OpenSSL in 21CS VSEⁿ

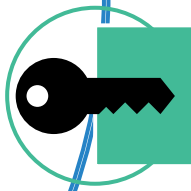
Shahin R Krishna
VSEⁿ Software Engineer

Introduction to SSL

What is Secure Sockets Layer (SSL)

Encryption based network security protocol

Developed & released by Netscape in 1995.



Ensures privacy, authentication and data integrity.

SSL evolved into Transport Layer Security(TLS)

SSL & TLS

Secure Sockets Layer(SSL)

- Originally Designed and Released by Netscape
- SSL 1.0 never publicly released
- SSL 2.0 released in 1995
  - Major vulnerabilities found and lead to development of SSL 3.0
- SSL 3.0 released in 1996
  - Was widely used till 2014 – Google Security team finds a major security vulnerability

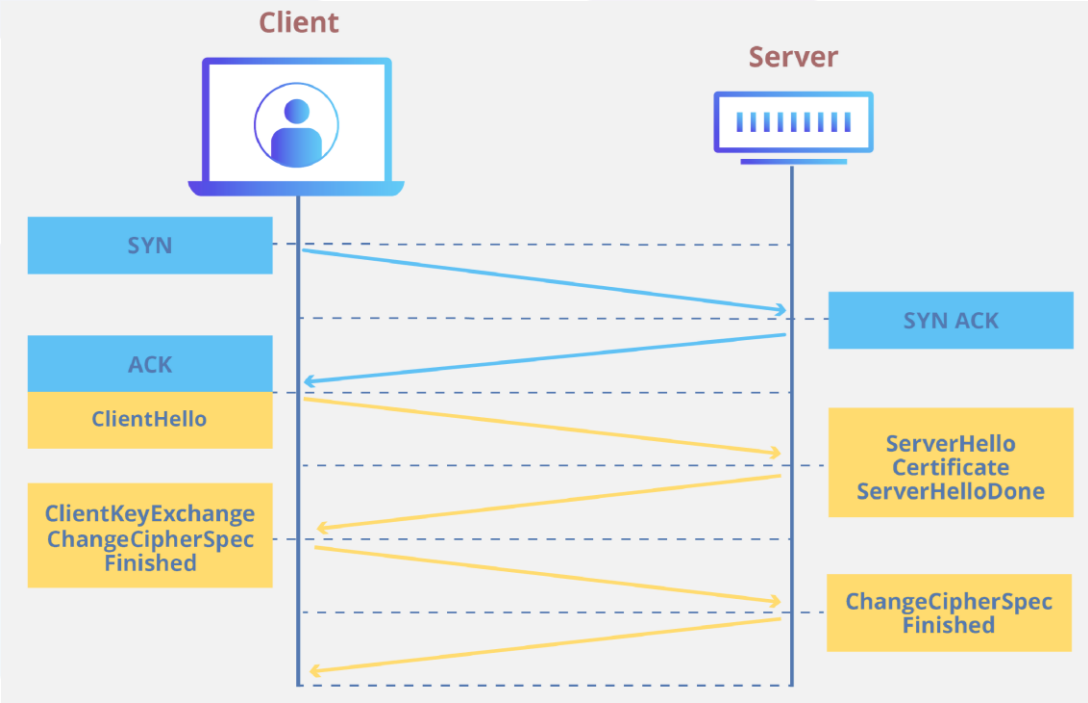
Transport Layer Security(TLS)

- Designed as an upgrade of SSL 3.0 in 1999
- SSL 3.0 & TLS 1.0 can’t interoperate
- Current Version is TLS V1.3 [TLS V1.0, TLS V1.1, TLS V1.2]

Secure Communication

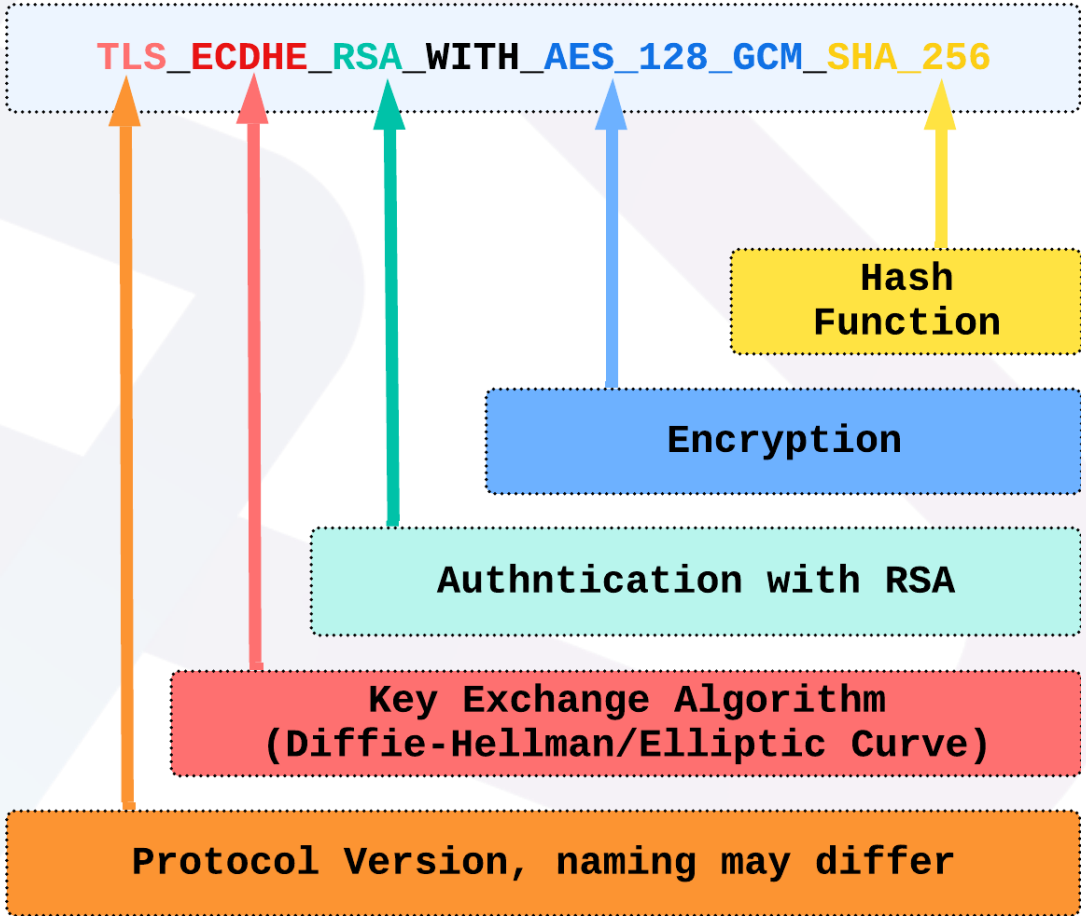
TLS Handshake

- Version of TLS to Use
- Cipher suite selection
- Authenticate the identity of the server using server's certificate
- Key Exchange
- Generate session keys for encrypting messages



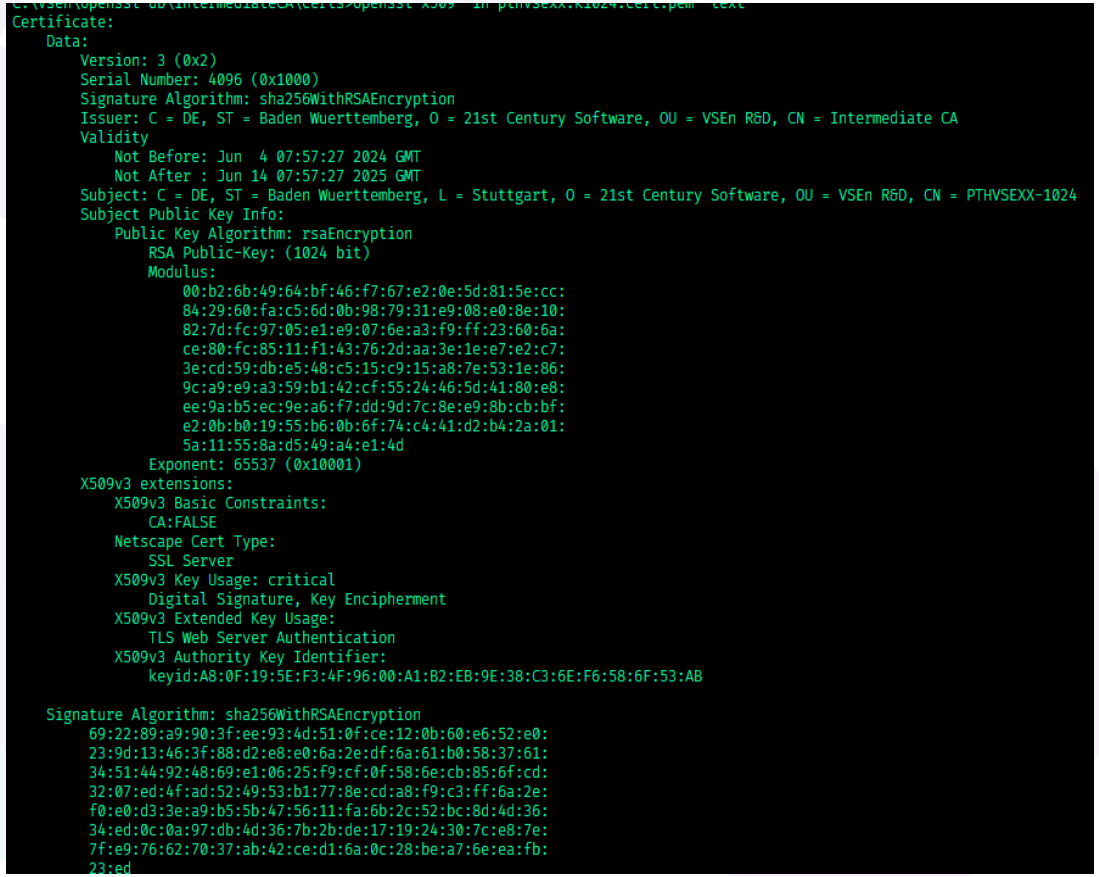
Cipher Suite

- A collection of cryptographic algorithms.
  - Hash Functions
  - Asymmetric encryption algorithm for handshake
  - Symmetric encryption algorithm to maintain session security



SSL Certificate

- A small data file that contains
  - Domain Name
  - Owner (Company, Device or person)
  - Subdomain names
  - Issuing Certificate Authority (CA or Intermediate CA)
  - CA’s or Intermediate CA’s digital signature
  - Issuance Date
  - Expiration Date
  - Public Key



OpenSSL in VSE^{n}

OpenSSL in VSE^{n}

- OpenSSL 1.1.1t (Current)
  - VP00081 (VA00077)
    - CVE-2023-0286
    - CVE-2023-0215
    - CVE-2022-4450
    - CVE-2022-4304
  - VP00107 (VA00113)
    - CVE-2023-0464
    - CVE-2023-0465
    - CVE-2023-2650
    - CVE-2023-3446
    - CVE-2023-3817
- OpenSSL 3.0 (Underway)

Preparing for SSL enablement

- Setup TCP/IP Socket API Multiplexer - EDCTCPMC
- Generate Certificates
- Upload Key and Certificate as PEM to VSE^{n}
- Configure applications to use SSL

TCP/IP Socket API Multiplexer - EDCTCPMC

Sample Job: ICCF.62(EDCTCPMC)

```
EDCTCPME SYSID=‘nn',PHASE=‘xxxxxxxx',SSLPHASE='IJBSSLLE',
      EZAPHASE='xxxxxxxx',EZASSL='IJBEZAOS’
```

** For OLTP Web Support you may have to add the following in your OLTP startup job

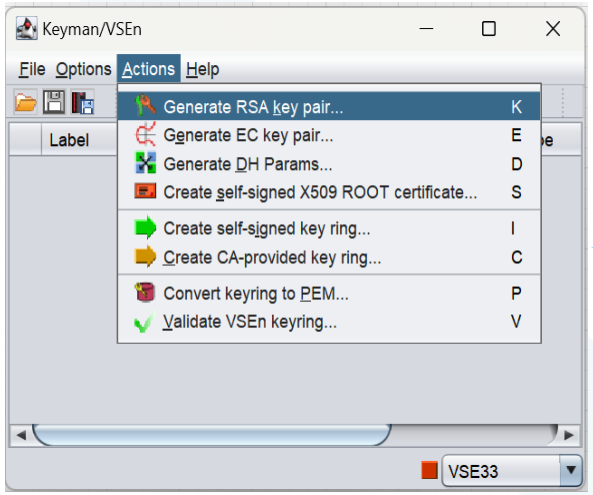
```
// SETPARM [SYSTEM,]BPX$GSK=‘IJBGSKOS’
```

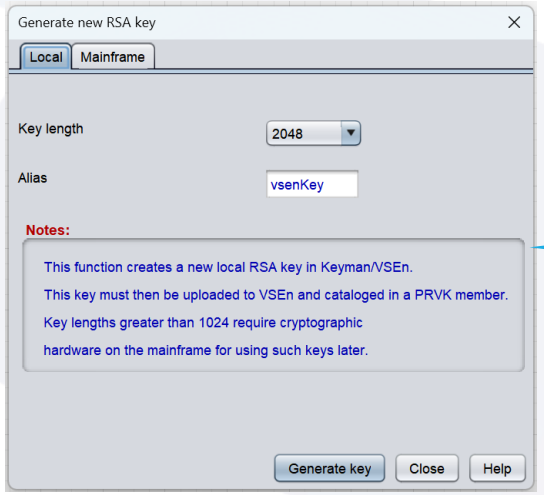
Generating Certificates

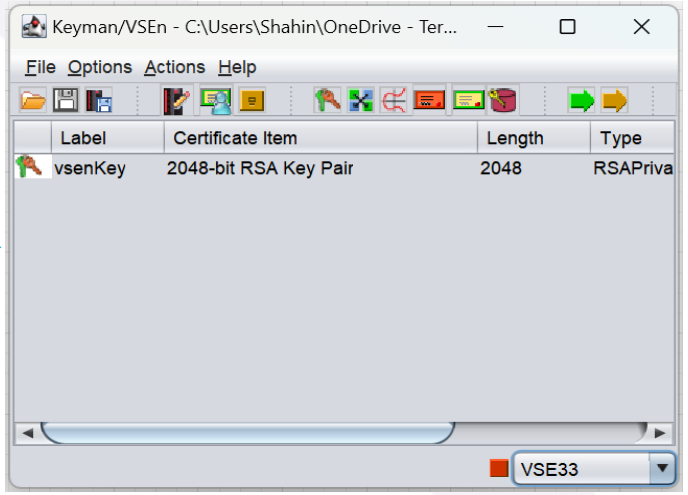
- To generate certificates, You can use
  - VSEⁿ Keyman
  - OpenSSL CLI Tool
  - Java Keytool

Generating Certificate using VSE^{n} Keyman

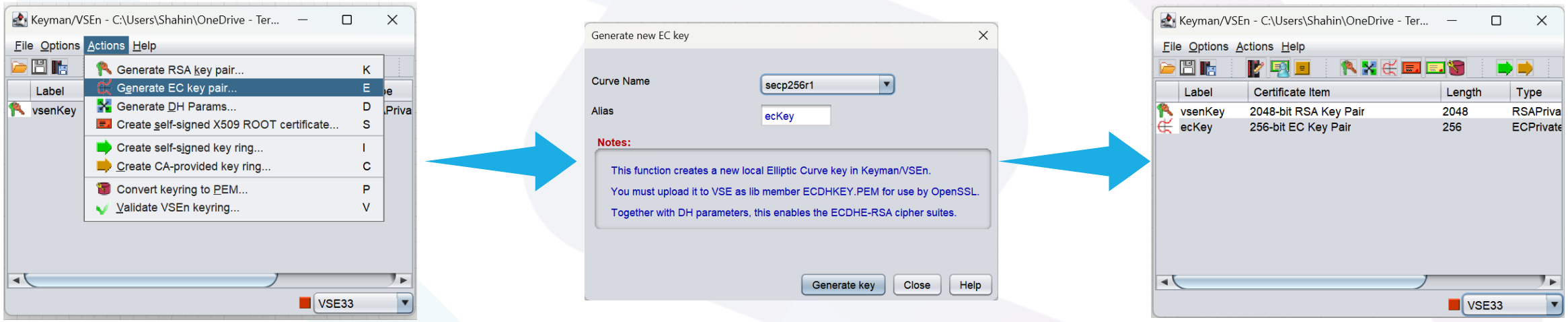
Generate Key Pair



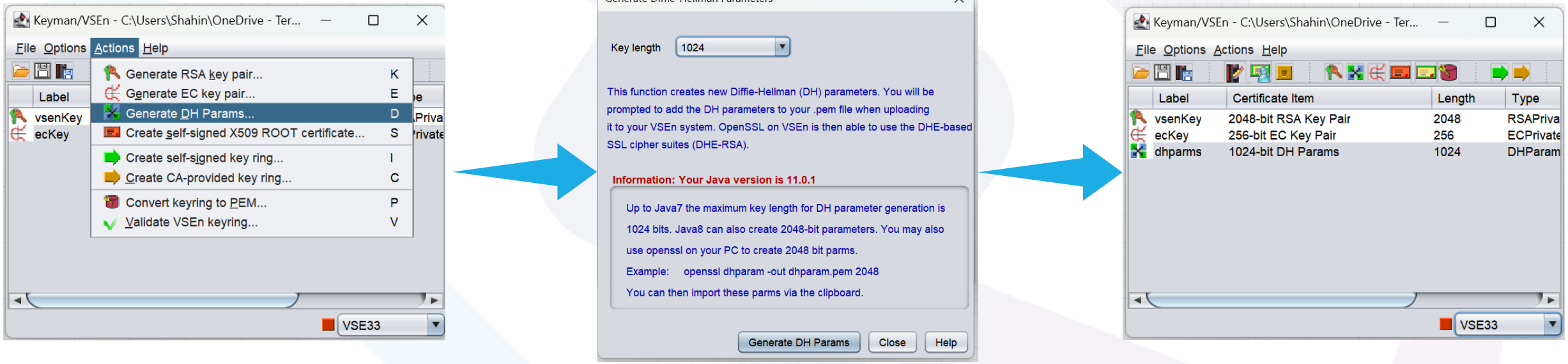




Generate EC Key Pair

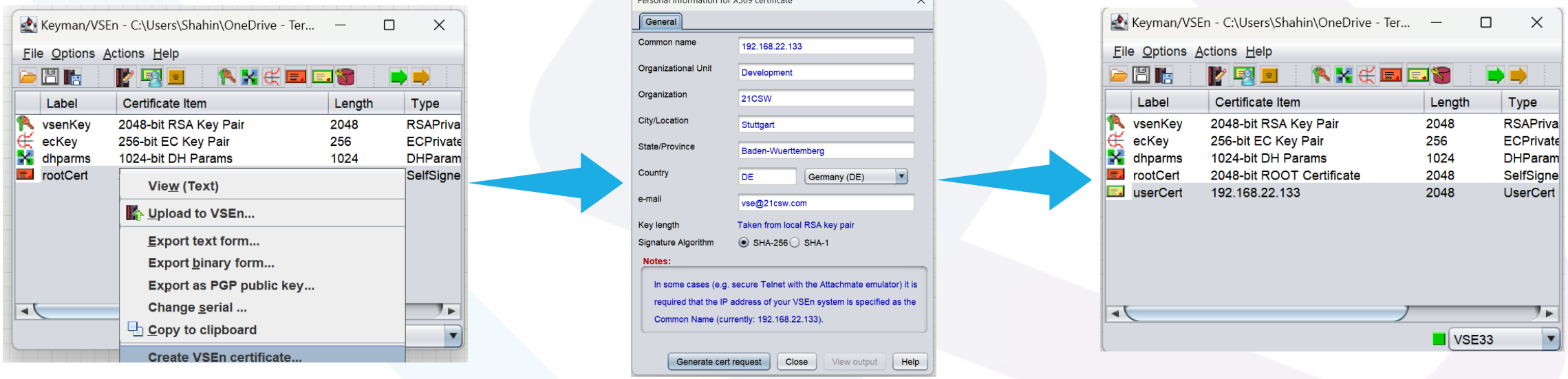


Generate DH Parameters

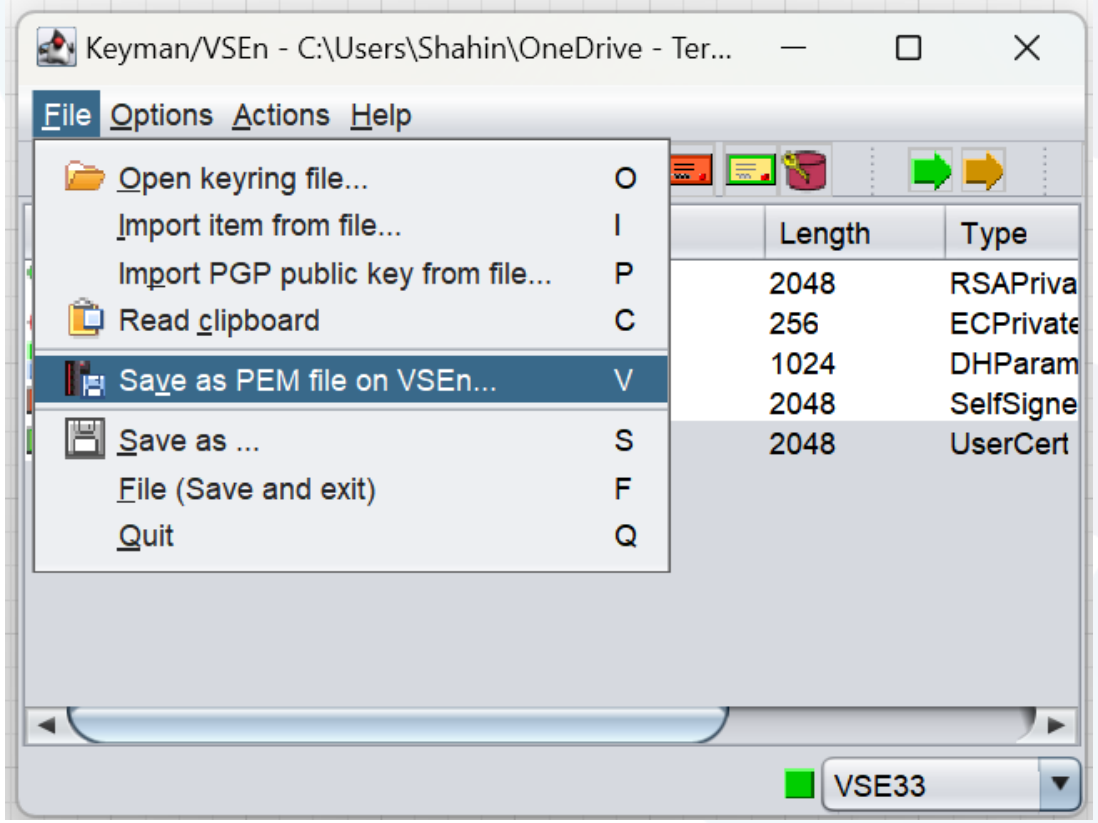


Generate Self Signed Root Certificate

Generate Server Certificate



Uploading Certificate to VSE^{n}



- Use File->Save As to save the generated certificates in PEM format
- OpenSSL supports only Privacy Enhanced Mail(PEM) format certificates

OpenSSL & VSE^{n} TCP/IP Stacks

IPV6 for VSE^{n}

- Uses OpenSSL as SSL Engine

TCP/IP for VSE^{n}

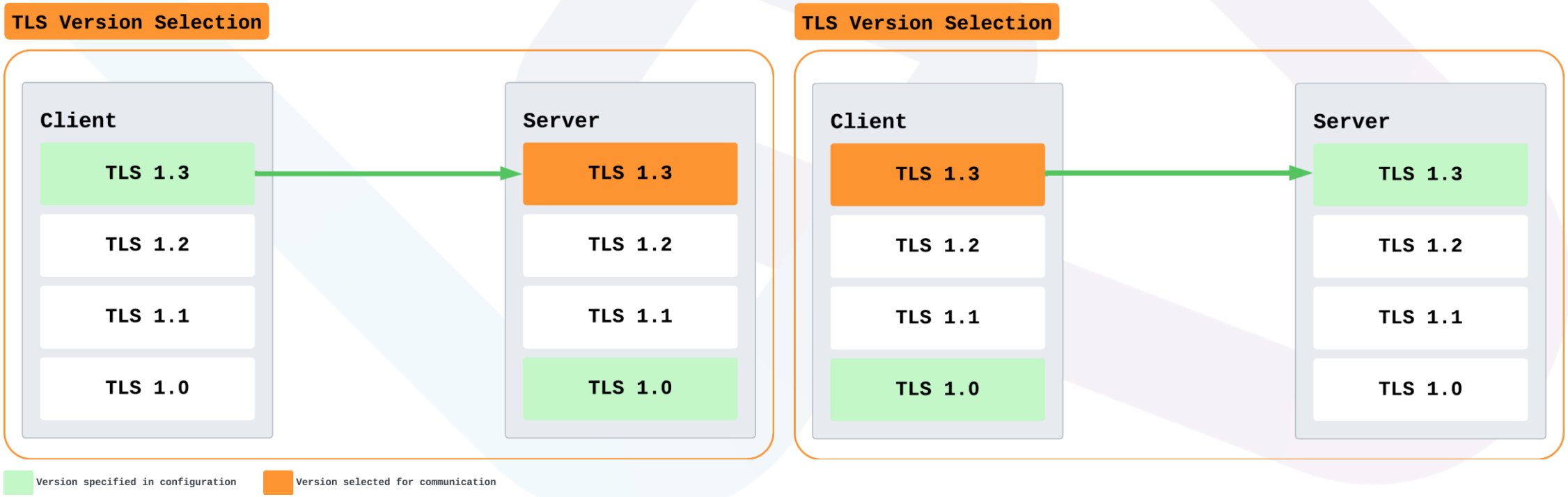
- By default, uses builtin SSL Engine (**CSINTSSL**); supports up to TLSV1.2
- To switch to OpenSSL
  - Configure EDTCPMC to use IJBSSLE as SSLPHASE
  - Include **SET OPENSSL ON** in IPINIT
  - Alternatively:
    use **// SETPARM CRYSRV='OPENSSL'**

SETPARMs for SSL control

```
// SETPARM SSL$MIN='[TLSV1 | TLSV1.2 | TLSV1.3]'
// SETPARM SSL$MAX='[TLSV1 | TLSv1.2 | TLSV1.3]'
// SETPARM SSL$CPH='C027C014C013C012'
// SETPARM SSL$DBG='[YES|NO]'
// SETPARM BPX$GSK=IJBGSKOS & ENCRYPTION=STRONG in DFHSIT
// SETPARM SSL$TRC='DD:lib.sublib(membername.membertype)'
// SETPARM SSL$ICA='[YES|NO]'
// SETPARM SSL$CSI='[YES|NO]'
```

TLS Version Control

- Specifying a TLS version means – “This version or Higher”



TLS Version Control

- Use SETPARM SSL$MIN & SSL$MAX to control the minimum and maximum version supported dynamically.
  - E.g.,

    You can have VCS with **SSLVERSION = TLSV1.1**. and in VCS startup job

    // SETPARM SSL$MIN=‘TLSV1.3’

    // SETPARM SSL$MAX=‘TLSV1.3’

    This will force the VCS to use TLSV1.3

TLS Version Control

```
09-06-2024 09:44:56   ----------------------------------------------------
09-06-2024 09:44:56   Summary:
09-06-2024 09:44:56     OpenSSL version     = OpenSSL 1.1.1t  7 Feb 2023
09-06-2024 09:44:56     Compiler            = IBM z/OS XL C/C++ METAL
09-06-2024 09:44:56     sec_types           = [TLSV1.2]
09-06-2024 09:44:56       override min      = [TLSV1.3]
09-06-2024 09:44:56       override max      = [TLSV1.3]
09-06-2024 09:44:56     keyring             = [CRYPTO.KEYRING]
09-06-2024 09:44:56     keyring_pw          = []
09-06-2024 09:44:56     keyring_stash       = []
09-06-2024 09:44:56     V2_session_timeout  = 0
09-06-2024 09:44:56     V3_session_timeout  = 86400
09-06-2024 09:44:56     LDAP_server         = []
09-06-2024 09:44:56     LDAP_port           = 0
09-06-2024 09:44:56     LDAP_user           = []
09-06-2024 09:44:56     LDAP_password       = []
09-06-2024 09:44:56     LDAP_CA_roots       = 0
09-06-2024 09:44:56     auth_type           = 0
09-06-2024 09:44:56     &gsk_init_data      = 0x88406378
09-06-2024 09:44:56   ----------------------------------------------------
```

- If you have set SSL$DBG=YES, then you get to see the version override information in the Handshake summary.

Recommendations

Symmetric Encryption

- Advanced Encryption Standard(AES) - 128/256 bits

Asymmetric Encryption

- RSA(Rivest, Shamir, Adleman) - 2048 bits
- ECC (Elliptic Curve Cryptography) - with RSA, 256-bit curve

Hash Algorithms

- SHA-2 - 256 bits

SSL/TLS Protocol Versions

- TLS 1.3 (If available)
- TLS 1.2
- TLS 1.0/1.1

Thank You