In the name of Allah

External Communication Tests

1. **FCS_TLSC_EXT.1.1**:

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher- level protocol, e.g.,, as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
- Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.
- Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server- selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.
- Test 4: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.
- Test 5: The evaluator shall perform the following modifications to the traffic:

i. Test 5.1: Change the TLS version selected by the server in the Server Hello to a non- supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.

ii. Test 5.2: Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE cipher suite) or that the server denies the client's Finished handshake message.

iii. Test 5.3: Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.

iv. Test 5.4: Modify the signature block in the Server's Key Exchange handshake message, and verify that the client rejects the connection after receiving the Server Key Exchange message.

v. Test 5.5: Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.

vi. Test 5.6: Send a garbled message from the Server after the Server has issued the ChangeCipherSpec message and verify that the client denies the connection.

2. **FCS_TLSC_EXT.1.2:**

The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g., Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the TOE. The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

- Test 1: The evaluator shall present a server certificate that does not contain an identifier in either the Subject Alternative Name (SAN) or Common Name (CN) that matches the reference identifier. The evaluator shall verify that the connection fails.
- Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the

SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type.

- Test 3: The evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.
- Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds.
- Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier:
    i. Test 5.1: The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g., foo.*.example.com) and verify that the connection fails.
    ii. Test 5.2: The evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g., *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g., foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left- most label as in the certificate (e.g., example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g., bar.foo.example.com) and verify that the connection fails.
    iii. Test 5.3: The evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g., *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g., foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g., bar.foo.com) and verify that the connection fails.
- Test 6: [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails.

- Test 7: [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails.

3. **FCS_TLSC_EXT.1.3:**

   The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following additional test:

   - Test 1: The evaluator shall demonstrate that a peer using a certificate without a valid certification path results in an authenticate failure. Using the administrative guidance, the evaluator shall then load the trusted CA certificate(s) needed to validate the peer's certificate, and demonstrate that the connection succeeds. The evaluator then shall delete one of the CA certificates, and show that the connection fails.

4. **FCS_TLSC_EXT.4.1:**

   The evaluator shall verify that TSS describes the supported Elliptic Curves Extension and whether the required behavior is performed by default or may be configured. If the TSS indicates that the supported Elliptic Curves Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the supported Elliptic Curves Extension. The evaluator shall also perform the following tests:

   - Test 1: The evaluator shall configure the server to perform an ECDHE key exchange message in the TLS connection using a non-supported ECDHE curve (for example, P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.

5. **FCS_TLSS_EXT.1.1:**

   The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS. The evaluator shall also perform the following tests:

   - Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern

the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

- Test 2: The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.
- Test 3: The evaluator shall use a client to send a key exchange message in the TLS connection that does not match the server-selected cipher suite (for example, send an ECDHE key exchange while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA key exchange while using one of the ECDSA cipher suites.) The evaluator shall verify that the application disconnects after receiving the key exchange message.
- Test 4: The evaluator shall perform the following modifications to the traffic:
    i. Test 4.1: Change the TLS version selected by the server in the Server Hello to a nonsupport TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.
    ii. Test 4.2: Modify at least one byte in the client's nonce in the Client Hello handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.
    iii. Test 4.3: Modify the signature block in the Client's Key Exchange handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.
    iv. Test 4.4: Modify a byte in the Clint Finished handshake message, and verify that the server rejects the connection and does not send any application data.
    v. Test 4.5: After generating a fatal alert by sending a Finished message from the client before the client send a ChangeCipherSpec message, send a Client Hello with the session identifier from the previous test, and verify that the server denies the connection.
    vi. Test 4.6: Send a garbled message from the client after the client has issued the ChangeCipherSpec message and verify that the Server denies the connection.

6. **FCS_TLSS_EXT.1.2:**

The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS

versions, and any configuration necessary to meet the requirement must be contained in the AGD guidance.

- Test 1: The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0, TLS 1.0, TLS 1.1, and TLS 1.2 if it was selected.

7. **FCS_TLSS_EXT.1.3:**

The evaluator shall verify that the TSS describes the key agreement parameters of the server key exchange message. The evaluator shall verify that any configuration guidance necessary to meet the requirement must be contained in the AGD guidance.

- Test 1: The evaluator shall attempt a connection using an ECDHE cipher suite and a configured curve and, using a packet analyzer, verify that the key agreement parameters in the Key Exchange message are the ones configured. (Determining that the size matches the expected size for the configured curve is sufficient.) The evaluator shall repeat this test for each supported NIST Elliptic Curve and each supported Diffie-Hellman key size.

8. **FIA_X509_EXT_1.1:**

The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

- Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails.
- Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

- Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL, OCSP, or OCSP Stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method:
    i. The evaluator shall test revocation of the node certificate.
    ii. The evaluator shall also test revocation of an intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA), if intermediate CA certificates are supported.

  The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.
- Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.
- Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)
- Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
- Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

9. **FIA_X509_EXT_1.2:**

The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the selfsigned Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

- Test 1: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.

- Test 2: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the CA flag in the basicConstraints extension not set. The validation of the certificate path fails.
- Test 3: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the CA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.

10. FIA_X509_EXT_2.2:

The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates. The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed. The evaluator shall perform the following test for each trusted channel:

- Test 1: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.
- Test 2: The evaluator shall demonstrate that an invalid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity cannot be accepted.