Secure Email Server (SES)

Admin Guide

ASCERTIA LTD

July 2017

Document Version- 5.5.0.1

CONTENTS

FIGURES

TABLES

1 Introduction

1.1 Scope

This manual describes how to install and configure the Ascertia Secure Email Server (SES) software.

1.2 Intended Readership

This manual is intended for Secure Email Server Administrators responsible for its installation and configuration. It is assumed that the reader has a basic knowledge of standard email protocols, PKI and IT security.

1.3 Conventions

The following typographical conventions are used in this guide to help locate and identify information:

- **Bold text** identifies menu names, menu options, items you can click on the screen, file names, folder names, and keyboard keys.
- `Courier` font identifies code and text that appears on the command line.
- **`Bold courier`** identifies commands that you are required to type in.

1.4 Technical Support

If Technical Support is required, Ascertia has a dedicated support team providing debugging assistance, integration assistance and general customer support. Ascertia Support can be accessed in the following ways:

Support Website	www.ascertia.com/support
Support Email	support@ascertia.com
Knowledge base	http://kb.ascertia.com/display/AKBS/Ascertia+Knowledge+base

In addition to the free support service describe above, Ascertia provides formal support agreements with all product sales. Please contact sales@ascertia.com for more details.

A Product Support Questionnaire should be completed to provide Ascertia Support with further information about your system environment. When requesting help it is always important to confirm:

- System Platform details;
- SES and ADSS Server version numbers and build date;
- Details of the specific issue and the relevant steps taken to reproduce it;
- ADSS Server Database version and patch level;
- The product log files.

2 Concepts & Architecture

2.1 Introduction

Today email is one of the most popular applications on the Internet. This position has been earned because of its clear business benefits. However for several years, there has been a dramatic rise in email fraud and phishing, so much so that the "From:" field in an email message or the message content itself can no longer be trusted as a face value. It is easy to spoof these details, opening the way for attackers to create and use false identities with minimum effort. The recent record level fraud committed at a leading European bank, which at the core level involved email spoofing, serves as clear evidence of what happens when emails are trusted blindly without any security trails.

To overcome such threats in electronic communication, Ascertia provides the multi-purpose Advanced Digital Signature Services (ADSS) Server as a solution for digitally signing and verifying any type of electronic document so that it's content can be easily trusted and the identity of its authors, reviewers and/or approvers can be automatically verified.

2.2 Feature & Benefits

Ascertia offers the Secure Email Server as a client application to the ADSS Server with the objective of managing the:

- Signing of outgoing emails (using standard S/MIME digital signatures which are verifiable by most email clients like Microsoft Outlook, Lotus Notes and Thunderbird).
- Signing of outgoing email attachments (using PDF digital signatures, XML Digital signatures or PKCS#7/CMS signatures. Advanced long-term signature profiles using CAdES and XAdES as well as the PDF equivalent are also supported).
- Verification of incoming signed emails (including sender's identity check using real-time OCSP)
- Verification of incoming signed email attachments (including sender's identity check using real-time OCSP).

2.2.1 Key Management

One of the benefits of the Secure Email Sever and ADSS Server solution is that no cryptographic keys need to be deployed to end-users. End-users in fact don't even need to learn how to sign their emails as the signatures are automatically applied. The signing keys are managed by the ADSS Server in either a secure HSM or held in encrypted form in the ADSS Server database. The choice of how many signing keys to use is optional, e.g.:

- Set-up an organization, departmental, or role level key in the ADSS Server, and sign all emails using this group key. E.g. signing of emails by the Accounts Department, or signing of corporate material using a corporate signing key.
- Set-up unique user keys for each email user. Then the ADSS Server applies the signature using the signing key of the email sender (using the information from the email's "FROM" field to identify the user).

2.2.2 Security

The solution has been designed for a high-level of practical security. In particular, the secure ADSS Server is used for managing all cryptographic keys and signing/verification operations and other auxiliary trust services. Being an e-Trust service provider, the ADSS Server has been designed with security in mind:

- All service request/responses to the ADSS Server are logged in secure cryptographically-protected logs.
- Operator console access to the ADSS Server is protected using HTTPS with client authentication.

- ADSS server comes with optional dual control facility so that at least two operators are required to make configuration changes and then approve these changes before settings can be changed.
- Client applications like the Secure Email Server need to be registered on ADSS Server, plus the service request messages can optionally be signed by the Secure Email Server.
- The ADSS Server can use any PKCS#11 tamper-resistant HSM for cryptographic processing. Safenet (Luna and Eracom devices) and nCipher HSMs have been tested, others can be supported on request.
- The ADSS Server operates on a secure internal network, e.g. in a secure computing room.
- High-availability is offered with the ability for the Secure Email Server to contact a fallback ADSS Server instance in case primary server becomes unavailable.

2.2.3 ADSS Signing Server

Ascertia ADSS Server is a multi-function application offering the following underlying e-Trust services:

- **Signing Service:** Ability to digital sign on the server using server-held keys. It supports most popular signature formats including S/MIME, PDF, XML DSig, PKCS#7, CMS, CAdES-T, CAdES-X-Long, XAdES-T and XAdES-X-Long. The signing service is available through an XML/SOAP web service interface and watched folders.
- **Verification Service:** Ability to fully verify the above signature formats, including the complete validation of associated certificate chains. The verification service is available through an XML/SOAP web service interface. Historical signature verification using archived information is also supported.
- **Key Manager & Certification Service:** Ability to generate cryptographic keys and certify the public keys via an internal CA or external online or offline CAs. Supports communication with HSMs. This service is also available through an XML/SOAP web services interface.
- **Timestamping Service:** Provides cryptographic timestamping service. Required to form long-term signatures with embedding trusted timestamp information.
- **OCSP Validation Authority Service:** Provides real-time certificate status information, which can again be embedded in advanced signature formats so that such signatures can be verified in the long term.

ADSS Server can support multiple CAs and multiple PKI trust models, allowing a single centralized trust service to be established. For further details see the ADSS Server Admin Manual.

2.3 Architecture

Ascertia Secure Email Server is drop-in MTA server which supports both SMTP and POP3 protocols. It is built using the open source Apache James, a popular platform-independent pure java mail server. Apache James provides a mail application platform with an embedded standard extension mechanism built around:

- **Matchers:** Message selection code, i.e. specific code written to filter emails which are then passed to the mailet.
- **Mailets:** Message processing code, i.e. specific code written which then processes the filter email, e.g. in case of SES the mailet calls ADSS Server to digitally sign or verify the email (or email attachment). Therefore the Apache James server can be seen as a container for the matcher and mailet.

The high-level architecture of Secure Email Server is shown below. The green boxes show standard Apache James modules including the service engines SMTP and POP3 which deliver email to Apache James. The core mail processing engine shown with the orange boxes represents Ascertia specific matcher and mailet for digital signature creation and verification functionality:

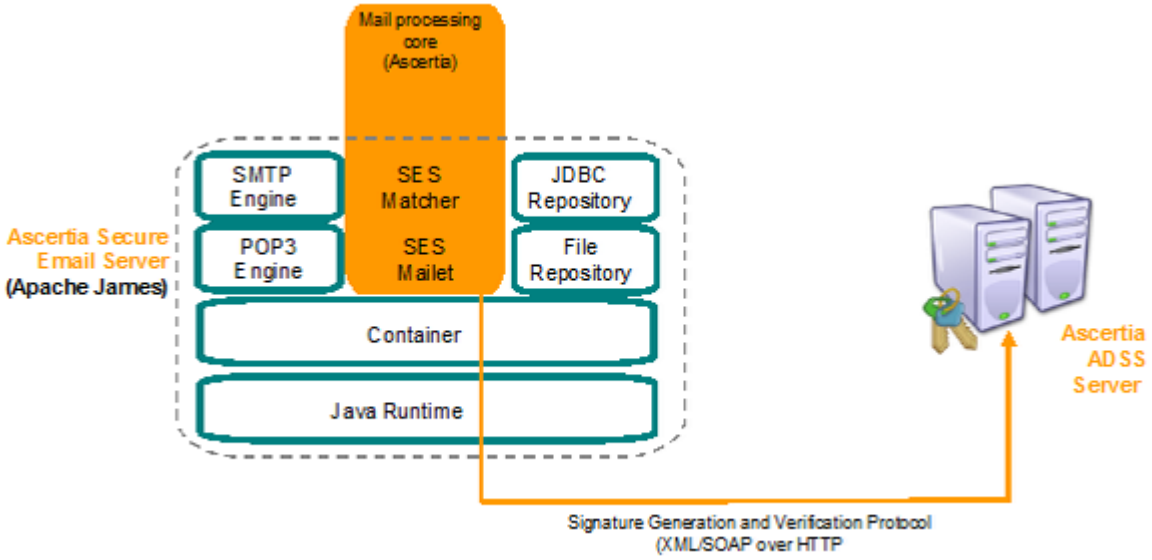

Figure 1 - SES Architecture

In future Ascertia plans to provide extended functionality which allows email archiving and email encryption/decryption services through this matcher/mailet architecture. For further details on these services, please contact info@ascertia.com

2.4 Deployment Scenarios

A typical deployment of Secure Email Server is shown below, with an internal email server handling emails as normal, but then passing these to the Secure Email Server for signing. If the SES Matcher determines that the email requires signing the SES Mailet then sends a signing request message to the ADSS Sever (operating on secure network behind an internal firewall).

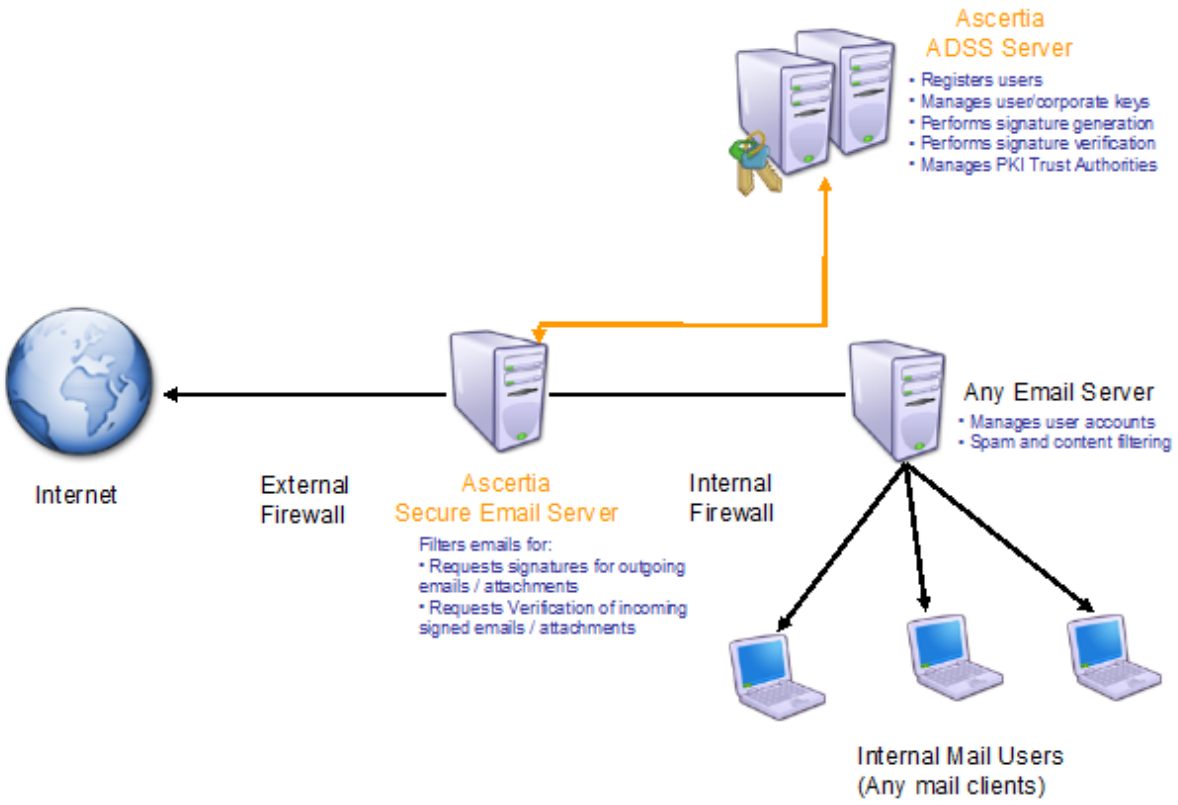

Figure 2 - Deployment Scenario

Similarly for incoming emails, the SES Matcher can filter emails and attachments which are signed and then the SES Mailet can make a request to the ADSS Server to verify these signatures. The original email and the verification results are then passed to the internal email server for later delivery to the internal mail recipient(s) or administrators in case of errors in verification.

2.4.1 Outgoing Email & Attachment Signing

The following diagram illustrates a typical scenario where outgoing emails are automatically signed before being released from the company using Secure Email Server:

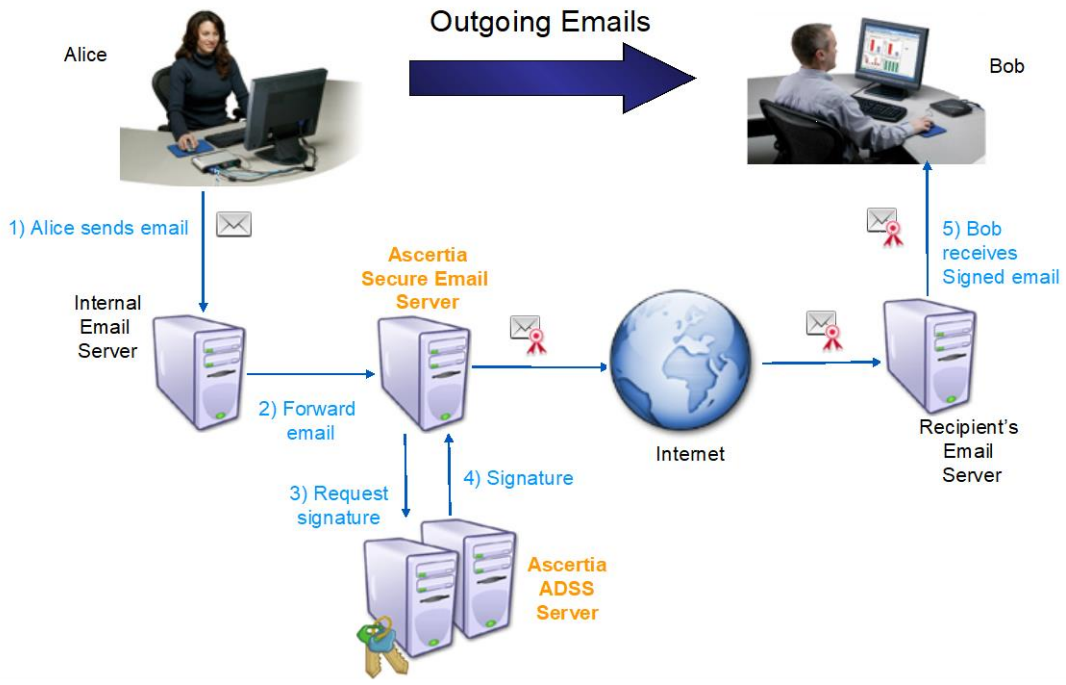

Figure 3 - Outgoing Email & Attachment Signing

1. Here the client e.g. Alice sends an email. Her email is sent to her corporate internal email server e.g. her existing email server so that no re-configuration of email clients is necessary.
2. The internal email server then forwards the emails to the Ascertia Secure Email Server. The SES Matcher processes the email to determine if it requires a signature. Typical matcher might filter emails based on:
    a. Keywords in the From, To, CC, BCC, Subjects and/or email body
    b. Whether the email contains a particular type of attachment (e.g. PDF, XML, DOC etc)
    c. Whether the email is signed
3. If the email matches a particular filter, the Secure Email Server will then make a request to the ADSS Server to perform the signature. Note a number of different operations could be asked from the Secure Email Server, the following options are allowed in the configuration file as explained later:
    a. SIGN_ATTACHMENT
    b. SIGN_EMAIL
    c. SIGN_ATTACHMENT, SIGN_EMAIL (signs both the email and the attachment)

Archiving encrypting and decrypting functionality will be added in a future release of the Secure Email Server.

4. ADSS Server on receiving the request processes it and returns the signature back to the Secure Email Server, which embeds the signature into the email and forwards this to the recipients Email Server. The recipient receives the signed email or signed document.
5. When signing a PDF email attachment (e.g. invoices) it is possible to configure the Secure Email Server Mailet to specify how it wishes the signature appearance to look and where to stamp it

on the PDF document. Example signature appearance is shown below:

6. Normally the Mailet will not want to provide this information in every signing request message to the ADSS Server, so signature profiles can be pre-configured on the ADSS Server and referred to by the Mailet in the request message. Note the language for the above labels is also configurable; see the ADSS Manual for further details.

2.4.2 Verification of Incoming/Outgoing Signed Emails/Attachments

Similar to the above scenario it is possible for incoming emails which are signed, or where the email attachment is signed, it is possible for the Secure Email Server to automatically verify the signatures and then provide these results to the internal recipient or administrator as depicted below:

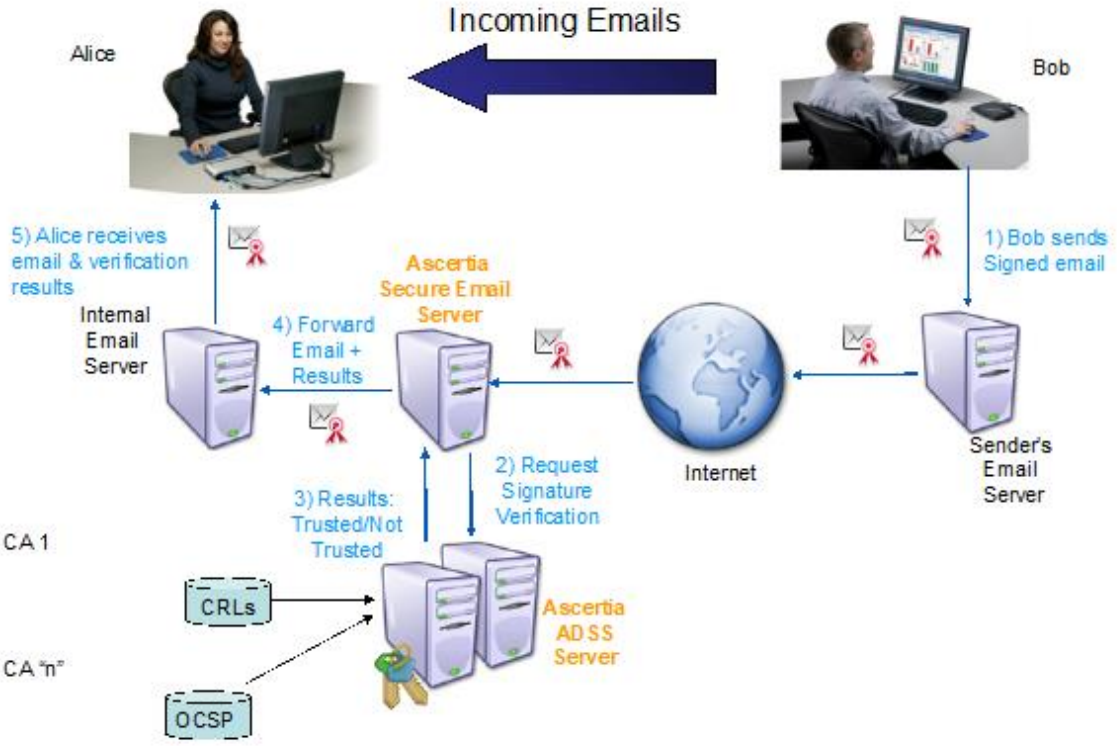

Figure 4 - Verification of Incoming/Outgoing Signed Emails/Attachments

1. An external user “Bob” sends a signed email or signed attachment to “Alice” an internal user.
2. When the email enters the corporate network it is routed to the Secure Email Server, where the Matcher filters emails to detect if they contain signatures or if the attachments are signed. The filter as explained above can also be set-up to look for specific keywords in various email fields. If the

email or attachment is signed then the Mailet processes it by making a signature verification request to the ADSS Server.

3. The ADSS Server is responsible for verifying digital signatures. It gathers certificate status information from the configured Certificate Authorities (CAs) using either offline CRLs or online OCSP mechanism. The ADSS Server provides a response back to the Mailet on the signature verification results which can be that the signature is “trusted”, “not trusted” or “indeterminate” (i.e. the status of the signature cannot be determined because of insufficient information). The Mailet on receiving the response from the ADSS Server can perform various actions depending on the configuration, e.g.:
    a. If results show signature is trusted, then pass the signed email to the original recipient with the results shown in the email body (or as an attachment)
    b. If results are not trusted, then pass the signed email to a system administrator.
4. The Secure Email Server then passes the email to the internal corporate email server.
5. This internal email server then delivers the email to the internal user.

3 System Requirements

The following table summarizes the minimum requirements for installing Secure Email Server (SES):

Component	Minimum Requirements
Operating System	Windows Server 2016, 2012 R2, 2012
CPU/RAM	A modern fast CPU with 4 GB RAM (6 GB or more is recommended if larger documents are being processed) and 200MB disk space.

Table 1 - System Requirements

4 Installation

4.1 Installation Instructions

First extract the Secure Email Server Installation zip to any directory where you want Secure Email Server to be installed. The extracted directory will contain following folders:

- bin
- conf
- docs
- james-2.3.1
- jre
- logs
- setup
- ssl
- temp

Run **install.bat** located within the **[SES Installation Directory]/setup** folder. Launching install.bat will install SES Service. The Secure Email Server is installed as a Windows Service (Windows Control Panel/Administrative Tools/Services) under the name **Ascertia-SES**.

To run SES, launch the windows services panel and select the Ascertia-SES Service and click start button, to stop it click on stop button.

*Ascertia-SES service can also be run by clicking on **run.bat** file located at: **[SES Installation Directory]/james-2.3.1/bin**.*

The Secure Email Server setup is not like other installations which create desktop icons, shortcuts, asks for target folder where the product is to be installed, etc. Instead it is provided as a zipped set of files and running setup.bat merely performs the basic configuration before it can be used.

After installation you will still need to perform further configuration as described in the next section.

*The SES service only reads the **ses.xml** configuration file when it starts. SES must therefore be restarted whenever a change is made to the ses.xml configuration file.*

4.2 Uninstall Instructions

To uninstall SES Service, Run **uninstall.bat** located within the **[SES Installation Directory]/setup** folder.

5 Apache James Configurations

This section describes Apache James configurations (e.g. add user, configure Domain etc.) to configure mail server. This section only describes the information relevant to using Secure Email Server for signing and verifying emails and not general operation on how to set-up other email services (e.g. POP3 accounts). For this we recommend detailed information available from http://james.apache.org/index.html

Most of the default configurations are already made in the setup, if you like to change the default configurations, please follow the steps mentioned in this section.

5.1 DNS Configuration

DNS Transport services are controlled by **config.xml** configuration file located inside **[SES Installation Directory]/james-2.3.1/apps/james/SAR-INF** folder. This configuration file affects SMTP remote delivery.

Domain Name System (DNS) is the name resolution protocol for TCP/IP networks, such as the Internet. Client computers query a DNS server to resolve memorable, alphanumeric DNS names to the IP addresses that computers use to communicate with each other. The **dnsserver** tag defines the boundaries of this configuration block. It encloses the entire relevant configuration for the DNS server. The behaviour of the DNS service is controlled by the attributes and children of this tag.

The standard children of the dnsserver tag are:

Servers: This is a list of DNS Servers to be used by James and are specified by one, or more server elements, which are child elements. Each server element is the IP address of a single DNS server.

```
<dnsserver>
      <servers>
            <server>127.0.0.1</server>
            <server>166.181.19.205</server>
      </servers>
</dnsserver>
```

5.2 POP3 Server

The Post Office Protocol version 3 (POP3) is an application-layer Internet standard protocol, to retrieve e-mail from a remote server over a TCP/IP connection. The POP3 service is controlled by a configuration block in the **config.xml**. The **pop3server** tag defines the boundaries of this configuration block. It encloses the entire relevant configuration for the POP3 server. The behaviour of the POP service is controlled by the attributes and children of this tag.

This tag has an optional Boolean attribute **enabled** that defines whether the service is active or not. The value defaults to **"true"** if not present.

```
<pop3server enabled="true">
      <port>110</port>

      <!--Uncomment this if you want to bind to a specific inetaddress-->
      <bind>192.168.0.181</bind>

      <!--Uncomment this if you want to use TLS (SSL) on this port-->
      <useTLS>true</useTLS>
</pop3server>
```

5.3 SMTP Server

SMTP (Simple Mail Transfer Protocol) is a TCP/IP protocol used in sending and receiving e-mail. However, since it is limited in its ability to queue messages at the receiving end, it is usually used with one of two other protocols, POP3 or IMAP, that let the user save messages in a server mailbox and download them periodically from the server. In other words, users typically use a program that uses SMTP for sending e-mail and either POP3 or IMAP for receiving e-mail. The SMTP service is controlled by a configuration block in the config.xml. The **smtpserver** tag defines the boundaries of this configuration block. It encloses the entire relevant configuration for the SMTP server. The behavior of the SMTP service is controlled by the attributes and children of this tag.

This tag has an optional Boolean attribute **enabled** that defines whether the service is active or not. The value defaults to **"true"** if not present.

```
<smtpserver enabled="true">
      <!--port 25 is the well-known/IANA registered port for SMTP-->
      <port>25</port>

      <!--Uncomment this if you want to bind to a specific ip address
      otherwise it will listen all interfaces installed on the system-->
      <bind>192.168.0.181</bind>

      <!--Uncomment this if you want to use TLS (SSL) on this port-->
      <useTLS>true</useTLS>
</smtpserver>
```

5.4 Authenticated SMTP (SMTP Auth)

Authenticated SMTP is a method of securing your SMTP server. With SMTP AUTH enabled senders who wish to relay mail through the SMTP server (that is, send mail that is eventually to be delivered to another SMTP server) must authenticate themselves to James before sending their message. Mail that is to be delivered locally does not require authentication. This method ensures that spammers cannot use your SMTP server to send unauthorized mail, while still enabling users who may not have fixed IP addresses to send their messages.

Configuring James for Authentication SMTP is a multi-step process. It requires several adjustments of the **config.xml**. To enable **SMTP AUTH**, follow these steps:

1. As mentioned above, **SMTP AUTH** requires that James be able to distinguish between mail intended for local delivery and mail intended for remote delivery. James makes this determination by matching the domain to which the mail was sent against the **<servernames>** element of the James configuration block. Any local domains should be explicitly listed as **<servername>** elements in this section

   ```
   <James>
         <servernames autodetect="true" autodetectIP="true">
               <servername>localhost</servername>
               <servername>ascertia-ses.com</servername>
         </servernames>
   </James>
   ```

2. James is configured out of the box so as to not serve as an open relay for spammers. This is done by restricting the IP addresses from which mail will be accepted using the mailet **RemoteAddrNotInNetwork**. This restriction must be lifted before users can send from arbitrary clients. To do this, comment out or remove the mailet tag containing the class attribute **RemoteAddrNotInNetwork**. This tag can be found in the **spoolmanager** configuration block, in the root processor configuration. Comment out this tag.

```
<!--If you are using SMTP authentication then you should disable
this matcher/mailet pair-->
<mailet match="RemoteAddrNotInNetwork=127.0.0.1"
class="ToProcessor">
```

3. Set the **authRequired** element of the **smtpserver** configuration block to **"true"**.

```
<!--Uncomment this if you want to require SMTP authentication-->
<authRequired>true</authRequired>
```

4. If you wish to ensure that authenticated users can only send email from their own account, you may optionally set the **verifyIdentity** element of the **smtpserver** configuration block to **"true"**.

```
<!--Uncomment this if you want to verify sender addresses, ensuring
that the sender address matches the user who has authenticated-->
<verifyIdentity>true</verifyIdentity>
```

5.5 Server Wide Configurations

Postmaster

The body of this element is the address that the server will consider its postmaster address. This address will be listed as the sender address of all error messages that originate from Apache James.

Servernames

This element determines exactly which mail domains and IP addresses the server will treat as local.

Usernames

This element has no body, but instead has three required Boolean attributes. These are **ignoreCase**, **enabledAliases**, and **enableForwarding**. The first of these determines whether email user names will be treated as case-insensitive or not. The second attribute configures whether local user aliasing will be enabled. Finally, the value of the third attribute determines whether forwarding to potentially remote users will be enabled.

```
<James>
      <postmaster>postmaster@ascertia-ses.com</postmaster>
      <servernames autodetect="true" autodetectIP="true">
            <servername>localhost</servername>
            <servername>ascertia-ses.com</servername>
      </servernames>
      <usernames ignoreCase="true" enableAliases="true"
      enableForwarding="true"/>
</James>
```

5.6 Remote Manager

In James, user accounts are created through the **RemoteManager**. So, after installation is complete, the first step to adding users is to configure the **RemoteManager**.

```
<remotemanager enabled="true">
      <port>4555</port>
      <!--Uncomment this if you want to bind to a specific inetaddress-->
      <bind>192.168.0.181</bind>
</remotemanager>
```

5.7 Setting up Administrator Account

You also need to setup administrator accounts so that administrator can login and create users. This is done in the same **RemoteManager** tag.

```
<remotemanager enabled="true">
      <handler>
            <administrator_accounts>
                  <!--Change the default login/password-->
                  <account login="root" password="root"/>
            </administrator_accounts>
      </handler>
</remotemanager>
```

After you've done this, restart James to ensure that any changes you've made in the configuration are incorporated into the running system. You are now ready to create user accounts, follow these instructions:

1. Open Command Prompt and telnet to the host and port on which the **RemoteManager** is listening. For command-line telnet clients this is generally done by typing **"telnet <host> <pass>"** where <host> is the James hostname and <port> is the **RemoteManager** port specified in the James config.xml e.g.:

   ```
   telnet localhost 4555
   ```

2. You will be prompted for your administrator **userid** and **password**. Enter the values you specified in the James **config.xml**.
3. After logging in, type **"adduser <user> <password>"** where <user> is the user name and <password> is the password of the account you wish to create.

   ```
   Adduser user1 password123
   ```

*The user name should NOT be a complete email address. Rather, all email addresses of the form **<user>@<domain>** (where <domain> is any of the values specified in the <servernames> block) will be delivered to this account by default. Mailet configuration can change this default behavior.*

Repeat step 3 for all user accounts you wish to create. That's it. Your user accounts are now created and can be used by all James services.

5.8 Configuring SES Server to Relay Outgoing Emails

When SES Server is used as an internal email server or there is another internal email server which relays all emails to the SES Server for Email/Attachment signing, if the outgoing email is for an external email server then below settings are required in SES Server so that it can relay the emails to external email server:

1. Go to location: **[SES Server Installation Directory] /james-2.3.1/apps/james/SAR-INF**
2. Edit the file **config.xml**
3. Search for the parameter **"<mailet match="RecipientIsLocal" class="LocalDelivery"/>"** and add these settings right after this parameter:

```
<!--Required when the outgoing email is intended for an external
domain-->
<mailet match="SenderHostIs=DOMAIN_NAME" class="RemoteDelivery">
      <outgoing>file://var/mail/outgoing/</outgoing>

      <!--Delivery Schedule based upon RFC 2821, 4.5.4.1-->
      <!--5 day retry period, with 4 attempts in the first hour,
      two more within the first 6 hours, and then every 6 hours
      for the rest of the period-->
      <delayTime>5 minutes</delayTime>
      <delayTime>10 minutes</delayTime>
      <delayTime>45 minutes</delayTime>
      <delayTime>2 hours</delayTime>
      <delayTime>3 hours</delayTime>
      <delayTime>6 hours</delayTime>
      <maxRetries>25</maxRetries>

      <!--The number of threads that should be trying to deliver
      Outgoing messages-->
      <deliveryThreads>1</deliveryThreads>

      <!--If false the message will not be sent to given server
      if any recipients fail-->
      <sendpartial>false</sendpartial>

      <!--remote email server settings to which the email should be
      relayed -->
      <gateway>SMTP_SERVER_TO_RELAY_OUTGOING_EMAILS</gateway>
      <gatewayPort>25</gatewayPort>
      <gatewayusername>USER @ REMOTE_SERVER</gatewayusername>
      <gatewayPassword>USER_PASSWORD</gatewayPassword>
</mailet>
```

4. Save the changes and restart the SES Server.

In this case the SES server acts like a proxy server. Email relay must be turned on in the SMTP server to relay the outgoing emails to SES server.

A Single instance of SES server cannot be configured to relay emails to two different email servers at the same time i.e. for incoming and outgoing emails. The solution to this issue is:

1. *Deploy two SES servers one for relaying all incoming email and another for relaying all outgoing email.*
2. *Use SES server as internal mail server.*

5.9 Configuring SES Server to Relay Incoming Emails

When SES Server is used as an internal email server or there is another internal email server to whom SES Server relay all the incoming emails after Email/Attachment verification, provided that the incoming email is from an external email server then below settings are required in SES Server so that it can relay the emails to internal email server:

1. Go to location: **[SES Server Installation Directory] /james-2.3.1/apps/james/SAR-INF**
2. Edit the file **config.xml**
3. Search for the parameter **<mailet match="RecipientIsLocal" class="LocalDelivery"/>** and add these settings right after this parameter:

```
<!--Required when the outgoing email is intended for an external
domain-->
<mailet match="HostIsLocal" class="RemoteDelivery">
      <outgoing>file://var/mail/outgoing/</outgoing>

      <!--Delivery Schedule based upon RFC 2821, 4.5.4.1-->
      <!--5 day retry period, with 4 attempts in the first hour,
      two more within the first 6 hours, and then every 6 hours
      for the rest of the period-->
      <delayTime>5 minutes</delayTime>
      <delayTime>10 minutes</delayTime>
      <delayTime>45 minutes</delayTime>
      <delayTime>2 hours</delayTime>
      <delayTime>3 hours</delayTime>
      <delayTime>6 hours</delayTime>
      <maxRetries>25</maxRetries>

      <!--The number of threads that should be trying to deliver
      Outgoing messages-->
      <deliveryThreads>1</deliveryThreads>

      <!--If false the message will not be sent to given server
      if any recipients fail-->
      <sendpartial>false</sendpartial>

      <!--remote email server settings to which the email should
      Be relayed-->
      <gateway>SMTP_SERVER_TO_RELAY_INCOMING_EMAILS</gateway>
      <gatewayPort>25</gatewayPort>
      <gatewayusername>USER @ REMOTE_SERVER</gatewayusername>
      <gatewayPassword>USER_PASSWORD</gatewayPassword>
</mailet>
```

4. Search the parameter **"<authorizedAddresses>127.0.0.0/8</authorizedAddresses>"** and add the IP address of your SMTP Server so that apache james can relay the emails to it e.g.:

```
<authorizedAddresses>192.168.0.150/8</authorizedAddresses>
```

5. Save the changes and restart the SES Server.

A Single instance of SES server cannot be configured to relay emails to two different email servers at the same time i.e. for incoming and outgoing emails. The solution to this issue is:

1. *Deploy two SES servers one for relaying all incoming email and another for relaying all outgoing email*
2. *Use SES server as internal mail server.*

6 Secure Email Server Configurations

This section describes the Secure Email Server specific configurations. All the configurations and workflows specific settings for incoming and outgoing emails are created using an XML based configuration file. This ses.xml file is located at **[SES Installation Directory]/conf** folder. The following sections explain each part of this file.

*The SES service only reads the **ses.xml** configuration file when it starts. SES must therefore be restarted whenever a change is made to the ses.xml file*

6.1 Sample Profiles

Following sample SES profiles are available in the SES package at location **[SES Installation Directory]/conf/samples/**:

1. **Outgoing Email Signing**

   For use when you wish to sign all out going emails. At the receiving end, the email will be verified and original email will be added as attachment. Also, verification response will be added as an xml attachment at receiving end.

   Use sample profile: Sign_Outgoing_Email.xml

2. **Outgoing Email Signing From Selected Senders**

   For use when you wish to sign all out going emails. Signing of the each email is done using each user's unique key referenced from the emails **"From"** field. At the receiving end, the email will be verified and original email will be added as attachment. Also, verification response will be added as an xml attachment at receiving end.

   Use sample profile: Sign_Outgoing_Email_From_Selected_Senders.xml

3. **Outgoing Email's PDF Attachment Signing**

   For use when you wish to sign PDF attachments (minimal basic settings), where an existing blank signature field needs to be signed through ADSS Signing service profile. Assuming there is a specific central signing key for this client on the ADSS Server. At the receiving end, the email will be verified and original email will be added as attachment. Also, verification response will be added as an xml attachment at receiving end.

   Use sample profile: Sign_Outgoing_Email_PDF_Attachment.xml

4. **Incoming Email Verification**

   For use when you wish to verify all the incoming emails. The email will be verified through ADSS Verification service profile and original email will be added as attachment. Also, verification response will be added as an xml attachment at receiving end.

   Use sample profile: Verify_Incoming_Emails.xml

5. **Incoming Email and Attachment Verification**

   For use when you wish to verify all the incoming emails along with attachments. The email and attachments will be verified through ADSS Verification service profile and original email will be added as attachment. Also, verification response will be added as an xml attachment at receiving end.

   Use sample profile: Verify_Incoming_Emails_And_Attachments.xml

6.2 Running the Sample Profiles

In order to run a sample profile, other than default PDF attachment signing, i.e. ses.xml, follow these steps:

- Stop SES service from Windows service or UNIX daemon.
- Go to location **[SES Installation Directory]/conf/** and rename the default **ses.xml** file to another name of your choice.
- Go to location **[SES Installation Directory]/conf/samples/** and rename the relevant sample profile to **ses.xml**, e.g. if it is required to run the **Sign_Outgoing_Emails_From_Selected_Senders.xml** then rename this file to **ses.xml** and place it in folder: **[SES Installation Directory]/conf/**.
- Start SES service from Windows service or UNIX daemon.

6.3 Profile Configurations

The following sections explain each part of ses.xml file. When reviewing these settings, it is important to note that some XML elements are required and some are optional. If an element is not required then you must (a) omit the tag, or (b) comment the tag.

*The SES service only reads the **ses.xml** configuration file when it starts. SES must therefore be restarted whenever a change is made to the ses.xml file*

6.3.1 General Settings

```
<SES>
      <Settings>
            <DomainName>ascertia-ses.com</DomainName>
            <AdminEmailAddress>admin@ascertia-ses.com</AdminEmailAddress>
            <OriginatorId>samples_test_client</OriginatorId>
            <EmailErrorAction>EMAIL_TO_ADMINISTRATOR</EmailErrorAction>
```

XML Tags	Description
SES	SES (Secure Email Server) is the root element.
Settings	The elements in between this tag are used for global configurations.
DomainName	Defines the name of your internal email domain.
AdminEmailAddress	Defines the email address of the Secure Email Server administrator.
OriginatorId	Defines the client ID for this SES Server and is included in the request messages which are sent to ADSS server. This client ID needs to be registered within the ADSS Client Manager module. Follow this link for more details: http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=client_manager
EmailErrorAction	Defines what to do with the email when an error occurs. It is useful in trapping such emails which failed to reach the recipient due to wrong filter configuration. Possible values are:

XML Tags	Description
	• **BLOCK_EMAIL** • **EMAIL_TO_ADMINISTRATOR** • **EMAIL_TO_RECIPIENTS** • **EMAIL_TO_ADMINISTRATOR_AND_RECIPIENTS** • **EMAIL_TO_SENDER** • **EMAIL_TO_ADMINISTRATOR_AND_SENDER**

Table 2 - General Settings

6.3.2 Signing Service Address Settings

```
<SigningServiceSettings>
      <SigningServiceAddress>
            <PrimaryAddress>http://192.168.0.201:8777/adss/signing/dss</Pri
            maryAddress>
            <SecondaryAddress>http://192.168.0.201:8777/adss/signing/dss</S
            econdaryAddress>
      </SigningServiceAddress>
      <NoFilterAction>EMAIL_TO_ADMINISTRATOR</NoFilterAction>
</SigningServiceSettings>
```

XML Tags	Description
SigningServiceSettings	The elements in between this tag are used for Signing Service configurations.
SigningServiceAddress	The elements in between this tag are used for defining ADSS Signing Service primary and secondary addresses.
PrimaryAddress	Defines the primary ADSS Signing Service address that will be used by default for all profiles.
SecondaryAddress	Defines the secondary ADSS Signing Service address that will be used by default for all profiles.
NoFilterAction	Defines what action to take if the mailet is failed to match configured filter criteria for signing process. Possible values are: • **BLOCK_EMAIL** • **EMAIL_TO_ADMINISTRATOR** • **EMAIL_TO_RECIPIENTS** • **EMAIL_TO_ADMINISTRATOR_AND_RECIPIENTS** • **EMAIL_TO_SENDER** • **EMAIL_TO_ADMINISTRATOR_AND_SENDER**

Table 3 - Signing Service Address Settings

6.3.3 Verification Service Address Settings

```
<VerificationServiceSettings>
      <VerificationServiceAddress>
            <PrimaryAddress>http://192.168.0.201:8777/adss/verification/dss
            </PrimaryAddress>
            <SecondaryAddress>http://192.168.0.201:8777/adss/verification/d
            ss</SecondaryAddress>
      </VerificationServiceAddress>
      <NoFilterAction>EMAIL_TO_ADMINISTRATOR</NoFilterAction>
      <ResponseNotTrustedAction>EMAIL_TO_ADMINISTRATOR</ResponseNotTrustedA
      ction>
</VerificationServiceSettings>
```

XML Tags	Description
VerificationServiceSettings	The elements in between this tag are used for Verification Service configurations.
VerificationServiceAddress	The elements in between this tag are used for defining ADSS Verification Service primary and secondary addresses.
PrimaryAddress	Defines the primary ADSS Verification Service address that will be used by default for all profiles.
SecondaryAddress	Defines the secondary ADSS Verification Service address that will be used by default for all profiles.
NoFilterAction	Defines what action to take if the mailet is failed to match configured filter criteria for verification process. Possible values are: • **BLOCK_EMAIL** • **EMAIL_TO_ADMINISTRATOR** • **EMAIL_TO_RECIPIENTS** • **EMAIL_TO_ADMINISTRATOR_AND_RECIPIENTS** • **EMAIL_TO_SENDER** • **EMAIL_TO_ADMINISTRATOR_AND_SENDER**
ResponseNotTrustedAction	Defines what action to take if verification response returned from ADSS Verification Service is not trusted. Possible values are: • **BLOCK_EMAIL** • **EMAIL_TO_ADMINISTRATOR** • **EMAIL_TO_RECIPIENTS** • **EMAIL_TO_ADMINISTRATOR_AND_RECIPIENTS** • **EMAIL_TO_SENDER** • **EMAIL_TO_ADMINISTRATOR_AND_SENDER**

Table 4 - Verification Service Address Settings

6.3.4 Proxy Settings

```
<ProxySettings status="disable">
      <ProxyHost>192.168.0.1</ProxyHost>
      <ProxyPort>8090</ProxyPort>
      <Credentials status="disable">
            <ProxyUserName>user1</ProxyUserName>
            <ProxyPassword>password1</ProxyPassword>
            <AuthenticationScheme>BASIC</AuthenticationScheme>
      </Credentials>
</ProxySettings>
```

XML Tags	Description
ProxySettings	The elements in between this tag are used for defining the proxy settings. **Note** it works only when status="enable".
ProxyHost	Defines the IP address or machine name of the proxy server.
ProxyPort	Defines the port number for communication with the proxy host.
Credentials	Defines the user credentials for proxy settings. **Note** it works only when status="enable".
ProxyUserName	Defines the user name for the proxy user authentication.
ProxyPassword	Defines the password for proxy user authentication.
AuthenticationScheme	Defines the authentication scheme. Possible values are: • **BASIC** • **DIGEST**

Table 5 - Proxy Settings

6.3.5 SSL/TLS Settings

```
<ClientAuthPfxSettings status="disable">
      <PfxFilePath>C:/Client.pfx</PfxFilePath>
      <PfxPassword>password12</PfxPassword>
</ClientAuthPfxSettings>
```

XML Tags	Description
ClientAuthPfxSettings	The elements in between this tag are used for defining the client authentication settings. **Note** it works only when status="enable".
PfxFilePath	Defines the path of the private key to be used for client SSL/TLS authentication when communicating with ADSS Server.
PfxPassword	Defines the password for the client SSL/TLS authentication private key.

Table 6 - SSL/TLS Settings

6.3.6 Sign Workflow Settings

```
<Outgoing>
     <Sign>
        <Workflow>
             <Filter>
                  <AND>
                     <To contains="bob@ascertia-ses.com"/>
                     <From contains="alice@ascertia-ses.com"/>
                     <Subject contains="sign"/>
                     <Body contains="secure"/>
                     <Email signed="false" encrypted="false"/>
                     <Attachment contains="pdf" signed="false"/>
                  </AND>
             </Filter>
             <OperationRule> SIGN_ATTACHMENT,SIGN_EMAIL</OperationRule>
             <ProfileSelection>
                  <SignAttachment profileId="adss:signing:profile:006">
                        <OverrideableAttributes>
                              <Attribute>
                                    <Name>SIGNING_LOCATION</Name>
                                    <Value>Guildford, Surrey</Value>
                              </Attribute>
                              <Attribute>
                                    <Name>SIGNING_REASON</Name>
                                    <Value>I certify this document</Value>
                              </Attribute>
                        </OverrideableAttributes>
                  </SignAttachment>
                  <SignEmail profileId="adss:signing:profile:007"/>
             </ProfileSelection>
             <CertificateSelection>
                  <SignAttachment alias="alice_document_signing"/>
                  <SignEmail alias="alice_email_signing"/>
            </CertificateSelection>
        </Workflow>
     </Sign>
</Outgoing>
```

XML Tags	Description
Outgoing / Incoming	The elements in between this tag are used to process outgoing or Incoming emails.
Sign / Verify	The elements in between this tag are used to process Sign or Verify operations. **Note** Either a single/multiple Sign or Verify element can appear under Outgoing/Incoming element.

XML Tags	Description
Workflow	The elements in between this tag are used to define the Sign or Verify workflow for outgoing emails.
Filter	The elements in between this tag are used to define outgoing email filters.
AND / OR	The elements under this tag will be used to filter the emails based on AND / OR operand. Secure Email Server does not support mix of AND / OR filtering, only one operand should be used for email filtering. Secure Email Server support different filter criteria, an email would be filtered even if a single criterion is met based on AND / OR operand. For AND / OR filter same elements are supported and elements can be omitted. Possible Filter criteria are: • **To** • **From** • **Subject** • **Body** • **Email** • **Attachment** **Note** User can specify multiple comma separated email domains in the "To" and "CC" filter criteria. E.g. <To contains="*@ascertia-ses.com,*@hotmail.com,*@gmail.com"/>
OperationRule	Defines the sequence of operations that the Secure Email Server Mailet will perform in one workflow. Possible values are: • **SIGN_ATTACHMENT** • **SIGN_EMAIL** • **SIGN_ATTACHMENT,SIGN_EMAIL**
ProfileSelection	The elements in between this tag are used to define the signing profile for email / attachment signing. Possible elements under this tag are: • **SignAttachment:** Defines the Signing Profile Name/ID to be included in the request messages sent to ADSS Server for attachment signing. This signing profile is defined within ADSS Signing Service module. ○ **OverrideableAttributes:** Optionally this tag can also defines the PDF signature appearance elements which are being overridden from the default values defined in the in ADSS Server Signing Profile. Possible values are: ▪ **SIGNING_REASON** This attribute allows you to configure the signing reason to be included in the signature. ▪ **SIGNING_LOCATION** This attribute allows you to configure the signing location to be included in the signature.

XML Tags	Description
	▪ **CONTACT_INFO** This attribute allows you to configure the signing Contact info to be included in the signature. ▪ **SIGNING_AREA** This attribute allows you to select a default location on the PDF document where the signature appearance will be stamped. Possible Values are: • 1 *(For Top Left)* • 2 *(For Top Right)* • 3 *(For Center)* • 4 *(For Bottom Left)* • 5 *(For Bottom Right)* ▪ **SIGNING_PAGE** This attribute defines which page of the document to sign on when using SIGNING_AREA. ▪ **SIGNING_FIELD** This attribute defines the empty signature field which already present in the PDF document. Use either Signing_AREA or SIGNING_FIELD ▪ **VISIBILITY** This attribute defines whether a visible or invisible signature will be created on the PDF document. Possible values are: • TRUE *(For Visible Signatures)* • FALSE *(For Invisible Signatures)* • **SignEmail:** Defines the Signing Profile Name/ID to be included in the request messages sent to ADSS Server for email signing. This signing profile is defined within ADSS Signing Service module.

CertificateSelection	The elements in between this tag are used to define the signing certificate alias to be included in the request messages sent to ADSS Server for attachment/email signing. Possible elements under this tag are:
	• **SignAttachment**
	• **SignEmail**
	
	Both of the above elements can have these possible values:
	• **Any Certificate Alias**
	If it is required to override the default signing certificate configured in the ADSS Server Signing Profile then configure your required certificate alias here e.g.
	`<SignAttachment alias="alice_signing_Key"/>`
	`<SignEmail alias="alice@ascertia.com"/>`
	
	This certificate alias should be generated either inside Key Manager or ADSS Server Certification Service. e.g.

XML Tags	Description

	• **FROM_EMAIL_ADDRESS**
	If it is required to use the sender's email address as a certificate alias for SignAttachment/SignEmail then configure this value e.g.
	`<SignAttachment alias="FROM_EMAIL_ADDRESS "/>`
	`<SignEmail alias="FROM_EMAIL_ADDRESS "/>`
	
	User email address must be used as certificate alias while generating signing certificate inside Key Manager or ADSS Server Certification Service.
	
	• **NONE**
	If it is required to use the default signing key associated with ADSS Server Signing Profile then configure this value e.g.
	`<SignAttachment alias="NONE"/>`
	`<SignEmail alias="NONE"/>`

Table 7 - Sign Workflow Settings

6.3.7 Verify Workflow Settings

```
<Incoming>
     <Verify>
        <Workflow>
             <Filter>
                  <AND>
                     <To contains="bob@ascertia-ses.com"/>
                     <From contains="alice@ascertia-ses.com"/>
                     <Subject contains="sign"/>
                     <Body contains="secure"/>
                     <Email signed="false" encrypted="false"/>
                     <Attachment contains="pdf" signed="false"/>
                  </AND>
             </Filter>
             <OperationRule>VERIFY_ATTACHMENT</OperationRule>
             <VerificationResultRule attachResponseXml="true">
             ADD_RESULT_TO_BOTTOM</VerificationResultRule>
        </Workflow>
     </Verify>
</Incoming>
```

XML Tags	Description
Incoming / Outgoing	The elements in between this tag are used to process Incoming or Outgoing emails.
Verify / Sign	The elements in between this tag are used to process Verify or Sign operations.

XML Tags	Description
	Note Either a Verify or Sign element can appear under Incoming or Outgoing element.
Workflow	The elements in between this tag are used to define the Verify workflow for Incoming emails.
Filter	The elements in between this tag are used to define Incoming email filters.
AND / OR	The elements under this tag will be used to filter the emails based on AND / OR operand. Secure Email Server does not support mix of AND / OR filtering, only one operand should be used for email filtering. Secure Email Server support different filter criteria, an email would be filtered even if a single criterion is met based on AND / OR operand. For AND / OR filter same elements are supported and elements can be omitted. Possible Filter criteria are: • **To** • **From** • **Subject** • **Body** • **Email** • **Attachment** **Note** User can specify multiple comma separated email domains in the "To" and "CC" filter criteria. E.g. <To contains="*@ascertia-ses.com,*@hotmail.com,*@gmail.com"/>
OperationRule	Defines the sequence of operations that the Secure Email Server Mailet will perform in one workflow. Possible values are: • **VERIFY_ATTACHMENT** • **VERIFY_EMAIL** • **VERIFY_ATTACHMENT, VERIFY_EMAIL**
VerificationResultRule	Defines how to handle the verification results. Possible values are: • **ADD_RESULT_TO_TOP** *(For Verify Attachment)* • **ADD_RESULT_TO_BOTTOM** *(For Verify Attachment)* • **ADD_ORIGINAL_EMAIL_AS_ATTACHMENT** *(For Verify Email)* • **SEND_ORIGINAL_EMAIL** **Note** If the end recipients of the email need to verify the signature of the original sender, then the email should be forwarded by SES without any changes (to ensure the original sender's signature is not corrupted). Therefore ensure that **SEND_ORIGINAL_EMAIL** option is selected.

Table 8 - Verify Workflow Settings

It is very important to note that restarting of the Ascertia-SES service is essential whenever any change is made in SES.xml configuration file.

The filter criteria for two workflows must not be same. If a filter criterion is met the Secure Email Server does not check any further filters and start processing the email based on the first matched workflow.

7 Running SES Over SSL/TLS Using Client Authentication

SES supports communication with ADSS Server over SSL/TLS using client SSL/TLS authentication. This ensures SES instance is uniquely identified within ADSS Server, and the instance must authenticate itself to ADSS Server before using services. The configuration requires set-up on the target ADSS Server and once complete, adding the necessary configuration to ses.xml file. Follow this KB article for configurations on both SES and ADSS Server:

http://kb.ascertia.com/pages/viewpage.action?pageId=1671344#ConfiguringSSLAuthentication-StepsforSSLClientAuthenticationcommunicationwithADSSServices

Once configured ensure the **ClientAuthPfxSettings** is enabled and correctly configured. These settings are found under the SSL/TLS Settings section.

SES service must be restarted once the configuration changes have been made.

8 Upgrading SES

Follow these instructions to upgrade any older version of ADSS SES to the latest one:

1. Stop and uninstall the old SES (see the section uninstall Instructions).
2. Download and extract the latest package of SES. Note do not overwrite the current SES file set with the latest version.
3. Copy the **ses.xml** file from location: **[Old SES Installation Directory]/conf** and overwrite it at location: **[New SES Installation Directory]/conf.**
4. Install the new SES (see the section Installation Instructions).
5. If the ADSS Server is secured by TLS (standard in production systems) then it is required to copy the **jssecacerts** file from location: **[Old SES Installation Directory]/jre/lib/security** and overwrite it at the corresponding new location: **[New SES Installation Directory]/jre/lib/security**.
6. Start the SES from the Windows NT Services Panel / UNIX Daemon. Note the new service will point at the new directory structure, and ultimately, new **ses.sh** or **ses.bat** file.

*** End of document ***