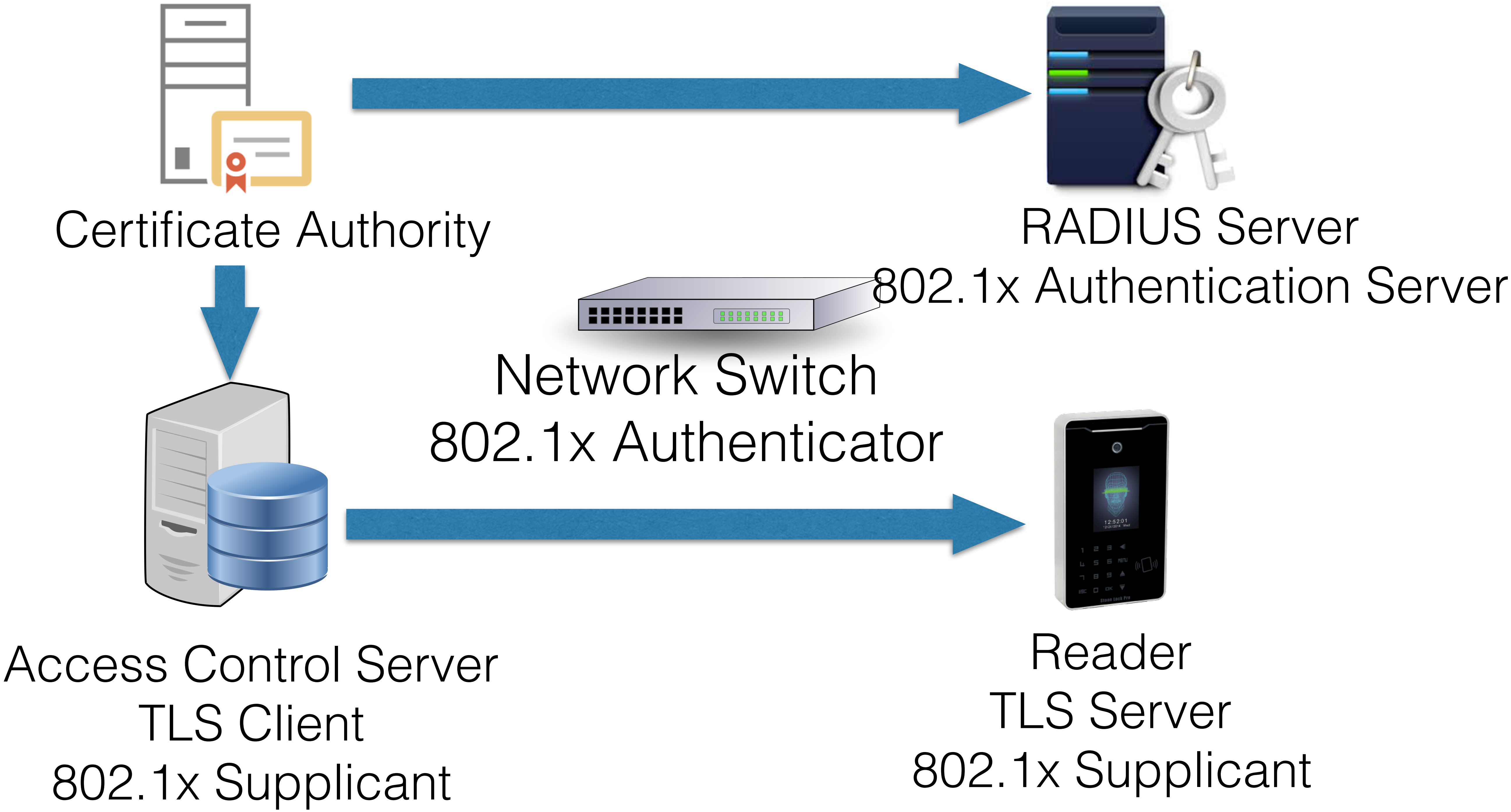
TLS/802.1x Credential Workflow

Key Components



Certificate Authority

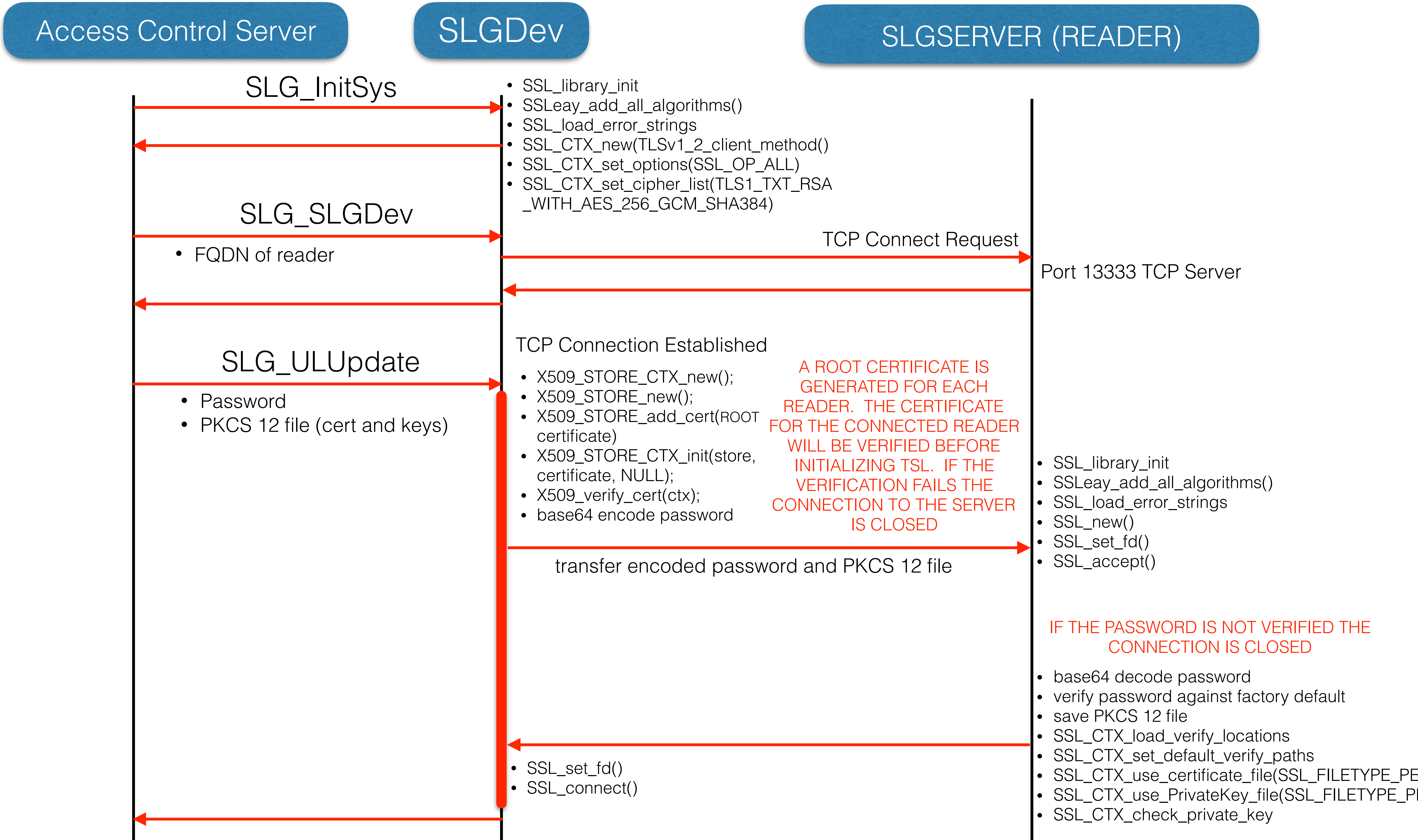
RADIUS Server
802.1x Authentication Server

Network Switch
802.1x Authenticator

Access Control Server
TLS Client
802.1x Supplicant

Reader
TLS Server
802.1x Supplicant

TLS Time Sequence Diagram



Access Control Server

SLGDev

SLGSERVER (READER)

SLG_InitSys

- SSL_library_init
- SSLeay_add_all_algorithms()
- SSL_load_error_strings
- SSL_CTX_new(TLSv1_2_client_method()
- SSL_CTX_set_options(SSL_OP_ALL)
- SSL_CTX_set_cipher_list(TLS1_TXT_RSA _WITH_AES_256_GCM_SHA384)

SLG_SLGDev

- FQDN of reader

TCP Connect Request

Port 13333 TCP Server

TCP Connection Established

SLG_ULUpdate

- Password
- PKCS 12 file (cert and keys)

- X509_STORE_CTX_new();
- X509_STORE_new();
- X509_STORE_add_cert(ROOT certificate)
- X509_STORE_CTX_init(store, certificate, NULL);
- X509_verify_cert(ctx);
- base64 encode password

A ROOT CERTIFICATE IS GENERATED FOR EACH READER. THE CERTIFICATE FOR THE CONNECTED READER WILL BE VERIFIED BEFORE INITIALIZING TSL. IF THE VERIFICATION FAILS THE CONNECTION TO THE SERVER IS CLOSED

- SSL_library_init
- SSLeay_add_all_algorithms()
- SSL_load_error_strings
- SSL_new()
- SSL_set_fd()
- SSL_accept()

transfer encoded password and PKCS 12 file

IF THE PASSWORD IS NOT VERIFIED THE CONNECTION IS CLOSED

- base64 decode password
- verify password against factory default
- save PKCS 12 file
- SSL_CTX_load_verify_locations
- SSL_CTX_set_default_verify_paths
- SSL_CTX_use_certificate_file(SSL_FILETYPE_PE
- SSL_CTX_use_PrivateKey_file(SSL_FILETYPE_PI
- SSL_CTX_check_private_key

- SSL_set_fd()
- SSL_connect()

TLS SEQUENCE OF OPERATION

1. The Access Control Server (ACS) calls the SLGDev library method,SLG_InitSys. This calls initializes the openssl library.
2. The ACS calls the SLGDev library method, SLG_SLGDev passing the FQDN as an argument. The library establishes a TCP connection with the server (reader).
3. If TLS is required, the ACS generates a key pair (public/private) using openssl and completes the following tasks.
4. The ACS requests an X.509 certificate from the Certificate Authority (CA) for the reader (device)
5. The CA generates a certificate based on the Fully Qualified Domain Name of the reader
6. If a CA is not configured the ACS will generate a self-signed certificate
7. The name of the certificate will be the FQDN of the reader
8. The ACS stores this certificate in the directory ‘c:\\rootca'
9. The ACS packages the X.509 certificate and the key pair into a PKCS 12 file
10. The ACS reads the PKCS 12 file as binary data
11. The ACS calls the CPMDev library method CPM_ULUpdate to transfer the certificate binary contents to the reader with a base64 encrypted password.
12. The library verifies the root certificate. If the certificate cannot be verified for the requested FQDN or the certificate is expired, the client TCP connection is closed.
13. The CPMDev library completes a network transfer of the byte array and password through port 13333
14. The reader authenticates the CPM_ULUpdate message with the transferred password
15. If the password is authenticated, the reader saves the PKCS 12 byte array as a file accessible by openssl, otherwise the client TCP connection is closed.
16. The reader openssl library will verify the certificate and read the PKCS 12 keys
17. The reader will initialize the TLS/TCP server with the read keys as the public key
18. If the certificate is expired, the CPMDev will send a callback certificate failure message and close the connection
19. If the certificate is expired, the ACS will request a new certificate, generate new keys and restart the process
20. The server openssl will periodically check the certificate while the connection is open
21. The CPMDev library TLS client will make a “client hello” call to the reader
22. The reader’s TLS server will reply with a “server hello” including the certificate and public key received in the PKCS 12 file
23. The TLS client uses the server’s public key to encrypt the random byte string to derive the message keys
24. The TLS client sends the secret key information to the server
25. The server and client exchange ‘finished’ messages
26. Bi-directional message transmissions start with the new shared secret key

Credentialing Steps 802.1x