# z/OS 3.1 IBM Education Assistant

Solution Name:

- Limit Elliptic Curve Cryptography (ECC) key exchange
- Optimized TLS V1.3 – Sysplex caching
- Call CSFPPS2 to offload RSA digital signature
- GSKKYMAN use stash file support

Solution Element(s): System SSL

July 2023

# Agenda

- Trademarks
- Objectives
- Solutions
  - Limit Elliptic Curve Cryptography (ECC) key exchange (ZRM-410)
  - Optimized TLS V1.3 sysplex caching (ZRM-9578)
  - Call CSFPPS2 to offload RSA digital signature (ZRM-747)
  - GSKKYMAN use stash file support (ZRM-759)
- Repeated for each Solution
  - Overview
  - Usage & Invocation
  - Interactions & Dependencies
  - Upgrade & Coexistence Considerations
  - Installation & Configuration
- Summary
- Appendix

# Trademarks

- See URL http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

# Objectives

- At the end of this presentation, you will have an overview and understand the following enhancements from System SSL:
  - Limit Elliptic Curve Cryptography (ECC) key exchange for TLS V1.0 – TLS V1.2
  - Optimized TLS V1.3 sysplex caching
  - Call CSFPPS2 to offload RSA digital signature
  - GSKKYMAN use stash file support

# Limit Elliptic Curve Cryptography (ECC) Key Exchange for TLS V1.0 – TLS V1.2

# Overview

- Who (Audience)
  - Provide System SSL applications using TLS V1.0 – TLS V1.2 with the ability to negotiate x25519/x448 elliptic curves
  - Enhances the security of TLS V1.0 - TLS V1.2 key exchanges by improving the elliptic curve selection process
- What (Solution)
  - Implement x25519/x448 support for TLS V1.0 – TLS V1.2 as specified by RFC 8422
  - Implement a new attribute that provides server applications with the ability to limit the elliptic curves used for TLS V1.0 - TLS V1.2 key exchanges
  - Support is available in z/OS 2.4 and 2.5 with APAR OA61783
- Wow (Benefit / Value, Need Addressed)
  - TLS V1.0 – TLS V1.2 negotiated connections can use x25519/x448 elliptic curves
  - Servers can ensure that stronger elliptic curves are used when available
  - Ability to query the selected key exchange elliptic curve on z/OS 3.1

# Overview

- Today, System SSL claims formal support for:
  - RFC 4492 – Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)
  - RFC 8446 – The Transport Layer Security (TLS) Protocol Version 1.3

- When System SSL added support for RFC 8446 in V2R4:
  - Support for x25519 and x448 was added for TLS V1.3 **ONLY**
  - Understanding at the time was that x25519 and x448 were ***only supported in TLS V1.3***
  - TLS V1.2 and earlier protocols were ***explicitly not*** supported

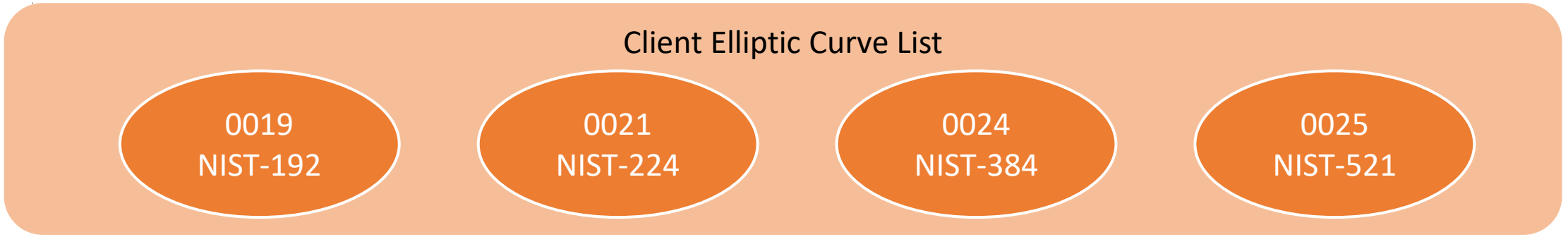
- Since then, RFC 8422 became an official standard:
  - Added support for x25519 and x448 to TLS V1.2 and earlier protocols
  - RFC 8422 obsoleted RFC 4492

# Overview

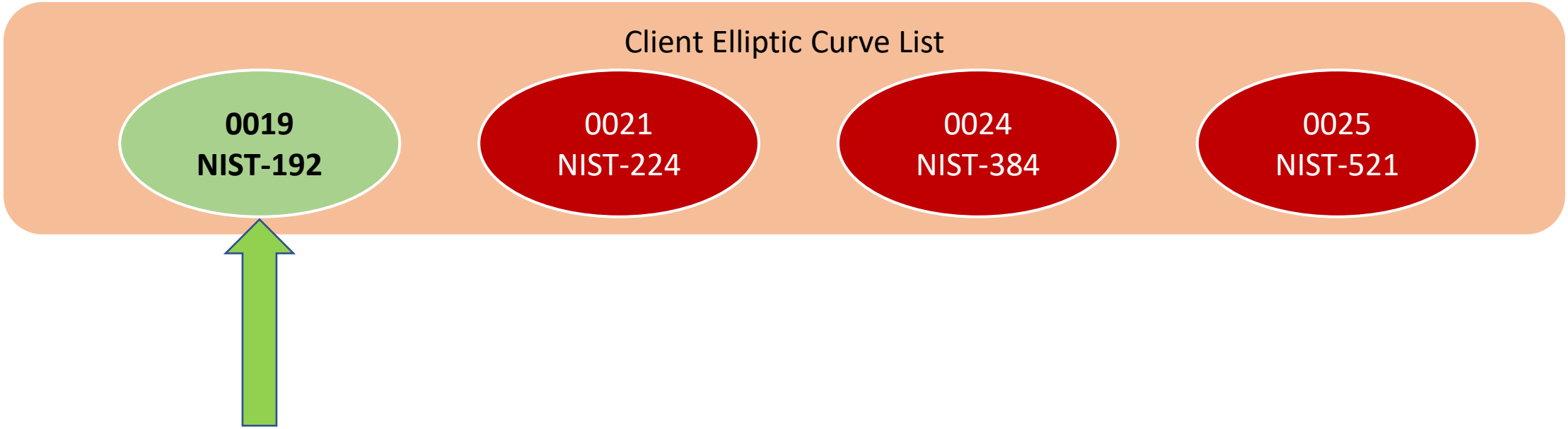
- Today, System SSL servers determine the elliptic curve to use for a key exchange in the following way:
  - Server receives the CLIENT HELLO message
  - Server processes the Supported Elliptic Curves TLS extension into a list
  - Selects the first elliptic curve found in the client's list
    - Skips unknown and unsupported (FIPS, Suite B, TLS V1.3, etc.) elliptic curves
- Once a supported elliptic curve is found, all remaining elliptic curves in the list are ignored
  - Stronger elliptic curves could be passed over in favor of weaker options earlier in the list

# Overview

## Client sent elliptic curve list




## First supported elliptic curve is selected, despite better alternatives



# Usage & Invocation

| Attribute | Description | Values |
|---|---|---|
| **GSK_SERVER_ALLOWED_KEX_ECURVES (New)** | Specifies the list of elliptic curve specifications that are allowed by the server for the TLS V1.0, TLS V1.1, and TLS V1.2 server key exchange when using ECDHE-based cipher suites as a string consisting of one or more 4-character values. | The valid values are:<br>- 0019 – secp192r1<br>- 0021 – secp224r1<br>- 0023 – secp256r1<br>- 0024 – secp384r1<br>- 0025 – secp521r1<br>- 0029 – x25519<br>- 0030 – x448<br><br>Default: 00230024002500210019<br><br>**gsk_attribute_[sg]et_buffer()** (connection or environment) |

# Usage & Invocation

- Client and Server Hello protocol updates
  - Allow x25519 and x448 to be added to and processed from the CLIENT HELLO Supported Elliptic Curves TLS Extension for TLS V1.0-TLS V1.2 handshakes
  - Updates to the processing of ECDHE ciphers
  - Updates to the selection process of a connection’s elliptic curve

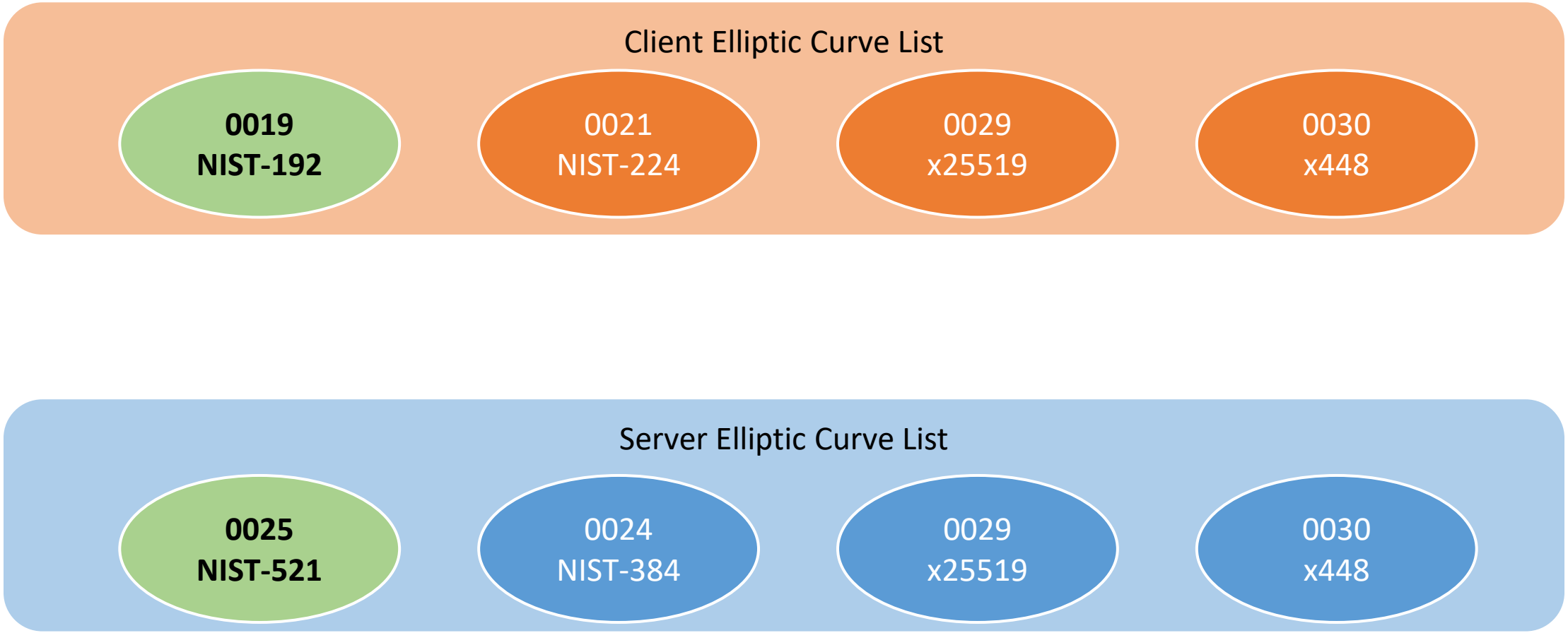
- System SSL server applications now use a server elliptic curve list in conjunction with the client’s elliptic curve list when selecting an elliptic curve for a connection
  - Server’s list is contained in the GSK_SERVER_ALLOWED_KEX_ECURVES
  - Client’s list comes from the supported elliptic curves TLS extension which is contained in the CLIENT-HELLO handshake message
  - System SSL client applications provide their supported elliptic curves thru the GSK_CLIENT_ECURVE_LIST setting

- Server determines the “preferred” elliptic curve by looking through the client's elliptic curve list for a match in the server's list

- The first matching elliptic curve is selected as the "preferred" elliptic curve

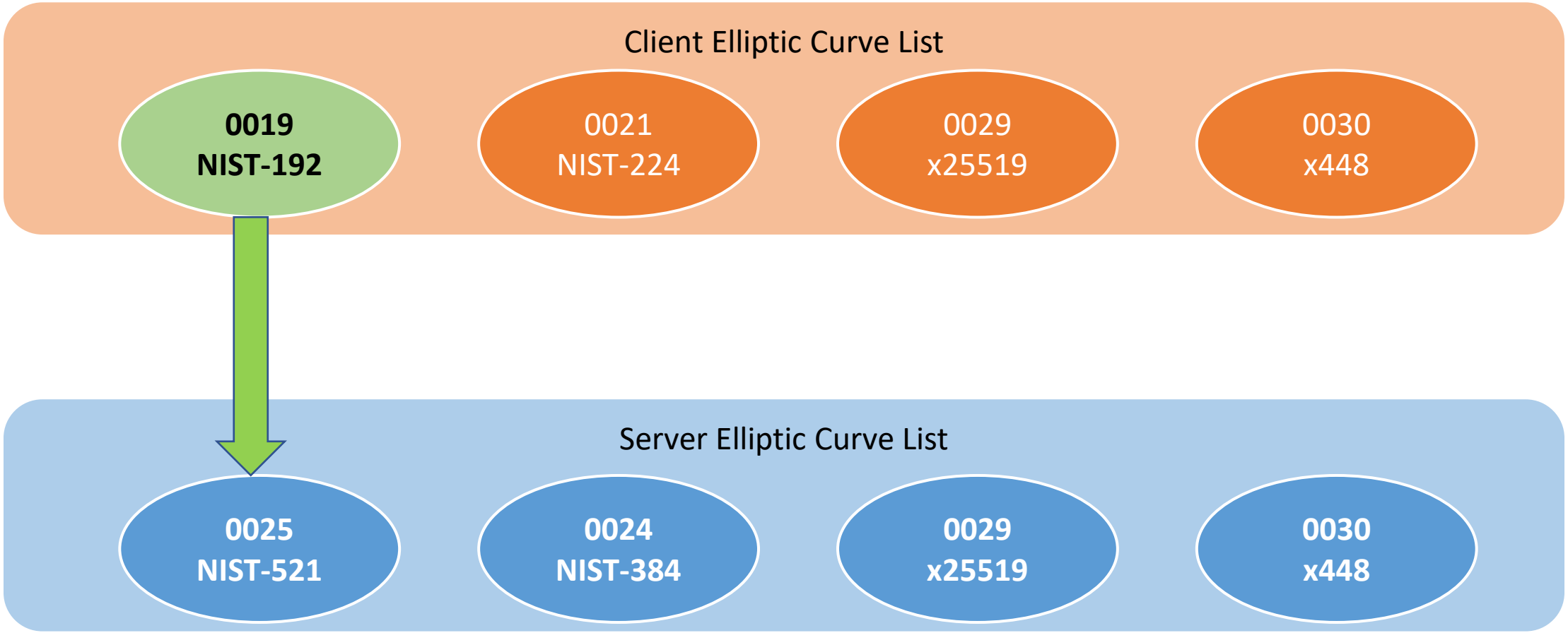
# Usage & Invocation

**Client and server have an elliptic curve list in preference order**

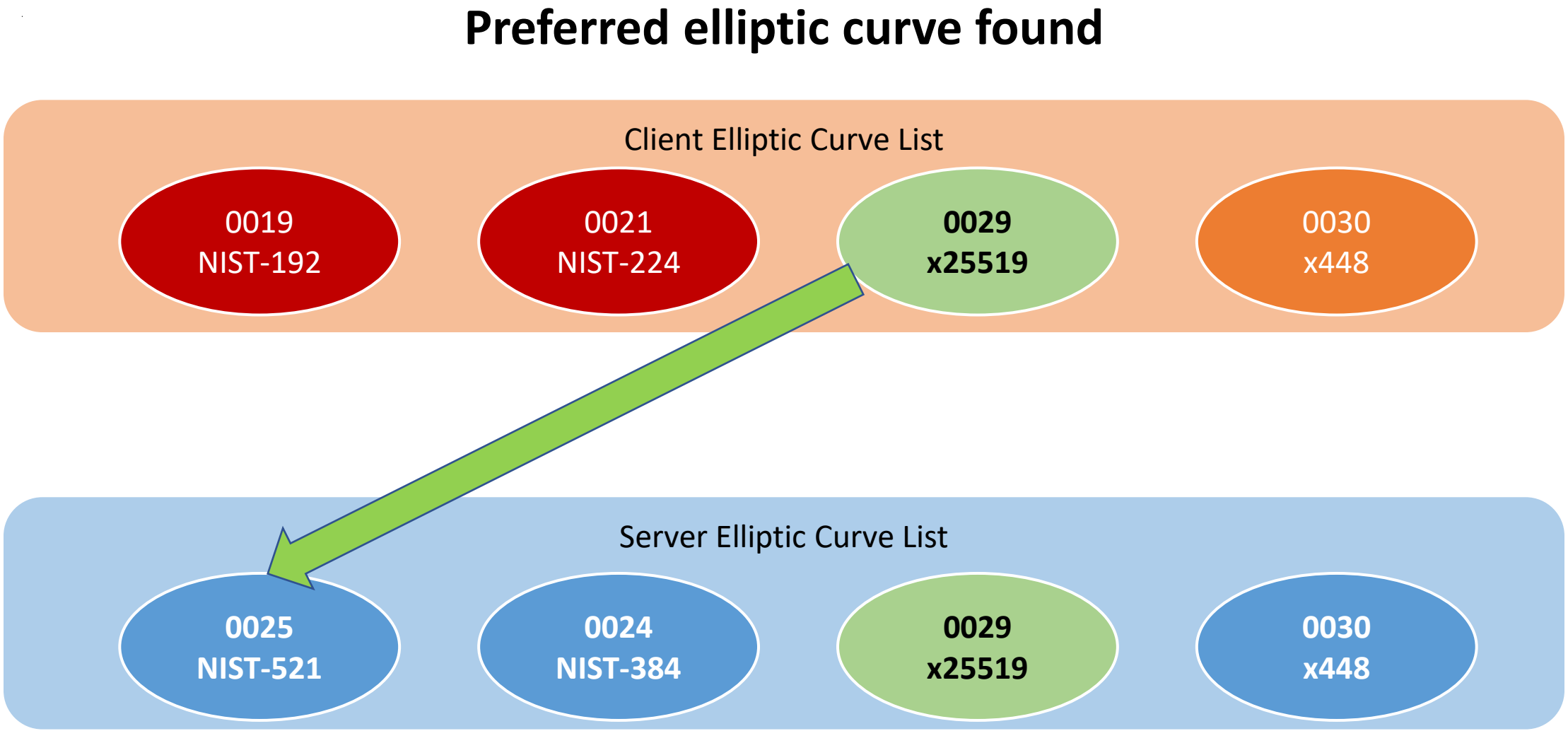


# Usage & Invocation

**Favoring the client's list, find the first common elliptic curve**



# Usage & Invocation



# Usage & Invocation

- Server and client applications on z/OS 3.1 can call the gsk_attribute_get_buffer() routine to determine the selected key exchange algorithm by specifying the GSK_CONNECT_KEX_ECURVE attribute

# Interactions & Dependencies

- Software Dependencies
  - Support for x25519 and x448 is available in ICSF
- Hardware Dependencies
  - None
- Exploiters
  - AT-TLS
  - Any z/OS System SSL application wishing to use x25519/x448 elliptic curves for TLS V1.0 - TLS V1.2 handshakes
  - Any z/OS System SSL server application wishing to limit the elliptic curves that can be used for TLS V1.0 – TLS V1.2 handshakes

# Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level:
  - No
- Migration/Toleration/coexistence APARs/PTFs
  - All systems in the sysplex must be at 2.4 or higher with PTFs for OA61783 applied when enabled for sysplex session ID caching (GSK_SYSPLEX_SIDCACHE=ON)
  - The support must be available on all systems and servers must be configured the same
    - If not, a full handshake may be performed.

# Installation & Configuration

- None

# Optimized TLS V1.3 sysplex caching

# Overview

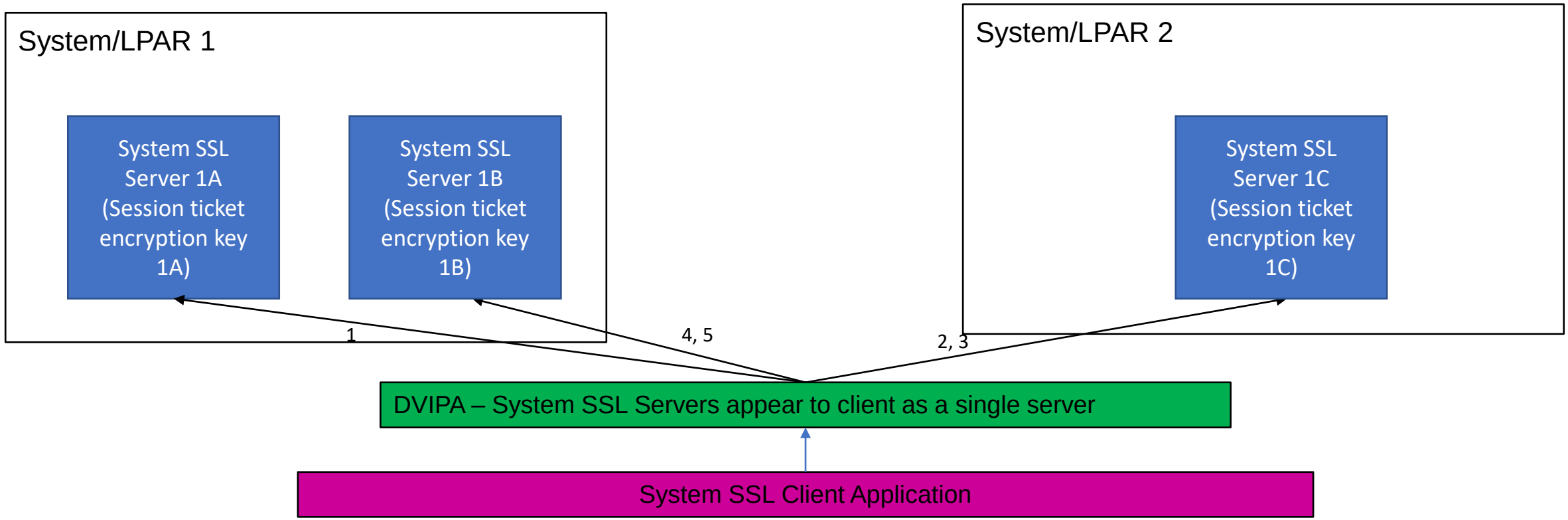
- Who (Audience)
  - System SSL like-server applications that negotiate TLS V1.3 sessions are not able to successfully resume prior established sessions on other like-server applications
  - Need the ability to monitor how the existing SID (session ID) and the new TLS V1.3 session ticket caches in the System SSL started task, GSKSRVR, are performing
  - System SSL client applications need an option to specify the maximum number of TLS V1.3 session tickets that can be stored per session
- What (Solution)
  - Add TLS V1.3 session ticket caching support to the System SSL started task, GSKSRVR. This gives the ability to allow like-server applications the capability to resume TLS V1.3 sessions created by other like-server applications
  - Add sysplex caching statistics to the GSKSRVR for the new TLS V1.3 session ticket cache and the existing SID cache
  - Add an external setting to allow client applications the ability to configure the number of session tickets stored per TLS V1.3 session
- Wow (Benefit / Value, Need Addressed)
  - By caching TLS V1.3 session ticket information, it allows for like-server applications the ability to resume TLS V1.3 sessions created by other like-server applications
  - Ability to monitor the performance of the existing SID cache and the new TLS V1.3 session ticket cache in the GSKSRVR
  - Client applications need the ability to configure the maximum number of TLS V1.3 session tickets that can be stored per session in its cache

# Overview

- TLS V1.2 and prior protocols
  - The client and server maintain full session state in their respective caches
  - Server includes a session ID in the SERVER-HELLO handshake message
  - When client needs to re-establish a secure connection, it includes the session ID in its CLIENT-HELLO handshake message
  - Server parses the CLIENT-HELLO handshake message for the session ID and looks in its cache for the session.
    - If found, an abbreviated or cached handshake is performed.

- TLS V1.3
  - After a TLS V1.3 handshake completes, the server may send 1 or more session tickets which are encrypted by an internal encryption key
  - Session tickets contain full session information and are not cached by the server.  Session IDs are not used. Only the client caches the received session tickets
  - When client needs to re-establish a secure connection, it includes a ticket in its CLIENT-HELLO handshake messages
  - Server parses the CLIENT-HELLO handshake message for the ticket and attempts to decrypt it
    - If ticket decrypts successfully and is still valid, a resumed handshake is performed.

# Usage & Invocation – Current support




1. Client application establishes a TLS V1.3 session with Server 1A. Server 1A sends session tickets encrypted with encryption key 1A. Session tickets get stored in the client cache.
2. Client application attempts to resume the previously established TLS V1.3 session by including one of the session tickets from Server 1A. This time the request gets redirected via DVIPA to Server 1C on LPAR 2.
3. Server 1C parses the incoming ticket and tries to decrypt it with encryption key 1C. The decryption fails so a full TLS V1.3 handshake must be done. Server 1C sends session tickets encrypted with encryption key 1C.
4. Client application attempts again to resume previously established TLS V1.3 session by including one of the session tickets from Server 1C. This time the request gets redirected via DVIPA to Server 1B on LPAR.
5. Server 1B parses the incoming ticket and tries to decrypt it with encryption key 1B. The decryption fails so a full TLS V1.3 handshake must be done. Server 1B sends session tickets encrypted with encryption key 1B.

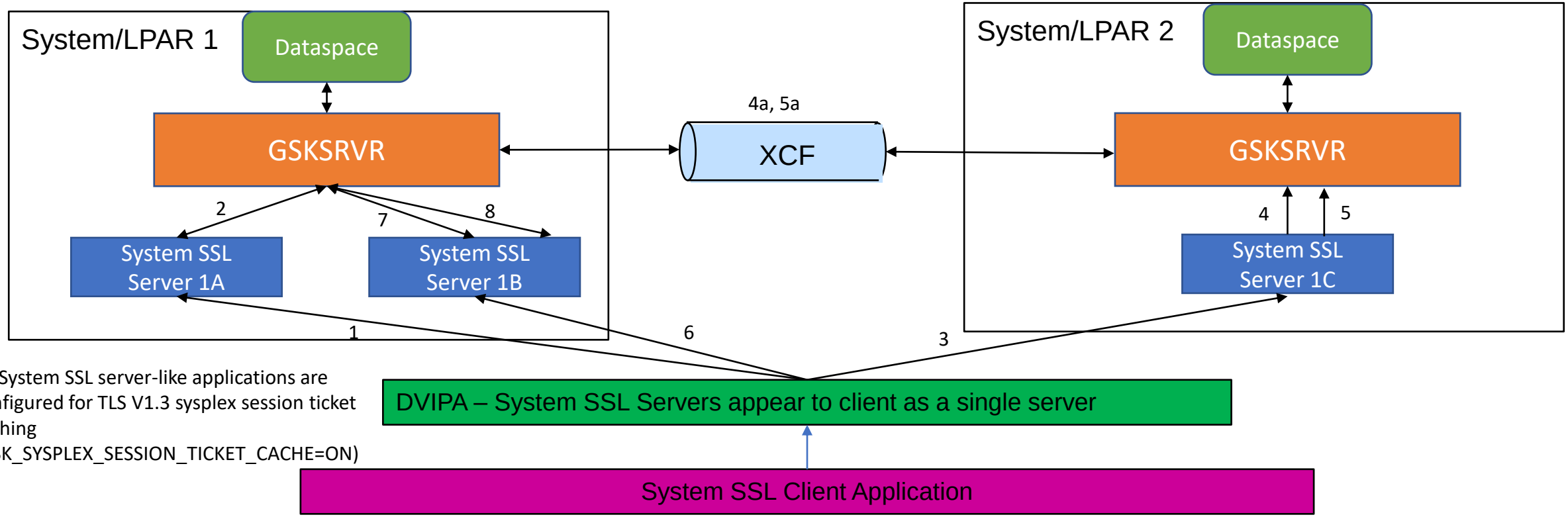
# Usage & Invocation - Settings

| Attribute | Description | Values |
|---|---|---|
| **GSK_SYSPLEX_SESSION_TICKET_CACHE (New)** | Specifies if sysplex session ticket caching for TLS V1.3 sessions are enabled for this server application | Environment variable allowed settings:<br>- OFF, 0, or DISABLED – Sysplex session ticket caching for TLS V1.3 sessions are not enabled for this server application<br>- ON, 1, or ENABLED – Sysplex session ticket caching for TLS V1.3 sessions are enabled for this server application<br><br>Default: OFF<br><br>**gsk_attribute_[sg]et_enum()** (environment):<br>• **GSK_SYSPLEX_SESSION_TICKET_CACHE_ON**<br>• **GSK_SYSPLEX_SESSION_TICKET_CACHE_OFF** |
| **GSK_SESSION_TICKET_CLIENT_MAXCACHED (New)** | Specifies the maximum number of session tickets that are allowed to be cached by the client for each TLS V1.3 session | The valid sizes are 1 through 128<br><br>Default: 8<br><br>**gsk_attribute_[sg]et_numeric_value()** (environment) |

# Usage & Invocation - Settings

| Attribute | Description | Values |
|---|---|---|
| **GSK_SESSION_TICKET_SERVER_TIMEOUT (Updated description/default)** | Specifies the maximum time that a server accepts a TLS V1.3 session resumption request from the client measured in seconds from the initial handshake. | Valid values are 1 through 604800 seconds (seven days)<br><br>If sysplex session ticket caching is not enabled (GSK_SYSPLEX_SESSION_TICKET_CACHE is set to OFF), the default session ticket timeout value is 300.<br><br>If sysplex session ticket cache is enabled (GSK_SYSPLEX_SESSION_TICKET_CACHE is set to ON), the default session ticket timeout value is 600.<br><br>**gsk_attribute_[sg]et_numeric_value()** (environment) |

# Usage & Invocation – New Support




1. Client application establishes a TLS V1.3 session with Server 1A.
2. Session gets cached in the GSKSRVR in System 1 and session tickets get sent to client from Server 1A.
3. Client application attempts to resume previously established TLS V1.3 session with a received ticket. However, this time gets redirected via DVIPA to Server 1C on LPAR 2.
4. Server 1C communicates with the local GSKSRVR instance on LPAR 2 to retrieve session. The GSKSRVR realizes that it does not have the cached session so the request gets sent over the XCF (4a) to retrieve the cached session info associated with the ticket to GSKSRVR on LPAR 1.
5. After successfully resuming the TLS V1.3 session, a new session ticket needs to be sent. The GSKSRVR instance on LPAR 2 needs to do a session update prior to sending a new session ticket which gets cached.
6. Client application attempts again to resume previously established TLS V1.3 session. It gets redirected to Server 1B by the DVIPA.
7. Server 1B retrieves the session from the local GSKSRVR instance and resumes the TLS V1.3 session.
8. After successfully resuming the TLS V1.3 session, a new session ticket needs to be sent. The GSKSRVR instance on LPAR 1 successfully updates the session info and Server 1B sends the ticket.

# Usage & Invocation – New Support

- When enabled for sysplex ticket caching:
  - GSKSRVR will store full TLS V1.3 session information along with keeping track of each individual ticket that is sent
  - These TLS V1.3 session tickets contain a ticket identifier which allows for a quick lookup in its cache
  - Once ticket is used on a TLS V1.3 resumption, it is marked as used and cannot be re-used

# Usage & Invocation – GSKSRVR support

- GSKSRVR has been updated to add a new session ticket cache for storing the session information for TLS V1.3 session tickets
- New supported GSKSRVR environment variable settings
  - GSK_SESSION_TICKET_CACHE_SIZE – Specifies the size of the sysplex session ticket cache in megabytes and is between 1 and 512 with a default of 256. The default of 256 is used if a valid value is not specified.
  - GSK_SESSION_TICKET_CACHE_TIMEOUT – Specifies the sysplex session ticket cache entry timeout in minutes and is between 1 and 10080 (7 days) with a default of 10. The default of 10 is used if a valid value is not specified.
  - GSK_SESSION_TICKET_CACHE_NUM_TICKETS – Specifies the number of session tickets that are stored in the sysplex session ticket cache entry per session and is between 16 and 320 with a default of 16. If a multiple of 16 is not specified, the number of session tickets per cache entry is rounded to the nearest multiple of 16. If number specified is less than 16 or greater than 320, the default of 16 is used.
- Changed default GSKSRVR environment variable setting
  - GSK_SIDCACHE_SIZE – Specifies the size of the sysplex session cache in megabytes and is between 1 and 512. The previous default was 20 but now it is 256.

# Usage & Invocation – GSKSRVR support

- New GSKSRVR operator modify commands (MODIFY GSKSRVR,*parameters)*
  - DISPLAY STATS,SIDCACHE – Displays detailed statistics for session ID cache for TLS V1.2 and earlier sessions
  - DISPLAY STATS,TICKETCACHE – Displays detailed statistics for the session ticket cache for TLS V1.3 sessions
  - RESET STATS,SIDCACHE – Displays and then resets the detailed statistics for the session ID cache for TLS V1.2 and earlier sessions
  - RESET STATS,TICKETCACHE – Displays and then resets the detailed statistics for the session ticket cache for TLS V1.3 sessions
  - DISPLAY TICKETCACHE – Displays the current number of cache entries by application user id and the maximum cache entry size in bytes

- Changed GSKSRVR operator command output:
  - DISPLAY SIDCACHE – Updated to display the maximum cache entry size in bytes

# Usage & Invocation – Example command output

**F GSKSRVR,DISPLAY STATS,TICKETCACHE**

```
GSK01079I Session ticket cache
 statistics

Current time:        2023/02/16 10:46:56 EST
Start time:          2023/02/16 10:34:50 EST
Reset time:          2023/02/16 10:34:50 EST
Number of resets:    0

Current cache size:  13M
Maximum cache size:  13M
Cache timeout:       1
Cache num tickets:   16

Add statistics                          Count
--------------                          -----
Requested:                               6020
Successful:                              5250
Failed full dataspace:                    770
Success percentage:                    87.209%
Replaced entries:                        1924

Update statistics                       Count
-----------------                       -----
Requested:                              22631
Successful:                             22631
Failed expired:                             0
Failed not found:                           0
Failed not authorized:                      0
Success percentage:                   100.000%

Retrieval statistics                    Count
--------------------                    -----
Requested:                              22643
Successful:                             22631
Failed expired:                             4
Failed not found:                           8
Failed not authorized:                      0
Success percentage:                    99.947%
```

**F GSKSRVR,DISPLAY TICKETCACHE**

```
GSK01078I Session ticket cache
 status
User              Count              Max size
*DSPACE*          13/13                   N/A
USER1              3326                  3261
```

# Usage & Invocation – Example command output

```
F GSKSRVR,DISPLAY STATS,SIDCACHE
GSK01081I Session cache
 statistics

Current time:        2023/02/16 13:19:16 EST
Start time:          2023/02/16 13:11:59 EST
Reset time:          2023/02/16 13:12:15 EST
Number of resets:    1

Current cache size:  5M
Maximum cache size:  5M
Cache timeout:       60

Add statistics                          Count
--------------                          -----
Requested:                               7028
Successful:                              5112
Failed full dataspace:                   1916
Success percentage:                    72.738%
Replaced entries:                        2556

Retrieval statistics                    Count
--------------------                    -----
Requested:                              55048
Successful:                             55028
Failed expired:                             4
Failed not found:                          16
Failed not authorized:                      0
Success percentage:                    99.964%
```

```
F GSKSRVR,DISPLAY SIDCACHE
GSK01032I Session cache status
User              Count              Max size
*DSPACE*            5/5                   N/A
USER1              2556                  1120
```

# Usage & Invocation

- System SSL client applications were previously limited to a maximum of 8 session tickets per cached TLS V1.3 session

- Support has been added to the client cache to store anywhere between 1 and 128 tickets per cached TLS V1.3 session
  - New GSK_SESSION_TICKET_CLIENT_MAXCACHED setting allows this to be configurable for an SSL environment within a client application

# Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - AT-TLS (Communication Server)

# Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level:  No
- Migration/Toleration/coexistence APARs/PTFs
  - None

# Installation & Configuration

- None

# Call CSFPPS2 to offload RSA digital signature

# Overview

- Who (Audience)
  - Currently when System SSL is executing in FIPS mode, RSA digital signature generation are always performed within System SSL software
  - This works but is less efficient than if System SSL could call ICSF to handle the processing in hardware
- What (Solution)
  - Add support for offloading clear key RSA digital signature processing to the accelerator when System SSL is running in FIPS mode, and the accelerator is available for use
- Wow (Benefit / Value, Need Addressed)
  - Improve component performance by running on the accelerator card rather than the System SSL software running on the CPU
  - Updates the RSA digital signature processing to be in line with how the other three operations work (verification, encryption, decryption) for consistency in overall support

# Usage & Invocation

- The digital signature processing will be automatically be done if an CEXxA optional cryptographic card is available for hardware offloading

# Interactions & Dependencies

- Software Dependencies
  - ICSF PKCS #11 processing through callable services: CSFPPS2 (31-bit) and CSFPPS26 (64-bit).
  - Need access to the CSFDSG resource profile in the CSFSERV class
- Hardware Dependencies
  - CEXxA – Optional cryptographic card for hardware offloading
    - New hardware path will only be utilized if an accelerator card is available
- Exploiters
  - AT-TLS (Communications Server)
  - Any System SSL application that runs in FIPS mode
  - Any users of **gskkyman** running in FIPS mode

# Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level
  - No

- Migration/Toleration/Coexistence APARs/PTFs
  - None

# Installation & Configuration

- None

# GSKKYMAN use stash file support instead of requiring password entry

# Overview

- Who (Audience)
  - Users of the z/OS System SSL **gskkyman** certificate management utility are finding difficulty in automating the time-consuming process of certificate management
  - The command line mode for **gskkyman** requires a manual password entry for most of its functionality which makes automating certificate management difficult for system administrators
- What (Solution)
  - System SSL will add command line options to support stash files, key database passwords, and PKCS#12 passwords
- Wow (Benefit / Value, Need Addressed)
  - These new command line options will allow for key database files and PKCS#12 files to be accessed without prompting for the password which allows for automation

# Usage & Invocation

- The following gskkyman command line functions have been updated (changes are highlighted in blue)
  - View certificate:
    - gskkyman -dc|-dcv [-k filename|-t tokenname|-p12 filename|-der filename] [-l label] [-kpw password|-sth] [-p12pw password]
    - -kpw or -sth can be used to open the KDB specified by the -k option.
    - -p12pw can be used to open the PKCS #12 key store specified by the -p12 option.
  - View KDB expiration:
    - gskkyman -dk [-k filename] [-kpw password|-sth]
    - kpw or -sth can be used to open the KDB specified by the -k option.
  - Import/export a certificate:
    - gskkyman -e|-i [-k filename|-t tokenname] [-l label] [-p filename] [-kpw password|-sth] [-p12pw password]
    - -kpw or -sth can be used to open the KDB specified by the -k option.
    - -p12pw can be used as the certificate password for the PKCS #12 certificate specified by the -p option.

# Usage & Invocation

- Command line updates continued (changes highlighted in blue):
  - Create a certificate request:
    - gskkyman -g [-x days] [-cr filename] [-ct filename] [-k filename|-t tokenname] [-l label] [-kt {ecgen|ecdsa|ecdh}] [-ca] [-ic] [-pss] [-kpw password|-sth]
    - -kpw or -sth options can be used to open the KDB specified by the -k option.
  - Create a KDB stash file:
    - gskkyman -s [-k filename] [-kpw password]
    - -kpw can be used to open the KDB specified by the -k option.
  - Help Command:
    - gskkyman –h|-?
    - Updated to show new options:
      - -kpw Key database file password.
      - -p12pw PKCS #12 file password.
      - -sth Key database stash file.

# Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - System SSL users that require certificate management through **gskkyman**

# Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level
  - No

- Migration/Toleration/Coexistence APARs/PTFs
  - None

# Installation & Configuration

- None

# Summary

- You should now be able to understand the following enhancements from System SSL:
  - Limit Elliptic Curve Cryptography (ECC) key exchange for TLS V1.0 – TLS V1.2
  - Optimized TLS V1.3 sysplex caching
  - Call CSFPPS2 to offload RSA digital signature
  - GSKKYMAN use stash file support

# Appendix

- z/OS Cryptographic Services System SSL Programming