**TLS with SMTP (without S/MIME)**

1. **Sending Email**
   - Client (generates message)

2. **SMTP**
   - Data exchanged are encrypted here

3. **Originator SMTP Server**
   - Server sends certificate & (optionally) requests client cert
   - (optionally) client sends its cert
   - Client generates encryption key & encrypts using server's public key
   - All data sent through TLS channel are encrypted & integrity protected – until one end terminates the secure connection

4. **Destination SMTP Server**

5. **Receiving Email**
   - Client (opens and reads message)

**NOTE:** TLS authenticates the two ends of a connection, over which all data are encrypted between the two points. As illustrated here, the TLS channel is from the originating organization’s SMTP server to the destination’s SMTP server. Since Stage 1 meaningful use is organization-to-organization, that should be acceptable. It’s possible to configure clients to use TLS to connect to the SMTP server as well. If this were done by both Originator and Destination organizations, we would have 3 TLS channels: 1) from sending email client to originator SMTP server, 2) from originator SMTP server to destination SMTP server; and 3) from Destination SMTP server to receiving email client.

**SMTP with S/MIME**

1. **Sending Email**
   - Client (encrypts and integrity protects email message using addressee’s public key; digitally signs using own private key; and sends message)

2. **S/MIME SMTP**
   - Message body is encrypted and integrity protected; header is in the clear

3. **Originator SMTP Server**
   - Routes message to addressee

4. **S/MIME SMTP**
   - Message body is encrypted and integrity protected; header is in the clear

5. **Originator SMTP Server Client**
   - Routes message to addressee

6. **S/MIME SMTP**
   - Message body is encrypted and integrity protected; header is in the clear

7. **Destination SMTP Server Client**
   - Routes message to addressee

8. **S/MIME SMTP**
   - Message body is encrypted and integrity protected; header is in the clear

9. **Receiving Email**
   - Client (verifies identity of originator using sender’s public key; decrypts message using own private key, and confirms integrity)

**NOTE:** S/MIME digitally signs the message and encrypts the message body using the recipient’s public key. For stage 1 meaningful use, when the sender and recipient are “organizations” instead of individuals, this makes it quite similar to TLS with SMTP, except that the S/MIME message header (to/from, subject) are sent in the clear.
What I would like to see: S/MIME SMTP over TLS