrypted e-mail

E-mail signing involves DSS signing the email with DSS's private key. When DSS sends the signed e-mail it also sends its public key certificate so the recipient can decrypt the signature. But in order to trust the DSS public key certificate it receives, each recipient must have already loaded the DSS CA's certificate into its Trusted Root Certification Authorities store. This means that the DSS CA certificate must have been exported from the DSS server and made available to all recipients of signed emails for loading into their Trusted Root Certification Authorities store before e-mail signing is used. DSS uses the SHA 256 hashing algorithm for e-mail signing.

FIPS Security Policy in Windows

Starting with DSS version 5.02.01 DSS can run with the security policy for FIPS enabled. The policy is shown in the screenshot below. For all versions before 5.02.01 the security policy must be disabled.

If DSS 5.02 is being installed over the top of a previous version of DSS 5.0 the FIPS security policy must remain disabled during the install. Once the installation is complete the policy may be enabled.



PDF Encryption when using DSS OCR

DSS provides an OCR engine which is capable of producing many output file formats. Included in these formats are Searchable PDF and Searchable PDF/A. When the output file type is one of these PDF types, PDF encryption may be applied. When DSS OCR creates the files the PDF encryption used is PDF 1.7 Extension Level 3 for AES encryption using 256-bit keys. Keys are generated from user-entered passwords.

When the system is configured to produce non-searchable PDF or PDF/A output files the encryption is done on the device itself before the file is sent to DSS. For this scenario please see the documentation for the device being used to determine the PDF encryption algorithm.

On some devices, such as MFP Flow devices, there is an OCR engine built into the device firmware itself. The internal OCR engine is used when the device is configured to send job directly from the device, the DSS OCR engine is used when the device is configured to send jobs via DSS. If the internal OCR engine is used then the device firmware applies PDF encryption. For this scenario please see the documentation for the device being used to determine the PDF encryption algorithm.