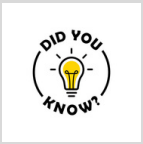
ECCO SP SECURITY FEATURES

AT A GLANCE

ECCo SP is a software for creating a secure, centrally managed network for data exchange. With ECCo SP, business applications can confidently exchange data with other organizations that are part of the same network. Unlike traditional methods like SFTP, SMTP, and MFT, ECCo SP automatically manages the PKI (Public Key Infrastructure) and allows only authorized participants to communicate. Additionally, ECCo SP stands apart from AS4 by being centrally governed and allowing for endpoints to be placed within a protected network zone, ensuring that they are shielded from external threats.

The key features of ECCo SP:

- All messaging offers Confidentiality, Authentication, Integrity and Non-Repudiation.
- The network can be scaled to support High Availability.
- Access to the network is restricted through a registration process and controlled centrally. Compromised endpoints can be excluded from the network.
- Secure Software Development Lifecycle, which guarantees regular vulnerability scanning, continuous updating of the software and immediate patching if necessary.

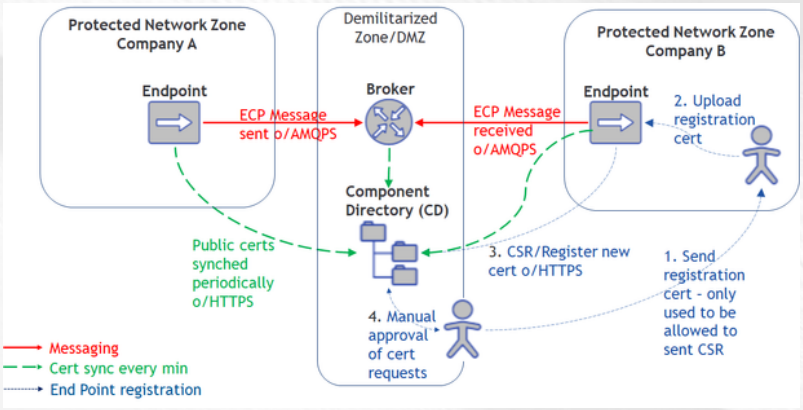


Over the past **4 years**, we have undergone 7 penetration tests (3 external & 4 internal) and fixed 26 issues as a result. Furthermore, we are proud to say that we have not experienced a compromised network or endpoint during this time. This historicity serves to bolster our claim of providing a safe system, and we will continue to prioritize security measures to maintain the trust of our users and keep their information protected.

ACCESS CONTROL

To be part of an ECCo SP network, a user/organization must ask the network owner to get access. As soon as registration has been completed, the endpoint will have the possibility to send/receive messages to the entire network (although some endpoints may restrict access to specific endpoints).

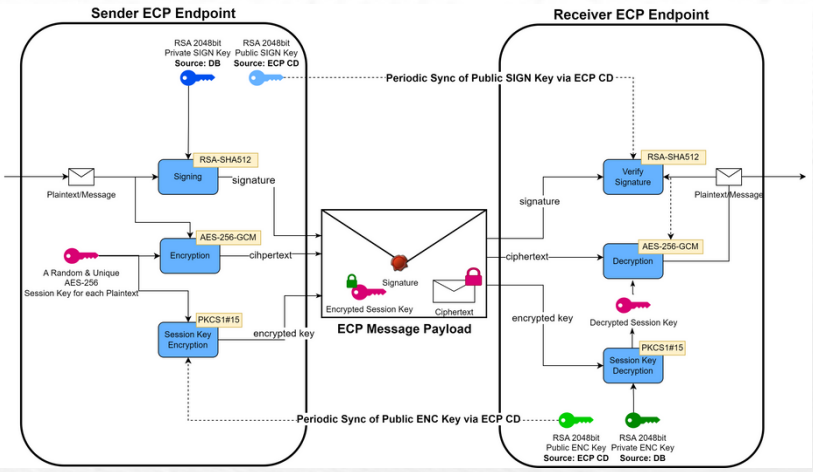
- A new endpoint registration requires both a special registration cert to register AND must be manually approved. When this happens the endpoint has 3 certificates that is valid in the network:
  - **AUTHENTICATION/AUTH** certificate – used for connecting over AMQPS to a broker or HTTPS to the CD, endpoint without valid AUTH cert cannot establish connection.
  - **ENCRYPTION/ENC** certificate – used to encrypt the message.
  - **SIGNING/SIGN** certificate – used to sign the message.
- Certificates are renewed automatically every 11 months by default.



- The messages are encrypted end-to-end with ENC/SIGN-certificates, while the AMQPS/HTTPS connections are encrypted using the AUTH-certificates, making effectively two layers of encryption.
- Each Broker/Endpoint can define a whitelist of endpoints and message types that is allowed, if detailed access control restrictions are wanted.

ENCRYPTION

- ECP uses standard encryption algorithms and behaves much the same way as TLS/SSL with client-certificate.
- Encryption process:
  - Plain-text (from Business Application) and sender-info is hashed, then encrypted with private blue Sender key (also known as the SIGN key/cert). This becomes the "signature"
  - Then the plain-text is encrypted with a random 256-bit key for every message, using the AES algorithm. This becomes the ciphertext.
  - Then the 256-bit key is encrypted using a 2048-bit key (AKA ENC key/cert) (the same key every time) using the RSA algorithm. The becomes the encrypted session key.
  - All of this is transmitted in one "ECP Message".
  - The receiver then decrypts the session key using its own private RSA-key.
  - With the decrypted session key in place, the receiver can then proceed to "unlock" the plain-text.
  - Lastly the receiver makes the same signature based on the plain-text and sender-info and can see that it is the same signature.



- This process takes care of :
  - **AUTHENTICATION:** Sender can't deny sending a message, because sender use a unique private (dark blue) key to sign it, and only sender's public (light blue) key can decrypt it. Thus, it's proven that the signature comes from the sender.
  - **NON-REPUDIATION:** Not shown here, but the receiving endpoint will return an ACK to Sender, in the same manner. Upon reception of this ACK, the Sender will know that the Receiver has indeed received the message.
  - **INTEGRITY:** The signature verification would fail if some bits were changed – the hash you make upon receiving will not match the one sent.
  - **CONFIDENTIALITY:** Only the receiver can decrypt the session key, thus reading the ciphertext, since it's the receivers public key that has been used.

VULNERABILITY MANAGEMENT

As part of ENTSO-E custom Secure Software Development Lifecycle (SSDLC) ECCo SP has implemented some protocols to identify the vulnerabilities, which includes the following activities:

- Vulnerability scanning
  - By Supplier - Done for each release and reports are available as part of each release package under the documentation folder.
  - By 3rd Party (ENCS)
    - Every week code is pushed into ENTOS-E Git by the supplier.
    - Automated Scan is run by the ENCS team.
    - Tickets are logged in JIRA for each issue found.
    - Any Critical/High tickets are fixed with 7 working days.
- Threat Modeling & Penetration Testing
  - Once a year ENTOS-E requests an independent organization (3rd party) to perform the threat modeling & penetration testing for ECCo SP.