Enrollment over Secure Transport

Technical field

The invention concerns an improved Enrollment over Secure Transport (EST) procedure.

Background

Enrollment over Secure Transport (EST) as defined in RFC7030 by the Internet Engineering Task Force (IETF), describes the use of Transport Layer Security (TLS) and Hypertext Transfer Protocol (HTTP) to provide an authenticated and authorized channel for Simple Public Key Infrastructure (PKI) Requests and Responses. Architecturally, the EST service is located between a Certification Authority (CA) and a client.

RFC9148 by the IETF defines a new transport for EST based on the Constrained Application Protocol (CoAP), since constrained devices and networks for the use-case of Internet-of-Things (IoT) devices use CoAP (UDP) & CoAPs (DTLS) instead of HTTP (TCP) & HTTPS (TLS). Datagram Transport Layer Security (DTLS) is used to secure CoAP messages.

Summary

According to a first aspect there is provided an apparatus comprising an Enrollment over Secure Transport (EST) client, the EST client being configured to:

store a trust anchor;

initiate a Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) handshake with an EST server;

receive a TLS/DTLS EST server certificate from the EST server;

perform validation of the TLS/DTLS EST server certificate using the trust anchor; and

when validation fails, complete the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session with the EST server.

The EST client can be further configured to:

send a certificate authority (CA) certificate retrieval request to the EST server using the non-trusted TLS/DTLS session; and

receive one or more CA certificates from the EST server.

The EST client can be further configured to store the CA certificate(s) in temporary or volatile memory.

The EST client can be further configured to send a validation request to the EST server for validating the CA certificate(s), wherein the validation request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receive a validation response from the EST server, wherein the validation response comprises a CA certificate response signed by the EST server and an EST signing certificate signed by the vendor CA; and

validate the CA certificate(s).

The EST client can be further configured to, after validating the CA certificate(s), move the CA certificate(s) to a trust anchor store. For example, the EST client can update an explicit TA store/database with the CA certificate(s).

The EST client can be configured to:

send a CA certificate retrieval request to the EST server using the non-trusted TLS/DTLS session, wherein the CA certificate retrieval request comprises a vendor identifier associated with a vendor CA associated with the trust anchor; and

receive a CA certificate response from the EST server, wherein the CA certificate response is signed by the EST server, and an EST signing certificate signed by the vendor CA; and

validate the CA certificate(s).

Typically, the CA certificate response is in a cryptographic message syntax (CMS) format.

The EST client can be configured to validate the EST signing certificate using the trust anchor, extract a public key of the EST server from the EST signing certificate, and use the public key to verify a signature of the CA certificate response.

In an embodiment, the EST client is configured to:

send an Establish Trust request to the EST Server, wherein the Establish Trust request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receive a response from the EST server comprising a TLS/DTLS EST server certificate signed by the EST server and an EST server signing certificate signed by the vendor CA;

validate the EST server signing certificate using the trust anchor;

extract a public key from the EST signing certificate; and

use the public key to validate the signed TLS/DTLS EST Server Certificate.

The EST client can be further configured to terminate the non-trusted TLS/DTLS session and initiate a new TLS/DTLS handshake with the EST server to establish a trusted TLS/DTLS session. For example, after validating the CA certificate response or the TLS/DTLS EST server certificate. Alternatively, the EST client is further configured to validate the non-trusted TLS/DTLS session to establish a trusted TLS/DTLS session.

According to a second aspect there is provided a method performed at an apparatus comprising an Enrollment over Secure Transport (EST) client, the method comprising:

storing a trust anchor;

initiating a Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) handshake with an EST server;

receiving a TLS/DTLS EST server certificate from the EST server;

performing validation of the TLS/DTLS EST server certificate using the trust anchor; and

when validation fails, completing the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session with the EST server.

The method may further comprise:

sending a certificate authority (CA) certificate request to the EST server using the non-trusted TLS/DTLS session; and

receiving one or more CA certificates from the EST server.

The method may further comprise storing the CA certificate(s) in temporary or volatile memory.

The method may further comprise:

sending a validation request to the EST server for validating the CA certificate(s), wherein the validation request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receiving a validation response from the EST server, wherein the validation response comprises a CA certificate response signed by the EST server and an EST signing certificate signed by the vendor CA; and

validating the CA certificate(s).

The method may further comprise, after validating the CA certificate(s), move the CA certificate(s) to a trust anchor store. For example, moving the CA certificate(s) to an explicit TA store/database.

The method may further comprise:

sending a CA certificate retrieval request to the EST server using the non-trusted TLS/DTLS session, wherein the CA certificate retrieval request comprises a vendor identifier associated with a vendor CA associated with the trust anchor; and

receiving a CA certificate response from the EST server, wherein the CA certificate response is signed by the EST server, and an EST signing certificate signed by the vendor CA; and

validating the CA certificate(s).

The CA certificate response may be in a cryptographic message syntax (CMS) format.

The method may further comprise validating the EST signing certificate using the trust anchor, extracting a public key of the EST server from the EST signing certificate, and using the public key to verify a signature of the CA certificate response.

The method may further comprise:

sending an Establish Trust request to the EST Server, wherein the Establish Trust request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receiving a response from the EST server comprising a TLS/DTLS EST server certificate signed by the EST server and an EST server signing certificate signed by the vendor CA;

validating the EST server signing certificate using the trust anchor;

extracting a public key from the EST signing certificate; and

using the public key to validate the signed TLS/DTLS EST Server Certificate

The method may further comprise terminating the non-trusted TLS/DTLS session and initiating a new TLS/DTLS handshake with the EST server to establish a trusted TLS/DTLS session. Alternatively, the method may comprise validating the non-trusted TLS/DTLS session to establish a trusted TLS/DTLS session.

According to a third aspect there is provided a system comprising:

an EST server configured to:

generate an asymmetric key pair;

send a certificate signing request (CSR) to a vendor CA;

receive a EST signing certificate from the vendor CA and associate the certificate with a vendor identifier associated with the vendor CA;

sign one or more certificate authority (CA) certificates using a private key of the asymmetric key pair;

an EST client configured to:

store a trust anchor associated with the vendor CA;

initiate a Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) handshake with the EST server;

receive a TLS/DTLS EST server certificate from the EST server;

perform validation of the TLS/DTLS EST server certificate using the trust anchor; and

when validation fails, complete the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session with the EST server.

The EST client can be further configured to:

send a certificate authority (CA) certificate retrieval request to the EST server using the non-trusted TLS/DTLS session;

receive one or more CA certificates from the EST server;

send a validation request to the EST server for validating the CA certificate(s), wherein the validation request comprises a vendor identifier associated with the vendor CA;

receive a validation response from the EST server, wherein the validation response comprises a CA certificate response signed by the EST server and an EST signing certificate signed by the vendor CA; and

validate the CA certificate(s).

According to a fourth aspect there is provided a method of establishing a trusted Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) session between an EST client and an EST server, the method comprising:

storing a trust anchor at the EST client, wherein the trust anchor is associated with a vendor CA;

initiating a TLS/DTLS handshake between the EST client and the EST server;

sending a TLS/DTLS EST server certificate from the EST server to the EST client;

performing validation of the TLS/DTLS certificate using the trust anchor; and

when validation fails, completing the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session between the EST client and the EST server.

Brief Description of Drawings

Figure 1 illustrates a signal flowchart for an EST procedure;

Figure 2 illustrates a signal flowchart for another EST procedure;

Figure 3 illustrates a signal flowchart for another EST procedure;

Figure 4 illustrates a signal flowchart for another EST procedure; and

Figure 5 illustrates a schematic diagram of an apparatus according to an embodiment.

Detailed Description

Figure 1 illustrates a signal flowchart 100 for an EST procedure. At step 101, the EST client is loaded with an implicit trust anchor (TA). For example, the EST client may be comprised by a smart meter, a modem or other apparatus, and the implicit TA can be

provided at the factory by the manufacturer. The implicit TA is then available on the client or server for use during EST TLS/DTLS authentication.

At step 102, the TA is loaded into the EST server at installation.

At step 103, the EST client performs a TLS handshake with the EST server to establish a TLS session 104. The EST client validates the EST server using the implicit TA.

At step 105, within the TLS session the EST client sends a CA certificate request to the EST server. For example, the EST client can use the "/cacerts" function to request the EST CA TA database information of the CA (in the form of certificates). The EST client and the EST server both support the /cacerts function (or the CoAP equivalent).

At step 106, the EST server sends a certificate response (e.g. in PKCS7 format). For example, the HTTP content-type of "application/pkcs7-mime" can be used. A Simple PKI Response is sent. The EST server includes the current root CA certificate in the response. The EST server can also include any additional certificates the client would need to build a chain from an EST CA-issued certificate to the current EST CA TA. For example, if the EST CA is a subordinate CA, then all the appropriate subordinate CA certificates necessary to build a chain to the root EST CA can be included in the response.

After out-of-band validation occurs, other certificates can be validated using normal certificate path validation (using the most recent CA certificate as the TA) before they can be used to build certificate paths during certificate validation.

The EST client can store the extracted EST CA certificate as an Explicit TA for subsequent EST server authentication. The EST client may then disable use of the implicit TA for the EST server.

At step 107, the EST client sends a client certificate request to the EST server. The EST client can request a certificate from the EST server with the "/simpleenroll" function. The EST client can thereby obtain a certificate for itself by submitting an enrollment request to the EST server. The client typically includes a Simple PKI Request (e.g. a PKCS#10 Certification Request as defined in RFC2986 by the IETF).

A Certification Signing Request (CSR) signature can be used to provide proof-of-possession of the client-possessed private key to the EST server. The EST client then generates the CSR signature using the private key.

The EST client may request additional certificates, when using an existing certificate in the TLS client authentication. For example, the client can use an existing certificate for TLS client authentication, when requesting a certificate that cannot be used for TLS client authentication.

At step 108, the EST server sends a client certificate request to the operational CA.

At step 109, the EST server receives a requested certificate from the operational CA.

At step 110, the EST server sends a client certificate response to the EST client. The response may be a "certs-only CMC Simple PKI Response" (as defined in RFC5272 by the IETF), containing only the certificate that was issued by the operational CA. This ends the enrollment process.

Authentication of the EST-CoAPs server (also referred to as "EST server" herein) by the EST-CoAPs client (also referred to as "EST client" herein) is based on certificate authentication in the DTLS handshake. The EST-CoAPs client is pre-configured with a TA database containing the trusted anchor of the EST server, which will enable the client to authenticate the EST-CoAPs server the first time before updating its trust anchor for the customer (Operational TA). The EST procedure described in relation to Figure 1 above can also be used with EST-CoAP, but with DTLS used instead of TLS.

This architecture (and signal flow) works if, at device manufacturing, the EST Server trust anchors are known and can be injected into devices. But in the case where EST Server TAs are unknown at device manufacturing, then additional steps may be needed after device manufacturing to configure devices with correct TAs based on the customer. These additional steps can create issues related to the risk of configuration errors, scale issues, and inventory shipment redirection. Also, the problem becomes more complex, when devices from multiple manufacturers are to be deployed in a single deployment, which is typical for IoT technology.

Another potential problem is that of a compromised operational CA or rollover for operational CA. If an operational CA is compromised and utility is moved to a new operational CA, then devices will not be able to validate the new operational CA when attempting to acquire new operational certificates from the new operational CA. The method proposed with this invention can help in validating a new operational CA.

Embodiments provided herein can at least partly solve these problems by providing an improved EST procedure.

Figure 2 illustrates a signal flow chart 200 for an EST procedure. The procedure is also applicable to EST-CoAP. Each vendor is associated with and publishes an identifier, such as the IEEE OUI assigned to a manufacturer. The vendor may also be the manufacturer of the device.

At step 201, at manufacturing, devices are injected with a TA which is known and trusted by the vendor. For example, the EST client may be comprised by a smart meter or other apparatus, and the TA can be provided at the factory by the manufacturer from a CA trusted by the manufacturer.

At step 202, a server TA is loaded into the EST server (e.g. at installation).The TA is related to a TLS certificate of the EST client, and allows the EST server to validate the EST client.

At step 203, a new signing asymmetric key pair is generated by the EST Server. The asymmetric key pair. This signing key pair is different from the EST server's TLS/DTLS Server asymmetric key pair, in order to enforce separation of concerns.

At step 204, the EST Server generates a Certificate Signing Request (CSR) for the new key pair, and sends the CSR to the vendor CA. Typically secure out of band techniques are used to send the CSR. In general, the EST server may send the CSR to each vendor of a plurality of vendors. Each vendor can then sign a respective certificate using its trusted CA (the vendor CA). The vendor CA corresponds to the TA, which was injected into the device with the EST client at manufacturing.

At step 205, the vendor CA sends a (vendor) signed certificate to the EST server for the new key pair. The certificate can be referred to as the EST signing certificate, which has been signed by the vendor CA.

At step 206, the EST server configures each certificate against a respective vendor identifier associated with the vendor (and vendor CA).

Steps 204 to 206 can be performed whenever devices from a different manufacturer are to be deployed.

At step 207, the EST client initialises a TLS/DTLS handshake 208 with the EST server.

At step 208, the EST Server exchanges its TLS/DTLS EST server certificate, which is signed using a customer specific PKI (different from the asymmetric key pair generated in step 203). This TLS/DTLS EST server certificate is different from the certificate that was received from the vendor CA during EST Server provisioning.

At step 209, the EST client performs certificate validation. Since the implicit TA of the EST client is not configured to trust the TLS/DTLS EST Server Certificate PKI chain, the EST client fails certificate validation. In conventional EST procedures, this would terminate the session establishment.

At step 210, instead of terminating the TLS/DTLS session, the EST client proceeds with the (non-trusted) TLS/DTLS Session. This step completes the TLS/DTLS handshake.

At step 211, the EST client sends CA Certificate retrieval request to the EST Server. This request is similar or equivalent to the CA certificate request of step 105 described in relation to Figure 1 above. For example, for a TLS session, the EST client can use HTTP and the “est/cacerts” path.

At step 212, the EST server sends one or more CA certificates to the EST client.

At step 213, since the client does not trust the TLS/DTLS session, the EST client stores the returned CA Certificates in volatile or temporary memory. This is different from the

procedure described in relation to Figure 1, where the EST client stores the CA certificates in an explicit TA store/database.

At step 214, the EST client sends a CA Certificate Validation request to the EST server. The request comprises the vendor identifier (e.g. using a new path "est/valcacerts/{vendor_identifier}" or CoAP equivalent).

At step 215, the EST Server recognizes the vendor identifier and processes the CA Certificate Validation request. The EST Server signs a CA Certificate response using the asymmetric private key created using the EST Server configuration (i.e. the new private key generated in step 203). The CA certificate response may include an expiration.

At step 216, the EST Server creates a CA certificate response signed with the private key. The response is typically in CMS (Cryptographic Message Syntax) format (defined by RFC5652 by the IETF).

At step 217, the EST server sends the CA certificate response and the EST server signing certificate associated with the asymmetric key pair (generated in step 203), and signed by the vendor CA, to the EST client. The EST client receives the response.

At step 218, the EST client validates the EST server's signing certificate included in the response using the TA, extracts the EST Server signing public key from the certificate upon successful validation, and uses this public key to validate the CA certificates associated with the signed CA certificate response. When the CA Certificates are thus validated, the CA Certificates are moved from volatile or temporary memory to the explicit TA store/database.

In one embodiment, to avoid large response messages, the CA Certificates may not be included in the CA certificate response, as the EST client already has a local copy from steps 212 and 213. The EST client can thus verify the received signature against the local copy.

At step 219, the EST client terminates the (non-trusted) EST session with the EST server.

At step 220, the EST client initiates a TLS/DTLS handshake with the EST server in order to establish a new (trusted) TLS/DTLS session. The EST client can use the explicit TA to validate the EST server. After establishing the trusted TLS/DTLS session, the EST procedure for enrolling the EST client may follow the steps as described in relation to Figure 1 above.

Figure 3 is a signalling flow chart showing an alternative EST procedure. Steps 301 to 310 are the same as steps 201 to 210 described in relation to Figure 2 above. Hence, the EST client establishes a non-trusted TLS/DTLS session with the Est server.

At step 311, instead of sending a conventional CA Certificate retrieval request (e.g. /est/cacerts) to the EST Server, the EST client sends CA certificate retrieval request with the vendor identifier (e.g. using a new path est/cacerts/{manufacturer_identifier}).

At step 312, the EST Server recognizes the vendor identifier and processes the CA Certificate retrieval request. The EST Server signs the CA Certificate response using the asymmetric private key created using EST Server configuration.

At step 313, the EST Server creates a CA certificate response. The response comprises the EST server's signing certificate associated with the asymmetric key pair (generated in step 203). The response is typically in CMS (Cryptographic Message Syntax) format (defined by RFC5652 by the IETF).

At step 314, the EST server sends the CA certificate response with the EST server signing certificate to the EST client.

At step 315, the EST client validates the EST Server's signing certificate, extracts the EST Server signing public key from the certificate upon successful validation, and uses this public key to validate the CA certificates. When the CA Certificates are thus validated, the CA Certificates are moved to an explicit TA store/database. The implicit TA may be removed.

After updating the TA store, Figure 3 shows two alternatives A, B for continuing the EST procedure.

Option A comprises steps 316 and 317, which are the same as steps as steps 219 and 220 described in relation to Figure 2 above, whereby the EST client terminates the TLS/DTLS session and initiates a new handshake. This procedure may be easy to implement with few if any changes to the existing EST protocol.

Option B provides a more optimal solution with a single step 318, but which may require a further update to the existing EST protocol, whereby the EST client does not terminate the non-trusted TLS/DTLS session, but instead validates the session to mark it as a trusted TLS/DTLS session.

Accordingly, at step 318, the EST Client validates the TLS/DTLS EST Server certificate and marks the TLS/DTLS Session as valid.

Figure 4 illustrates a signal flowchart for an EST procedure according to another embodiment. The embodiment is similar to the one illustrated in Figure 3 above. Steps 401 to 410 are the same as steps 301 to 310 described in relation to Figure 3 above.

At step 411, after a non-trusted TLS/DTLS session is established, the EST client sends an Establish Trust request to the EST Server with the vendor identifier (e.g. using a new path est/establishtrust/{manufacturer_identifier}). The EST client is not requesting CA certificates in this step (unlike in step 311 of Figure 3 described above). Instead, the EST client is first trying to establish trust with the EST server. Once trust has been established, a trusted TLS/DTLS session can be set up between the EST client and the EST server, over which the EST client can request and receive the CA certificates.

At step 412, the EST Server recognizes the vendor identifier and processes the Establish Trust request. The EST Server signs the TLS/DTLS EST Server Certificate Chain using the asymmetric private key created using the EST Server configuration (created at step 403).

At step 413, the EST Server creates a response (e.g. in CMS format) including the signed TLS/DTLS EST server certificate chain.

At step 414, the EST server sends the signed response with the EST server signing certificate (signed by the vendor CA) and the EST client receives the response and certificate.

At step 415, the EST client validates the EST server signing certificate included in the response, extracts the EST Server public key from the certificate upon successful validation and uses this public key to validate the signed TLS/DTLS EST Server Certificate Chain.

Steps 416 to 418 are the same as steps 316 to 318 described in relation to Figure 3 above. After step 418 the EST client can continue with a normal EST procedure (e.g. sending an /est/cacerts followed by est/simpleenrollment or CoAP equivalent messages to the EST server as described in relation to Figure 1 and 2 above)

Figure 5 shows an apparatus 500 for implementing through software and hardware an EST client as described herein. The apparatus comprises a transmitter 501 and a receiver 502 for transmitting/sending and receiving messages respectively (e.g. CoAP messages using DTLS). The apparatus 500 further comprises a processor 503 for processing messages and causing the EST client to perform the steps of the EST client of any one of Figures 1 to 4. The apparatus 500 further comprises a memory 504 for storing certificates. The memory 503 may also comprise volatile or temporary memory for storing CA certificates received from the EST server before validating the EST server.

While specific embodiments of the invention have been described above, it will be appreciated that the invention may be practiced otherwise than as described. The descriptions above are intended to be illustrative, not limiting. It will be apparent to one skilled in the art that modifications may be made to the invention as described without departing from the scope of the claims set out below.

Each feature disclosed or illustrated in the present specification may be incorporated in the invention, whether alone or in any appropriate combination with any other feature disclosed or illustrated herein.

CLAIMS:

1. An apparatus comprising an Enrollment over Secure Transport (EST) client, the EST client being configured to:

store a trust anchor;

initiate a Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) handshake with an EST server;

receive a TLS/DTLS EST server certificate from the EST server;

perform validation of the TLS/DTLS EST server certificate using the trust anchor; and

when validation fails, complete the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session with the EST server.

2. An apparatus according to claim 1, wherein the EST client is further configured to:

send a certificate authority (CA) certificate retrieval request to the EST server using the non-trusted TLS/DTLS session; and

receive one or more CA certificates from the EST server.

3. An apparatus according to claim 2, wherein the EST client is further configured to store the CA certificate(s) in temporary or volatile memory.

4. An apparatus according to claim 2 or 3, wherein the EST client is further configured to send a validation request to the EST server for validating the CA certificate(s), wherein the validation request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receive a validation response from the EST server, wherein the validation response comprises a CA certificate response signed by the EST server and an EST signing certificate signed by the vendor CA; and

validate the CA certificate(s).

5. An apparatus according to claim 4, wherein the EST client is further configured to, after validating the CA certificate(s), move the CA certificate(s) to a trust anchor store.

6. An apparatus according to claim 1, wherein the EST client is configured to:

send a CA certificate retrieval request to the EST server using the non-trusted TLS/DTLS session, wherein the CA certificate retrieval request comprises a vendor identifier associated with a vendor CA associated with the trust anchor; and

receive a CA certificate response from the EST server, wherein the CA certificate response is signed by the EST server, and an EST signing certificate signed by the vendor CA; and

validate the CA certificate(s).

7. An apparatus according to claim 4, 5 or 6, wherein the CA certificate response is in a cryptographic message syntax (CMS) format.

8. An apparatus according to any one of claims 4 to 7, wherein the EST client is configured to validate the EST signing certificate using the trust anchor, extract a public key of the EST server from the EST signing certificate, and use the public key to verify a signature of the CA certificate response.

9. An apparatus according to claim 1, wherein the EST client is configured to:

send an Establish Trust request to the EST Server, wherein the Establish Trust request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receive a response from the EST server comprising a TLS/DTLS EST server certificate signed by the EST server and an EST server signing certificate signed by the vendor CA;

validate the EST server signing certificate using the trust anchor;

extract a public key from the EST signing certificate; and

use the public key to validate the signed TLS/DTLS EST Server Certificate.

10. An apparatus according to any one of claims 4 to 9, wherein the EST client is further configured to terminate the non-trusted TLS/DTLS session and initiate a new TLS/DTLS handshake with the EST server to establish a trusted TLS/DTLS session.

11. An apparatus according to any one of claims 4 to 9, wherein the EST client is further configured to validate the non-trusted TLS/DTLS session to establish a trusted TLS/DTLS session.

12. A method performed at an apparatus comprising an Enrollment over Secure Transport (EST) client, the method comprising:

storing a trust anchor;

initiating a Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) handshake with an EST server;

receiving a TLS/DTLS EST server certificate from the EST server;

performing validation of the TLS/DTLS EST server certificate using the trust anchor; and

when validation fails, completing the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session with the EST server.

13. A method according to claim 12, further comprising:

sending a certificate authority (CA) certificate request to the EST server using the non-trusted TLS/DTLS session; and

receiving one or more CA certificates from the EST server.

14. A method according to claim 13, further comprising storing the CA certificate(s) in temporary or volatile memory.

15. A method according to claim 13 or 14, further comprising:

sending a validation request to the EST server for validating the CA certificate(s), wherein the validation request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receiving a validation response from the EST server, wherein the validation response comprises a CA certificate response signed by the EST server and an EST signing certificate signed by the vendor CA; and

validating the CA certificate(s).

16. A method according to claim 15, after validating the CA certificate(s), move the CA certificate(s) to a trust anchor store.

17. A method according to claim 12, further comprising:

sending a CA certificate retrieval request to the EST server using the non-trusted TLS/DTLS session, wherein the CA certificate retrieval request comprises a vendor identifier associated with a vendor CA associated with the trust anchor; and

receiving a CA certificate response from the EST server, wherein the CA certificate response is signed by the EST server, and an EST signing certificate signed by the vendor CA; and

validate the CA certificate(s).

18. A method according to claim 15, 16 or 17, wherein the CA certificate response is in a cryptographic message syntax (CMS) format.

19. A method according to any one of claims 15 to 18, further comprising validating the EST signing certificate using the trust anchor, extracting a public key of the EST server from the EST signing certificate, and using the public key to verify a signature of the CA certificate response.

20. A method according to claim 12, further comprising:

sending an Establish Trust request to the EST Server, wherein the Establish Trust request comprises a vendor identifier associated with a vendor CA associated with the trust anchor;

receiving a response from the EST server comprising a TLS/DTLS EST server certificate signed by the EST server and an EST server signing certificate signed by the vendor CA;

validating the EST server signing certificate using the trust anchor;

extracting a public key from the EST signing certificate; and

using the public key to validate the signed TLS/DTLS EST Server Certificate

21. A method according to any one of claims 15 to 20, further comprising terminating the non-trusted TLS/DTLS session and initiating a new TLS/DTLS handshake with the EST server to establish a trusted TLS/DTLS session.

22. A method according to any one of claims 15 to 20, further comprising validating the non-trusted TLS/DTLS session to establish a trusted TLS/DTLS session.

23. A system comprising:

an EST server configured to:

generate an asymmetric key pair;

send a certificate signing request (CSR) to a vendor CA;

receive a EST signing certificate from the vendor CA and associate the certificate with a vendor identifier associated with the vendor CA;

sign one or more certificate authority (CA) certificates using a private key of the asymmetric key pair;

an EST client configured to:

store a trust anchor associated with the vendor CA;

initiate a Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) handshake with the EST server;

receive a TLS/DTLS EST server certificate from the EST server;

perform validation of the TLS/DTLS EST server certificate using the trust anchor; and

when validation fails, complete the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session with the EST server.

24. A system according to claim 23, wherein the EST client is further configured to:

send a certificate authority (CA) certificate retrieval request to the EST server using the non-trusted TLS/DTLS session;

receive one or more CA certificates from the EST server;

send a validation request to the EST server for validating the CA certificate(s), wherein the validation request comprises a vendor identifier associated with the vendor CA;

receive a validation response from the EST server, wherein the validation response comprises a CA certificate response signed by the EST server and an EST signing certificate signed by the vendor CA; and

validate the CA certificate(s).

25. A method of establishing a trusted Transport Layer Security or Datagram Transport Layer Security (TLS/DTLS) session between an EST client and an EST server, the method comprising:

storing a trust anchor at the EST client, wherein the trust anchor is associated with a vendor CA;

initiating a TLS/DTLS handshake between the EST client and the EST server;

sending a TLS/DTLS EST server certificate from the EST server to the EST client;

performing validation of the TLS/DTLS certificate using the trust anchor; and

when validation fails, completing the TLS/DTLS handshake to establish a non-trusted TLS/DTLS session between the EST client and the EST server.