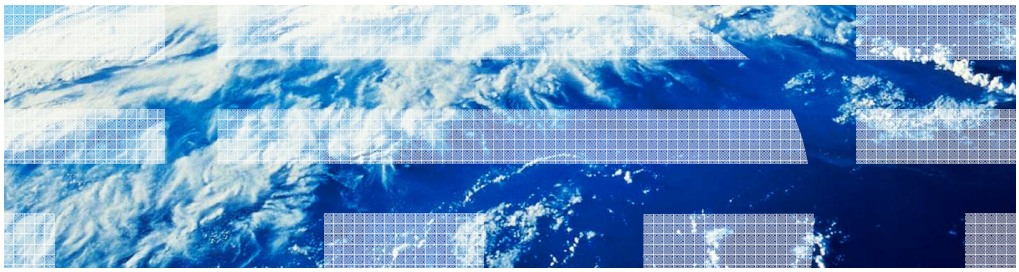
z/OS V1R13

SSL: ECC certificate creation and key agreement

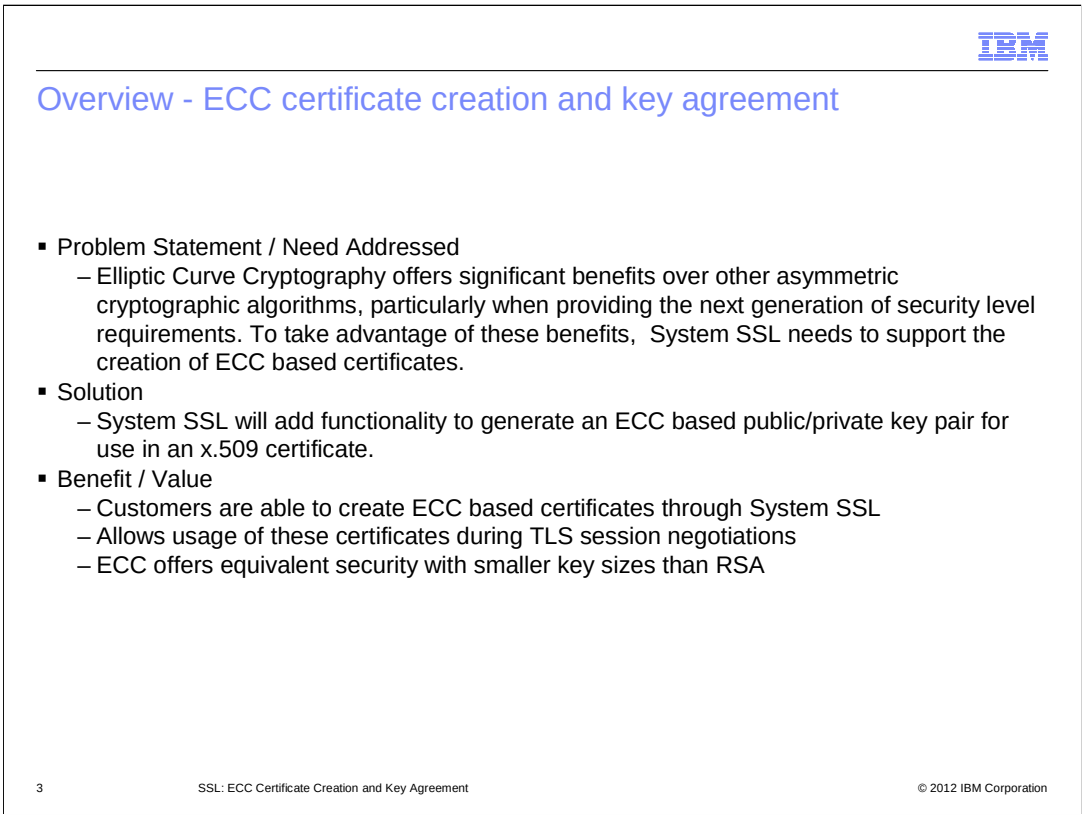


Session objectives

- At the end of this presentation, you should have an understanding of …
  - The System SSL line item enhancements for ECC certificate creation and key agreement
  - How to use the enhancements
  - Understand how these enhancements affect installation and migration

In V1R13, we are continuing to enhance our elliptic curve support. In V1R12 support was added for elliptic curve cryptography (ECC) certificates being imported into key database files and PKCS#11 tokens as well as the usage of the ECC certificates for digital signature sign and verification operations.

In V1R13 we are adding support for the generation of ECC certificates, ECC processing during the TLS handshake and usage of RACF certificates with their ECC private keys stored in ICSF's PKDS.



In 2005, the U.S. National Security Agency (NSA) announced Suite B Cryptography with implementation guidelines in 2009. Suite B cryptography includes Elliptic Curve Cryptography (ECC) for digital signature operations. ECC provides asymmetric cryptography along the same lines as RSA and DSA. Digital signature cryptography is used to authenticate the origin of data and protect the integrity of that data.

As higher levels of security is needed, one can either more to larger key sizes or move from RSA/DSA to elliptic curves. ECC is regarded as a faster algorithm that requires a smaller key than RSA cryptography. For example, a RSA 2048-bit key has equivalent strength to a 224-bit ECC key.

In z/OS V1R13, System SSL is going to expand on the support added in V1R12 to understand and process ECC style certificates and support the creation of ECC style certificates through gskkyman and the Certificate Management Services (CMS) APIs.

Overview - ECC certificate creation and key agreement

- Using the ECC creation line item, the customer can:
  - Create ECC certificates/requests through gskkyman
  - Create ECC certificates/requests through the Certificate Management APIs.
  - Store the certificates into key database files or PKCS#11 tokens
- Value:
  - Allows exploiters to create ECC certificates through gskkyman
  - For users using key database files or PKCS#11 tokens, ECC certificate support can be performed through the gskkyman command.
  - Applications currently using the CMS APIs to create and manage their certificates, can now create ECC style certificates.

Usage & Invocation (1 of 5)

- Management of certificates stored within key database files and PKCS#11 tokens is done through the gskkyman utility. The utility has been updated to support the creation of ECC certificates and certificate requests.
- When creating a new certificate using either option 6 of the Key/Token Management Menu (Create a self-signed certificate) or option 10 of the (Token) Key and Certificate Menu (Create a signed certificate and key) the Certificate Type Menu will offer two new options:
  - CA certificate with an ECC key
  - User or server certificate with an ECC key
- After selecting either option you will be prompted with a series of new menus to choose:
  - ECC Key Type (only for a user or server certificate)
  - ECC Curve Type
  - ECC Curve Name

System SSL's gskkyman certificate management utility has been updated to support ECC style certificates in both key database files and PKCS #11 tokens.

ECC certificates can be defined as either Certificate Authority Certificates or user/server certificates.

When creating the certificates, information is needed to determine how the certificate's ECC keys will be utilized, type of ECC key (NIST or Brainpool) and the actual curve value or name.

Usage & Invocation (2 of 5)

- When an ECC user or server certificate is requested the ECC Key Type Menu will appear:

```
ECC Key Type
   1 – General ECC key
   1 – ECDSA key
   3 - ECDH key

Select ECC key type (press ENTER to return to menu):
```

- This option affects the key usage that will be defined in the created certificate:
  - General ECC key - Digital Signature, Non-repudiation, Key Agreement
  - ECDSA key - Digital Signature, Non-repudiation
  - ECDH key - Key Agreement
- An ECC CA certificate will have a key usage allowing certificate and CRL signing.

gskkyman provides the ability to create ECC certificates that are designated for particular usages. This is achieved through the key usage extension that is present in the x.509 certificate.

ECC Certificate authority certificates are always given the ability to sign certificates and CRLs.

Usage & Invocation (3 of 5)

- When an ECC certificate is requested the ECC Curve Type Menu will appear (except in FIPS mode):

```
ECC Curve Type

    1 – NIST recommended curve

    2 – Brainpool standard curve

Select ECC curve type (press ENTER to return to menu):
```

- Followed by a menu of the supported curves for each type:

```
      NIST Recommended Curve Type
  1 - secp192r1
  2 - secp224r1
  3 - secp256r1
  4 - secp384r1
  5 - secp521r1


Select NIST recommended curve type (press ENTER to
return to menu):
```

```
Brainpool Standard Curve Type
   1 – brainpoolP160r1
   2 - brainpoolP192r1
   3 - brainpoolP224r1
   4 - brainpoolP256r1
   5 – brainpoolP320r1
   6 - brainpoolP384r1
   7 - brainpoolP512r1
Select Brainpool standard curve type (press ENTER to
return to menu):
```


Once the key type has been determined, the ECC name curve must be identified. ECC keys must be derived from supported ECC named curves which are limited to the listed NIST (National Institute of Standards and Technology) recommended EC name curves and Brainpool (BP) EC name curves. Brainpool is the European version of name curves.

In FIPS mode, Brainpool curves are not supported, so for a FIPS mode kdb the ECC Curve Type menu will not appear. gskkyman will go straight to the NIST Recommended Curve Type menu.

Usage & Invocation (4 of 5)

- The gskkyman –g command has been updated to allow signing of a certificate request containing an ECC key.
- A new option (-kt) allows the user to specify the key type (keyUsage extension settings) of the certificate to be created. Valid key type options are:
  - ecgen (default) - digitalSignature, nonRepudiation and keyAgreement are set
  - ecdsa - digitalSignature and nonRepudiation set
  - ecdh - keyAgreement set
- Example invocation to create an ECDSA certificate with keyAgreement usage capabilities

```
gskkyman -g -x 365 -cr eccp2.req -ct eccp2.der -k test.kdb -l ecc_ca -kt ecdh
```

- The **-kt** option is valid when signing an end user certificate containing an ECC Key. An ECC CA certificate has certificateSign and CRLSign set automatically. For other certificate types the **-kt** option is ignored.

The gskkyman command line utility provides support for various functions within gskkyman without having to go through the interactive menus. gskkyman command line was updated to allow signing of certificate request containing an ECC key.

To support the different usages of ECC certificates, option -kt was added to allow for key usage attributes to be defined depending on the usage of the certificate. For example, if the certificate is to be used during a TLS handshake for a fixed ECDH key exchange, the server certificate needs key agreement capabilities given with the ECDH option value.

Usage & Invocation (5 of 5)

- Certificate Management Services (CMS) APIs updated to allow ECDSA certificate authority and user/server certificates to be created.
- Updated CMS APIs:
  - gsk_construct_certificate
  - gsk_construct_self_signed_certificate
  - gsk_create_certification_request
  - gsk_create_database_signed_certificate
  - gsk_create_self_signed_certificate
  - gsk_create_signed_certificate
  - gsk_create_signed_certificate_record
  - gsk_create_signed_certificate_set
  - gsk_encode_ec_parameters
  - gsk_generate_key_pair
  - gsk_generate_key_parameters

Prior to the introduction of ECDSA certificates, System SSL supported the creation of certificates of RSA, DSA and DH style certificates. In z/OS V1R13 ECDSA style certificates was added.

The first set of APIs listed above were updated to allow the ECDSA certificate and key type to be specified.

The second set of APIs provides the ability to generate ECC style key pairs (private/public keys). These keys then can be passed to the creation APIs.

Interactions and dependencies

- Software Dependencies
  - ICSF Web Deliverable #10 (HCR7780) to be installed and operating on the system.
- Hardware Dependencies
  - None
- Exploiters
  - Any customers or users that may require ECC based certificates or want to perform TLS handshakes using ECC based key exchanges.

Prior to z/OS V1R12, System SSL contained software implementations for all utilized algorithms. Either through direct calls to CPACF or ICSF, System SSL is able to exploit cryptographic support provided through either the CPACF or Crypto Express cards. With the introduction of Elliptic Curve Cryptography in R12, System SSL no longer has its own software implementation. It will be relying on ICSF's PKCS#11 support for ECC.

ICSF HCR7780 is shipped in the base of z/OS V1R13 and provides ECC support. The usage of the ICSF callable services can be control through access to the CSFSERV resource class. System SSL ECC support requires read access to the CSF1DVK, CSF1GAV, CSF1GKP, CSF1PKS, CSF1PKV, CSF1TRC and CSF1TRD resources depending on the function being performed by SSL. If utilizing ECC private keys stored in the PKDS, READ may also be needed to the CSFKEYS resource class.

Migration and coexistence considerations

- None

Installation

- None

Session summary

- You should now be able to:
  - Understand the recent changes in System SSL
  - Understand the migration changes
  - Identify how to install System SSL
  - Be able to find any of the above information in the relevant publications

Appendix - References

- **Publication**
  - System Secure Sockets Layer Programming SC24-5901-10
- **Specifications**
  - RFC4492, Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) – http://tools.ietf.org/html/rfc4492
  - Standards for Efficient Cryptography Group (SECG), "SEC 1: Elliptic Curve Cryptography", http://www.secg.org/download/aid-780/sec1-v2.pdf
  - Standards for Efficient Cryptography Group (SECG), "SEC 2: Recommended Elliptic Curve Domain Parameters", http://www.secg.org/download/aid-386/sec2_final.pdf
  - RFC3279, Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile – http://www.rfc-editor.org/rfc/rfc3279.txt
  - RFC5480, Elliptic Curve Cryptography Subject Public Key Information – http://www.rfc-editor.org/rfc/rfc5480.txt
  - RFC5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation – http://www.rfc-editor.org/rfc/rfc5639.txt
  - FIPS 186-2, Digital Signature Standard (DSS) - http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf

Trademarks, disclaimer, and copyright information