z/TPF V1.1

TPF Users Group - Fall 2009

z/TPF Security Enhancements

Name: Mark Gambino
Venue: Communications Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

First a quick refresher on Secure Key Management support in z/TPF that was announced at previous TPF Users Group Conferences

Secure Symmetric Key Management Support

- **z/TPF APAR PJ31450 (May 2007)**
- **Enables you to create and manage symmetric encryption keys in a secure manner**
  - DES, TDES, and AES-128
  - APAR PJ32630 (April 2008) added AES-256 support
- **Applications can use the support to protect sensitive data stored on tape or disk (data at rest) or flowing over the network (data in flight)**
- **High performance designed for mainline application use**
- **Access controls to limit and log key usage**
- **Can help you meet the ever growing list of security and compliance standards**

PKI Support Phase 1 Summary

- APAR PJ32686 (December 2008)
- Create and manage RSA public key pairs in a secure manner on z/TPF
  - Create, activate, deactivate, display, delete RSA public key pairs
  - Backup and restore the PKI keystore
- Use the RSA keys generated on z/TPF to create digital certificate requests as well as self-signed digital certificates
- Enable z/TPF SSL applications and middleware to use private keys generated by z/TPF
- Ability to extract a public key from the z/TPF keystore that can be distributed to remote partners such as key managers
- Ability to import a symmetric key in a secure manner using public key cryptography

Certificate Management Enhancements

APAR PJ35513

Displaying Certificate Contents

- **ZPUBK DISPCERT command**
- **New operator command to display contents of a digital certificate in human-readable format**

Sample Certificate Display – Part 1 of 3

```
ZPUBK DISPCERT PATH-/mycerts/servcert.pem
PUBK0025I 11.44.06 CERTIFICATE DISPLAY
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 0 (0x0)
      Signature Algorithm: sha1WithRSAEncryption _
      Issuer: C=US, ST=NY, L=POK, O=IBM, OU=TPF, CN=TEST_CA
      Validity
         Not Before: Feb 11 16:06:31 2009 GMT
         Not After : Aug 12 16:06:31 2009 GMT
      Subject: C=US, ST=NY, L=POK, O=IBM, OU=TPF, CN=SERVER_PGM1/emailAddress=me@home
```

Sample Certificate Display – Part 2 of 3

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:bc:43:20:94:10:1c:70:5b:ae:cf:e2:af:4f:6a:
99:8e:e3:8b:be:43:a1:34:0d:38:55:97:4e:26:8d:
98:a7:4f:dc:16:b6:05:b2:6a:4d:eb:b4:8c:cf:c4:
13:04:93:f2:eb:ca:95:db:a9:21:3f:17:5e:24:4d:
86:62:85:d4:56:fe:16:aa:69:96:ea:76:67:d8:de:
83:f9:5b:c1:de:4f:64:bd:6c:c9:31:51:a2:b9:36:
40:04:c2:4d:9c:8b:8c:22:4c:51:0a:c1:4f:29:f5:
59:3f:ea:b4:67:22:8c:b4:0c:bf:3b:f2:39:69:96:
79:02:06:9b:1d:db:43:90:91

Exponent: 3 (0x3)

Sample Certificate Display – Part 3 of 3

Signature Algorithm: sha1WithRSAEncryption

9b:15:98:d1:6d:c5:b1:c8:59:4a:1f:dd:ce:b9:74:3f:d4:58:
4e:cb:14:c0:1a:17:c3:c3:c9:52:3a:f1:67:8c:c0:a4:fb:50:
33:a5:c6:8a:38:27:25:7d:0c:b4:41:f3:fb:70:9f:b6:69:3a:
90:0a:74:47:54:5b:65:64:d7:a1:e7:69:6f:a6:4f:20:65:4a:
87:68:91:31:3e:3f:04:82:72:b9:d8:f7:bf:92:57:c1:9e:c1:
d5:dd:5f:9f:b0:9c:d8:5c:51:1b:eb:97:b6:ae:8b:e2:ef:df:
bd:5b:fb:42:a4:51:09:85:28:99:54:6f:5d:07:47:ff:5d:8d:
b8:d3

END OF DISPLAY

Converting Certificate from PEM to DER

- **ZPUBK CONVCERT command**
- **Converts a certificate from Base64 Privacy Enhanced Mail (PEM) format to Distinguished Encoding Rules (DER) format.**
- **Some remote key managers only support DER format**

RSA APIs for Applications

APAR PJ35513

Encrypt User Data using RSA Public Key Cryptography

- **tpf_RSA_encrypt_data API**
- **Input:**
  - Pointer to and length of user data to be encrypted
  - Pointer to buffer into which to place the encrypted data
  - The RSA public key to use to encrypt the data
  - RSA padding method to use
- **Output:**
  - Encrypted data in user specified buffer
  - Length of encrypted data (including pad bytes)
    - The value is always the length of the RSA key

Different Ways to Specify which RSA Public Key to Use*

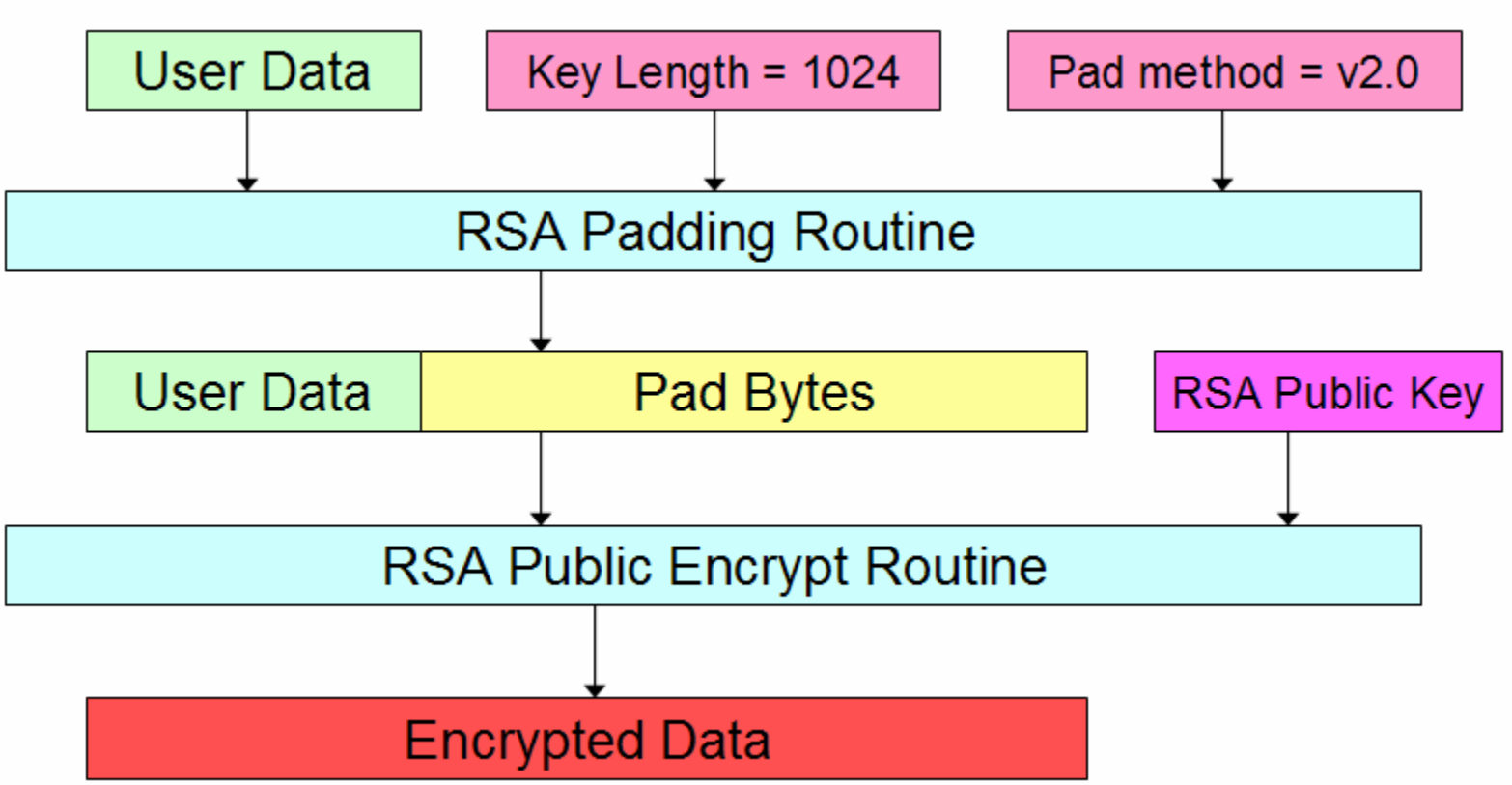
- **RSA key pair name**
  - z/TPF generated RSA key pairs only
- **Name of file containing certificate**
  - Public key is part of a certificate
- **Name of file containing public key**
- **Pointer to an RSA memory structure**
  - Public key portions of the structure are required to be filled in
  - Private key fields can be blank (NULL) in the structure.
- **Pointer to X509 memory structure**
  - This structure contains a certificate in memory

*This information applies to all RSA APIs where a public key is input

RSA Padding

- **Input to RSA encryption requires N number of bytes where N is the size of the RSA key**
  - For example, if RSA key size is 1024 bits, N = 128
- **Amount of user data to encrypt is typically much less than N; therefore, must add pad bytes to:**
  - Make it more difficult to crack the plain text
  - Enables the RSA decryption routine to detect tampered data
- **Padding methods are defined by Public Key Cryptography Standards (PKCS)**
- **z/TPF RSA APIs support:**
  - PKCS #1 v1.5 padding
  - PKCS #1 v2.0 padding
    - **Optimal Asymmetric Encryption Padding (OAEP)**

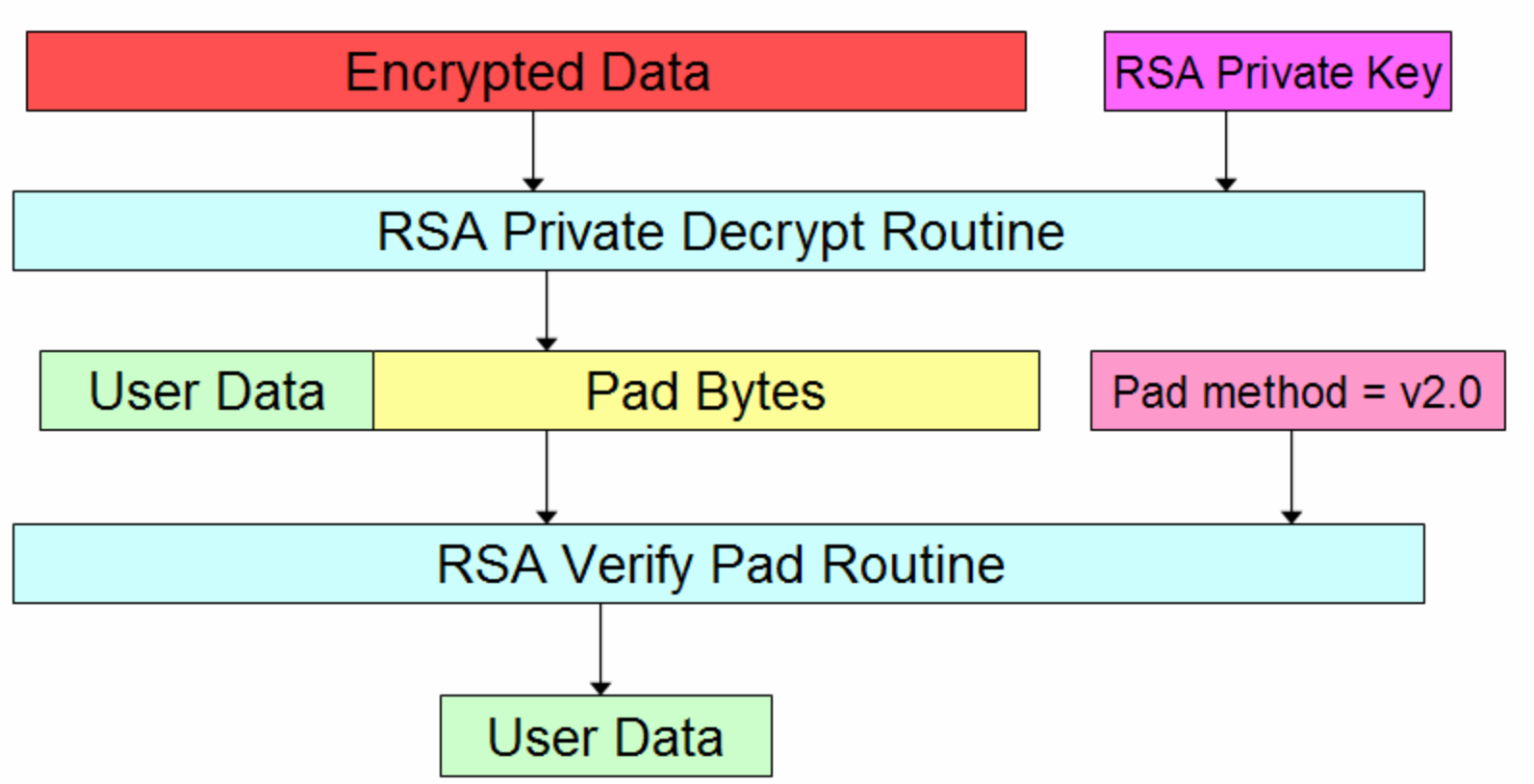
RSA Public Key Encryption Overview



Decrypt User Data using RSA Public Key Cryptography

- **tpf_RSA_decrypt_data API**
- **Input:**
  - Pointer to the encrypted data (length is the RSA key size)
  - Pointer to buffer into which to place the decrypted data
  - The RSA private key to use to decrypt the data
    - Name of the z/TPF generated RSA key pair
  - RSA padding method to use
- **Output:**
  - Decrypted data in user specified buffer
  - Length of decrypted data (excluding pad bytes)

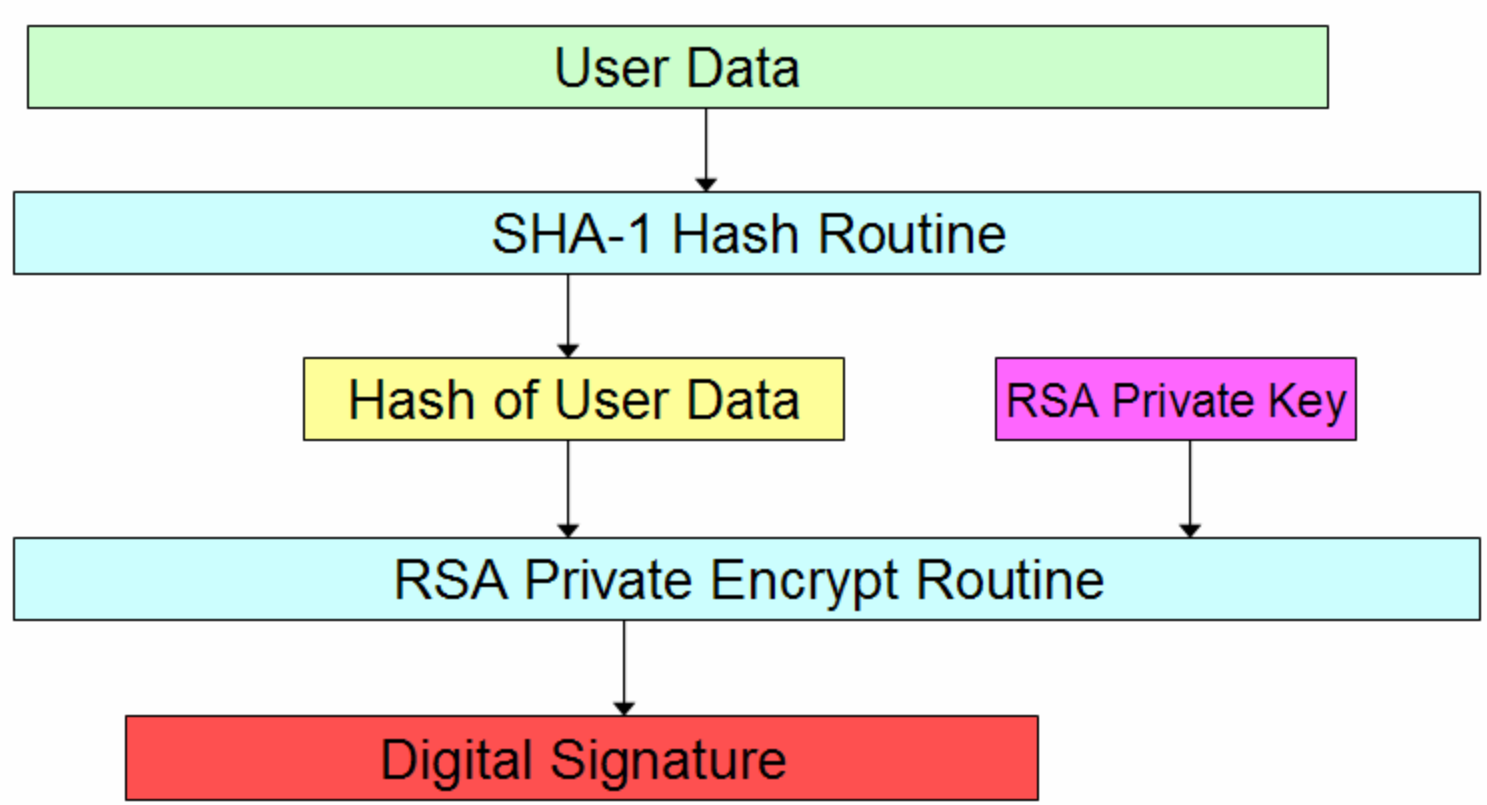
RSA Private Key Decryption Overview



Digitally Sign User Data using RSA Public Key Cryptography

- **tpf_RSA_sign_data API**
- **Input:**
  - Pointer to and length of user data to be signed
  - Pointer to buffer into which to place the signature
  - The RSA private key to use to sign the data
    - Name of the z/TPF generated RSA key pair
  - Hash algorithm to use is SHA-1
- **Output:**
  - Digital signature in user specified buffer
  - Length of signature
    - The value is always the length of the RSA key

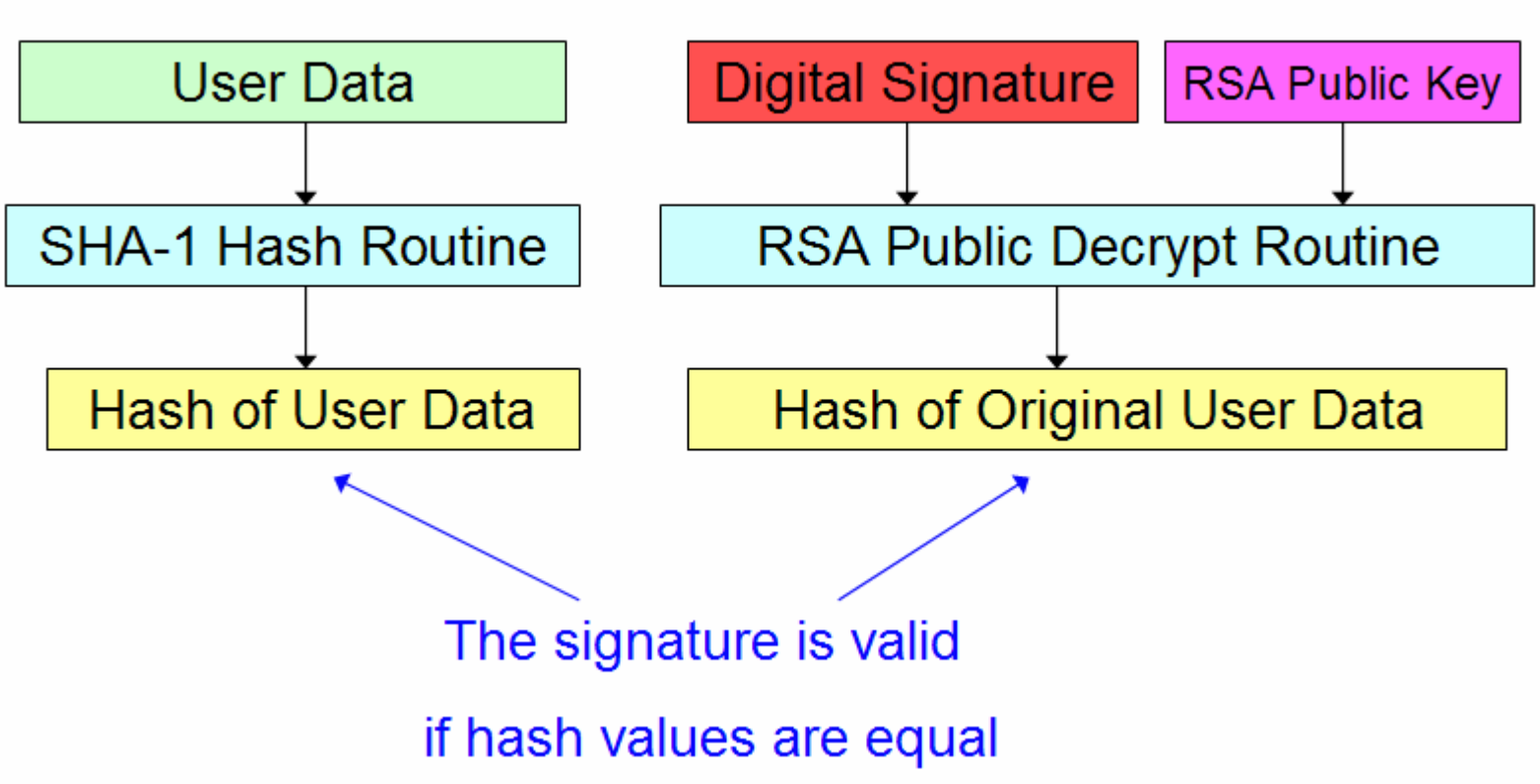
Creating RSA Digital Signature Overview



Verify RSA Digital Signature of User Data

- **tpf_RSA_verify_data API**
- **Input:**
  - Pointer to and length of user data
  - Pointer to the signature to be verified
  - The RSA public key to use to verify the signature
  - Hash algorithm to use is SHA-1
- **Output:**
  - Return code indicating whether the digital signature is valid

Verifying RSA Digital Signature Overview



Other RSA Digital Signature APIs

- **tpf_RSA_sign_digest API**
  - Same as the tpf_RSA_sign_data API except that the input includes the digest of the user data rather than the user data itself
- **tpf_RSA_verify_digest API**
  - Same as the tpf_RSA_verify_data API except that the input includes the digest of the user data rather than the user data itself

RSA Statistical Information

- **ZPUBK DISPLAY STATS SUMMARY command**
  - Displays statistical information about all RSA public and private key operations
    - z/TPF generated RSA keys or remote generated RSA keys
- **ZPUBK DISPLAY STATS PRIV command**
  - Displays statistical information about RSA private key operations for one or all key pairs in the z/TPF keystore
    - z/TPF generated RSA private keys
- **Statistical information includes RSA usage by applications (RSA user APIs) and by the SSL code**

ZPUBK DISPLAY STATS SUMMARY Example

	CURRENT	MAX
RSA1024 DIGITAL SIGNATURE VERIFIES/SECOND	585	1 051
RSA1024 PUBLIC ENCRYPTS/SECOND	126	200
RSA1024 PUBLIC SSL OPERATIONS/SECOND	3	43
RSA1024 DIGITAL SIGNATURE CREATES/SECOND	422	624
RSA1024 PRIVATE DECRYPTS/SECOND	125	200
RSA1024 PRIVATE SSL OPERATIONS/SECOND	1	15
RSA2048 DIGITAL SIGNATURE VERIFIES/SECOND	257	613
RSA2048 PUBLIC ENCRYPTS/SECOND	81	103
RSA2048 PUBLIC SSL OPERATIONS/SECOND	0	6
RSA2048 DIGITAL SIGNATURE CREATES/SECOND	81	87
RSA2048 PRIVATE DECRYPTS/SECOND	81	103
RSA2048 PRIVATE SSL OPERATIONS/SECOND	0	4

ZPUBK DISPLAY STATS PRIV ALL

```
PUBK0026I 10.11.40 ZPUBK DISPLAY STATS PRIV PROCESSING STARTED
```

KEYPAIR	CIPHER	APPL SIG/SEC	MAX SIG/SEC	APPL DEC/SEC	MAX DEC/SEC	SSL OPS/SEC	MAX SSL OPS/SEC
KEY006	RSA2048	80	87	0	0	0	1
KEY005	RSA2048	0	1	81	103	0	4
KEY004	RSA1024	0	0	123	200	0	1
KEY003	RSA1024	0	1	0	0	0	1
KEY002	RSA1024	307	416	0	0	1	7
KEY001	RSA1024	113	208	0	0	0	1

```
END OF DISPLAY
```

Shared SSL Enhancements

APAR PJ36179

Shared Secure Sockets Layer (SSL) Support

- **Allows an SSL session to be shared across ECBs**
- **Provides asynchronous I/O**
  - Activate_on_receipt (AOR) capability
- **APIs are processed by a shared SSL daemon process**
- **Multiple shared SSL daemon processes can be defined**

Shared Secure Sockets Layer (SSL) Support

- **Shared SSL daemon processes are defined in CTK2**
  - SSLPROC – number of shared SSL daemon processes
  - SSLTHRD – number of threads in each shared SSL daemon process
- **Before APAR PJ36179**
  - Changing SSLPROC required an IPL (load a new CTK2)
- **With APAR PJ36179**
  - SSLPROC can be changed via ZNKEY command
  - No IPL required – just stop and restart (or recycle) the SSL daemons via the ZSSLD commands

Memory Usage by Shared SSL Daemon Processes

- **Each ECB (or process if multi-threaded) has a limit on the amount of 31-bit ECB heap it can use**
  - EMPS value in CTKA defines the limit
- **As the number of active sessions increases within a shared SSL daemon process, the amount of ECB heap in use in that process also increases**
- **APAR PJ36179 adds monitor capability to SSL daemon processes to track ECB heap usage**
  - Included in ZSSLD DISPLAY
  - Console warning messages also issued if a shared SSL daemon process is nearing the ECB heap limit

Sample ZSSLD DISPLAY

```
SSLD0007I 10.11.48 CSSLZD - SSL STATISTICAL INFORMATION

                    LAST MINUTE  HIGH WATER MARK
                    -----------  --------------
  SESSIONS STARTED           33             194
  SSL_WRITES ISSUED       26941           26941
  SSL_READS ISSUED        34316           34316
  MEGABYTES SENT         121.41          154.20
  MEGABYTES RECEIVED     115.88          140.06

  SSL     ACTIVE    MAX ACT   CURRENT    MAX       HEAP      AVAIL   MAX HEAP
  DAEMON  THREADS   THREADS    SESS      SESS     IN USE     HEAP      IN USE
  ------  -------   -------   -------    -----  --------  --------  --------
     1       31        32        36        46        9.07     41.01     10.42
     2       31        32        36        46        8.51     41.57      9.43
     3       32        32        39        46        9.39     40.68     10.41
     4       32        32        39        46        8.63     41.44      9.42
     5       30        32        35        45        9.16     40.92     10.37
     6       28        32        35        45        8.55     41.53      9.43
END OF DISPLAY
```

Shared SSL Daemon Recovery

- **If an SSL daemon process fails for any reason (like hitting the ECB heap limit):**
  - Before APAR PJ36179
    - Can result in hung application ECBs that are waiting for shared SSL APIs to be completed
  - With APAR PJ36179
    - Monitor will detect the failure and issue console message indicating which shared SSL daemon process failed
    - ECBs that issued APIs that were being processed by the failed daemon are posted and notified of the failure
    - All shared SSL system tables/resources are cleaned up
    - Can use ZSSLD RECYCLE command to restart the SSL daemon processes (including the failed process)

Summary

- **Ability to display certificate contents in readable format**
- **Ability to convert a certificate from PEM to DER format**
- **APIs to encrypt and decrypt user data using RSA**
- **APIs to create and verify RSA digital signatures**
- **Ability to increase number of shared SSL daemon processes without an IPL**
- **Shared SSL daemon process monitoring capabilities to help tune heap limits, and enable recovery without an IPL or hung ECBs**

Trademarks

- **IBM is trademark of International Business Machines Corporation in the United States, other countries, or both.**
- **Other company, product, or service names may be trademarks or service marks of others.**

- **Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.**
- **All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.**
- **This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.**

- **All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.**

- **Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.**

- **Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.**

- **This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.**