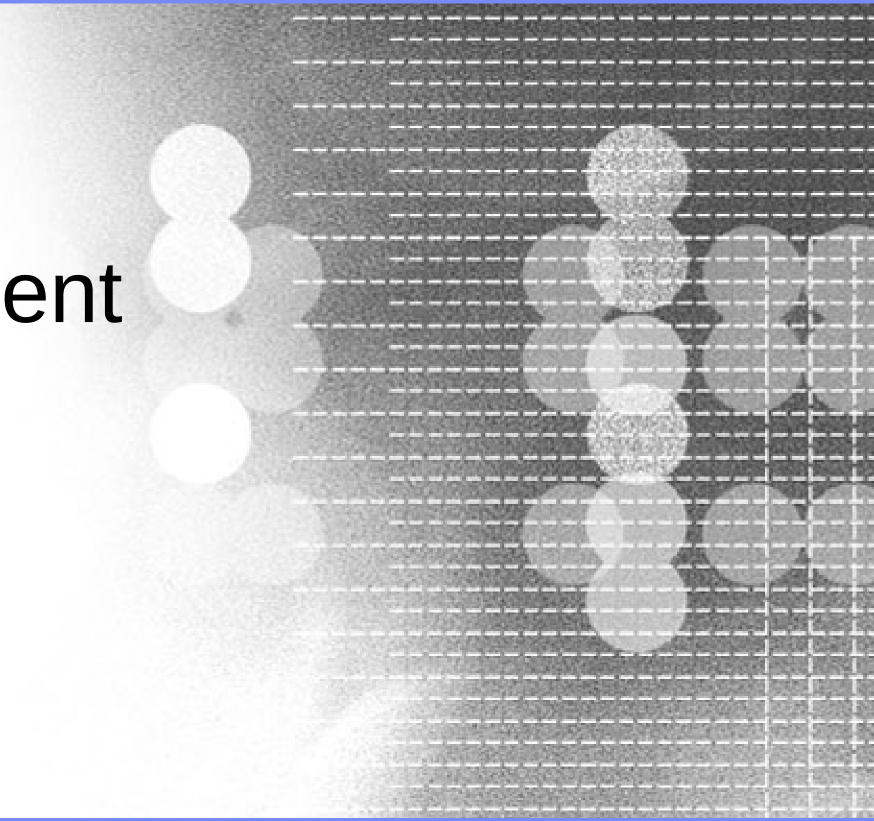
TPF Users Group Spring 2007



z/TPF Secure Key Management

Mark Gambino

Main Tent Presentation

AIM Enterprise Platform Software

IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Encryption of Data

- Increasing regulatory requirements driving need for security of data for audit and compliance
- Requirements for tighter security driving need for encryption of data
- Data compromised from a security breach can:
  - Take time and money to recover the data
  - Cause loss of revenue and customers
  - Severely damage the brand image!
- Recommendation:
  - Take proactive steps to protect your enterprise data

Understanding Casino Games

- Playing Texas Hold 'Em you hit the nut flush on the turn. The board pairs on the river... should you be worried?

- Red has come up 8 times in a row on a roulette table. Should you increase your bet and bet black on the next spin?

- Playing Blackjack, you have two 9's and the dealer's up card is also a 9... what do you do? Instead of a 9, suppose the dealer had an 8 - what do you do?

- Playing Craps, 2 is your lucky number. When should you place a "4 hardway (2&2)" bet versus a "2&2 hop" bet?

Understanding Cryptography (Yes, a Geek Page)

- Public key cryptography
  - Uses public/private key pair
  - RSA is most commonly used
  - In practice is used to encrypt small amounts of data because of the CPU overhead involved in performing private key operations
- Symmetric key cryptography
  - Same key used to encrypt and decrypt the data
  - Used for bulk data encryption, both:
    - Large amount of data
    - High number of encrypt/decrypt operations
  - Examples include DES, Triple-DES (TDES), and AES

Symmetric Key Encryption Using Hardware Cryptography

- Secure key hardware crypto cards
  - The good news... key values are stored on the card, not in host (OS) memory or database
  - The bad news... throughput is limited
- Central Processor Assist for Cryptographic Functions (CPACF)
  - Hardware cryptographic accelerator introduced on z990
  - Supports DES and TDES ciphers. AES128 support was added with z9
  - Designed for applications with high volume cryptographic needs
  - Supports clear keys only for performance reasons
    - Tens of thousands (up to a few hundred thousand) of encrypt/decrypt operations per second per CPACF (rate varies based on data size)

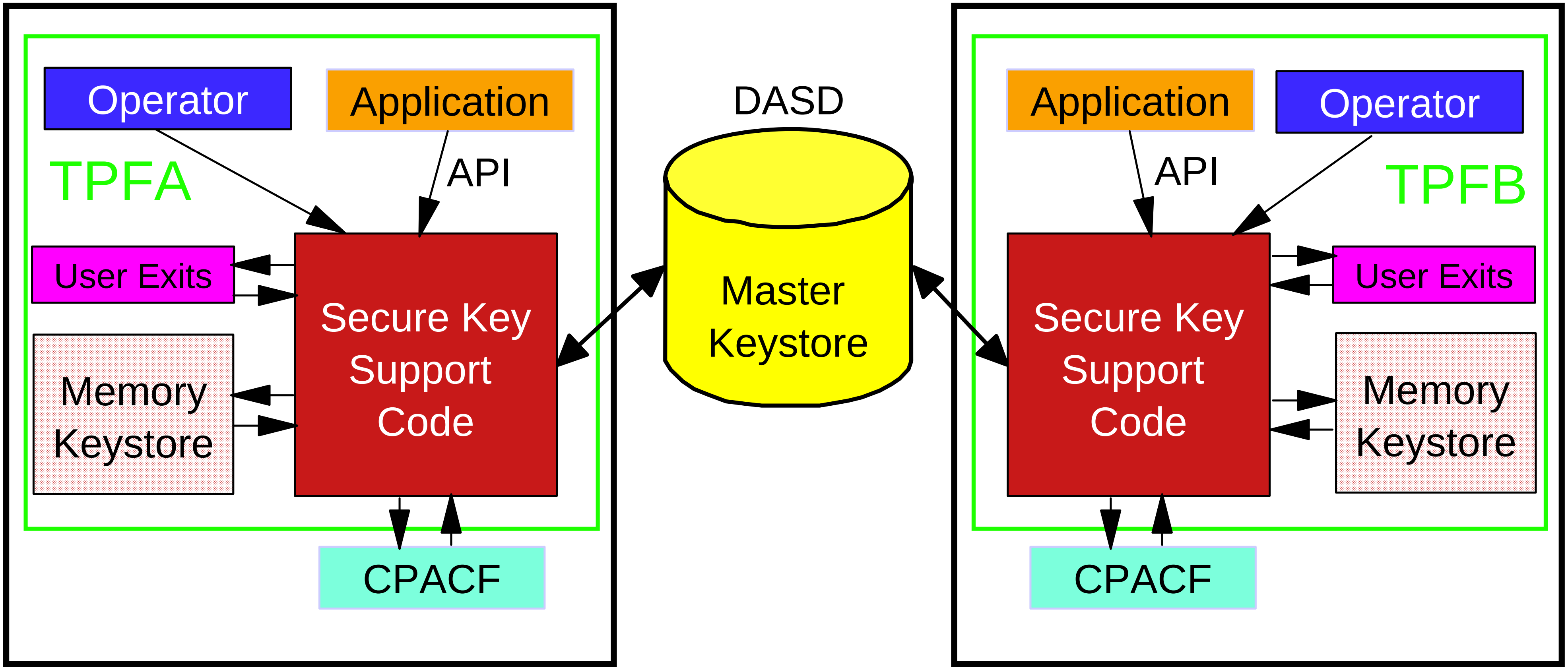
Managing Cryptographic Keys

- Key Management Functions:
  - Create encryption keys
  - Store keys
    - In database that is backed up
  - Change encryption keys
    - Common practice is that key values are changed every few months
  - Archive/access old keys
    - To decrypt data that was encrypted using an old key
  - Control and log key usage by applications
- Secure Key Management
  - All of the above, plus the ability to hide the key value from applications, operators, coverage staff, and so on

z/TPF Secure Key Management (PUT 3 APAR PJ31450)

- Enables you to create and manage symmetric encryption keys in a secure manner
- Applications can use the support to protect:
  - Data in flight
    - Sensitive data flowing over the network using private protocols or middleware that does not have encryption built in
  - Data at rest
    - Sensitive data stored on disk, stored or tape, or in a user table in memory
- Supports DES, Triple-DES (TDES), and AES-128 ciphers
  - In regular and cipher block chaining (CBC) mode
- Designed for high-volume applications

z/TPF Secure Key Management Components



Description of Components

- Master Keystore
  - Persistent copy of encryption/decryption keys on DASD
  - Shared by all processors in the complex
- Memory Keystore
  - Copy of the master keystore information in memory on each CPU
  - Exists for performance reasons
- Operator Interface
  - Commands to create/activate/change keys, display keystore information, backup/restore keystore information
- Application Interface
  - APIs to encrypt and decrypt data using secure keys
  - API to add a key to the keystore
- User Exits
  - Control and log key usage (encrypt/decrypt data APIs)
  - Control and log keystore adds (add key API)

How Secure is Secure?

- Each user key in the master keystore is encrypted using a different triple-DES master key and other cryptographic methods
- Integrity of each user key in the master keystore is protected by multiple SHA-1 digests
- Memory keystore resides in "hidden memory" which is visible only to a subset of the secure key management code
  - This memory is not visible to application programs or the rest of the z/TPF operating system
  - Memory keystore cannot be displayed using TPF operator commands and its contents are never included in dumps
  - For added security, keys in the memory keystore are not in the clear
- Code that manipulates master keystore information and accesses the memory keystore is object code only (OCO)
- Contents of an entry in the master or memory keystore are verified each time before a key is used
- User exits to control which applications can use which keys

Data Encryption Steps

1. Application issues an "encrypt data" API passing the data to be encrypted and the name of the encryption key to use
   - Application does not know the value of the encryption key
   - Application may not even know the cipher being used
2. User exit is called to verify this program is allowed to use the specified encryption key (and can be used to log key usage)
3. Secure key management code looks up the encryption key name in the memory keystore to get the associated cipher and key value
4. Secure key management code invokes CPACF to encrypt the data
5. Control is returned to the application program

z/TPF Secure Key Management Support Highlights

- Ability to change encryption key values (and in some cases upgrade the cipher) without requiring any application program changes
- Can scale to hundreds of thousands of crypto operations per second
- Ability to control and log key usage by applications
- Ability to control who can create keys (operators and applications)
- Ability to backup and restore keys
- Ability to migrate your existing keys to this support
- Safeguards to prevent a corrupted key (accidental or intentional) from ever being used
- APIs to encrypt and decrypt data using secure keys
  - Use in conjunction with message digest APIs enables your applications to ensure data integrity (detect data corruption)
- Performance and archive advantages over external crypto box solutions
- To summarize, a solution that enables you to protect vital data, and does so with traditional TPF scalability and performance characteristics

But Wait, There's More!

- More detailed presentation at tomorrow's communications subcommittee meeting, including:
  - Operator procedures
  - Application/database design considerations
  - Sample application logic
  - Integration with data integrity
  - Performance data
  - More bad jokes

Closing Message

If you want to do this...

... go to the casino

Data security is one area where you do not want to roll the dice!

Trademarks

IBM is a 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.

Notes

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.