zSTSU/RDS Gaitherburg

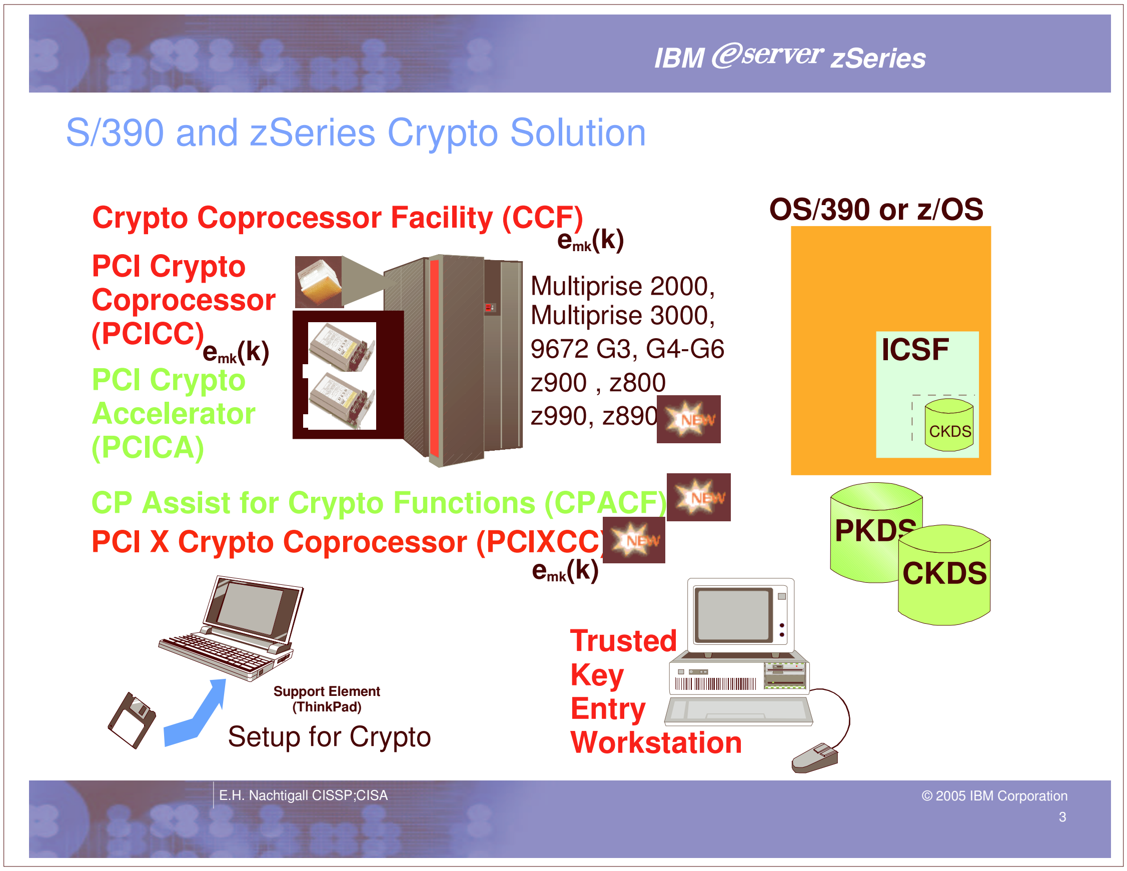
August 5, 2005

IBM Cryptographic Update

ON DEMAND BUSINESS™

Agenda

- zServer then and now
  - cryptographic hardware
- Free Crypto
  - CCF
  - CPACF
- Keys and more keys
  - Secure key
    - traditional Banking
  - Other
    - SSL
- SSL Only?
- Shop zSeries only?
- PassPhrase
- IMS/DB2 Encryption PRPQ
- TKE



S/390 and zSeries Crypto Solution

Crypto Coprocessor Facility (CCF) $e_{mk}(k)$

PCI Crypto Coprocessor (PCICC) $e_{mk}(k)$

PCI Crypto Accelerator (PCICA)

Multiprise 2000,
Multiprise 3000,
9672 G3, G4-G6
z900 , z800
z990, z890 NEW

CP Assist for Crypto Functions (CPACF) NEW

PCI X Crypto Coprocessor (PCIXCC) NEW $e_{mk}(k)$

OS/390 or z/OS

ICSF

CKDS

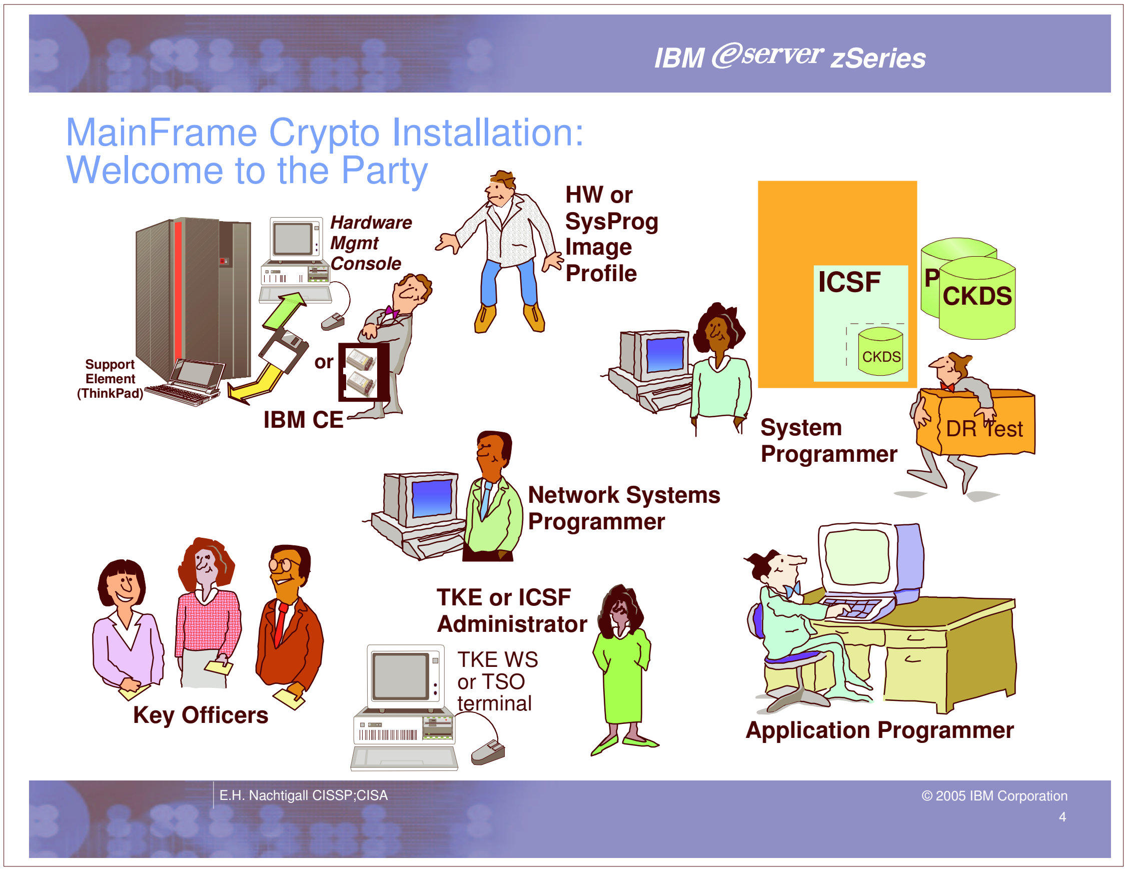
PKDS

CKDS

Support Element (ThinkPad)

Setup for Crypto

Trusted Key Entry Workstation



MainFrame Crypto Installation: Welcome to the Party

Hardware Mgmt Console

Support Element (ThinkPad)

or

IBM CE

HW or SysProg Image Profile

ICSF

CKDS

P

CKDS

System Programmer

DR Test

Network Systems Programmer

Key Officers

TKE or ICSF Administrator

TKE WS or TSO terminal

Application Programmer

Crypto is FREE!!!!!

- Set the expectation

- z900, turn it on, forget about it
  - any users will have full crypto, if they need it or not
    - many not even aware of functionality, just automagically use it
  - Super, super fast. 40X competitive crypto
    - affinity issues to CP 0 and CP 1
    - hard coded function, no dynamic update

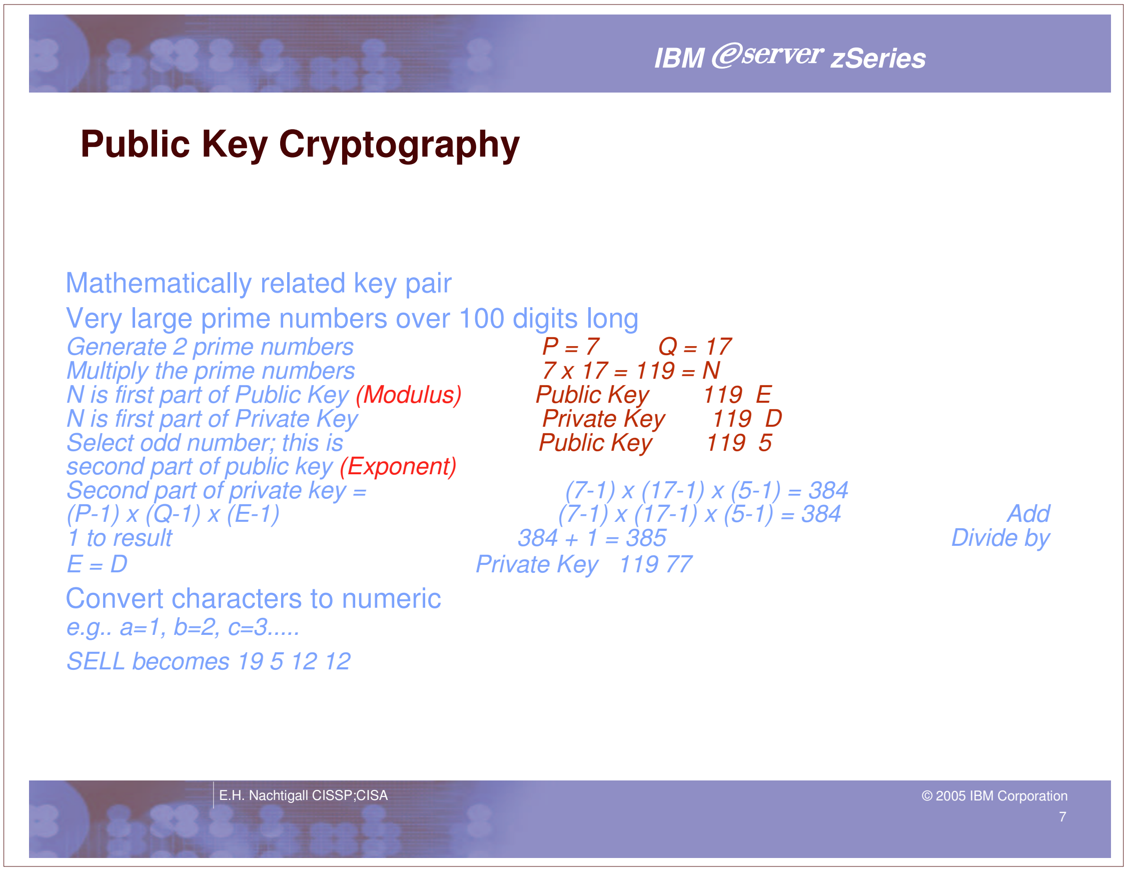
Crypto is FREE!!!!! ...performance or security are NOT

- Set the expectation

- z990/z9, turn it on, USELESS!!!!!!!!
  - basically a placeholder for what crypto functions you can add

- Most clients run their zServers at over 85% utilization
  - why pay for cycles not being used

- On Demand will help, but why expend the effort and cost!



Public Key Cryptography

Mathematically related key pair

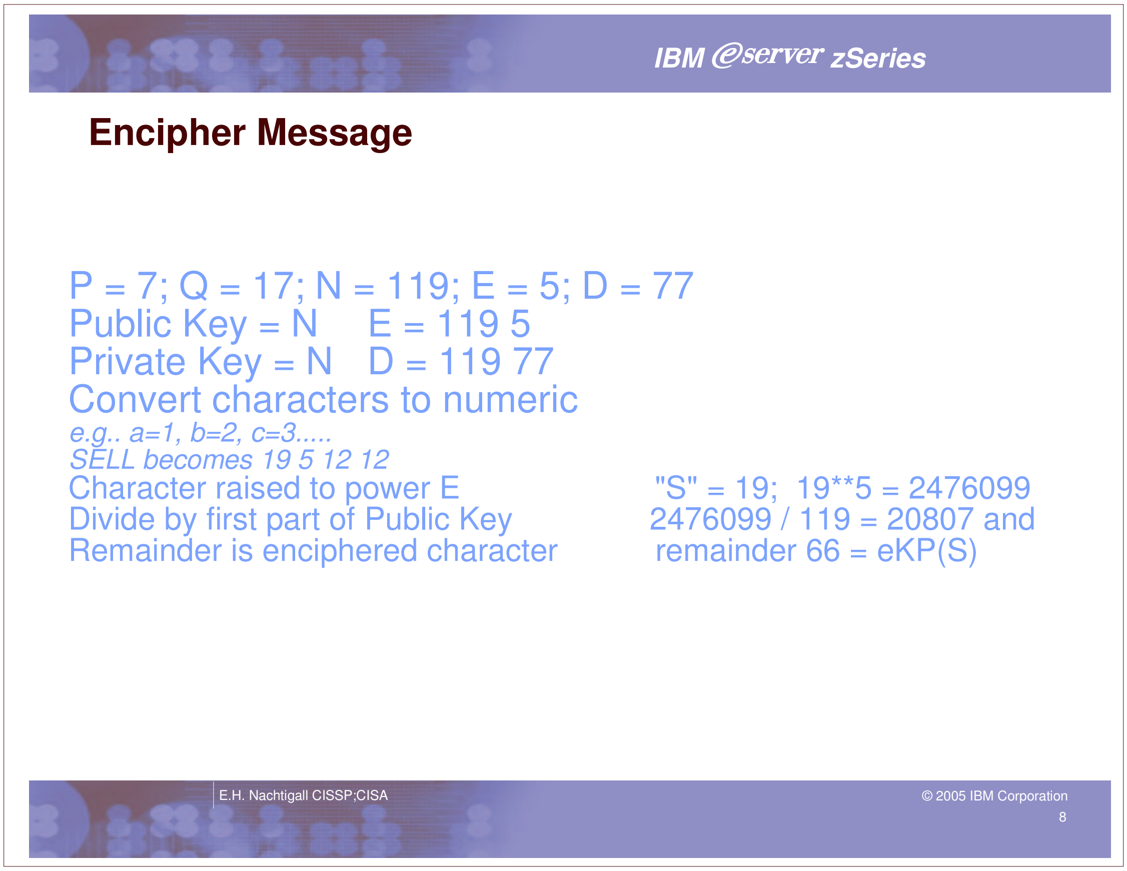
Very large prime numbers over 100 digits long

Generate 2 prime numbers *P = 7 Q = 17*
Multiply the prime numbers *7 x 17 = 119 = N*
N is first part of Public Key (Modulus) *Public Key 119 E*
N is first part of Private Key *Private Key 119 D*
Select odd number; this is *Public Key 119 5*
second part of public key (Exponent)
Second part of private key = *(7-1) x (17-1) x (5-1) = 384*
(P-1) x (Q-1) x (E-1) *(7-1) x (17-1) x (5-1) = 384* *Add*
1 to result *384 + 1 = 385* *Divide by*
E = D *Private Key 119 77*

Convert characters to numeric

e.g.. a=1, b=2, c=3.....

SELL becomes 19 5 12 12



Encipher Message

P = 7; Q = 17; N = 119; E = 5; D = 77
Public Key = N E = 119 5
Private Key = N D = 119 77
Convert characters to numeric
e.g.. a=1, b=2, c=3.....
SELL becomes 19 5 12 12
Character raised to power E "S" = 19; 19**5 = 2476099
Divide by first part of Public Key 2476099 / 119 = 20807 and
Remainder is enciphered character remainder 66 = eKP(S)

Decipher Message

P = 7; Q = 17; N = 119; E = 5; D = 77
Public Key = N E = 119 5
Private Key = N D = 119 77

a=1, b=2, c=3.....

SELL becomes 19 5 12 12

Character raised to power E
Remainder raised to power D
Result divided by first part of Private Key and Public Key
Remainder is numeric equivalent of character sent

66 ** 77 = 1273.......
1273..... / 119 = 1069
and remainder of 19
19 = "S"

Performance

- z9 is faster than z990
- CPACF should be faster
- Secure Key, same hardware
  - may see slight improvement due to ICSF parsing/routing code running on faster CP's
  - Do not expect to see double digit performance improvements

Performance

- CKDSN(CSF.Z990DEC.CKDS)
- PKDSN(CSF.Z990DEC.PKDS)
- DOMAIN(3)
- COMPAT(YES)
- SSM(YES)
- KEYAUTH(NO)
  - doubles number of crypto operations when using a key label up to 30% boost with (NO)
- CHECKAUTH(NO)
  - No SAF calls for authorized programs
- TRACEENTRY(1000)
- USERPARM(USERPARM)
- COMPENC(DES)
- REASONCODES(ICSF)
- PKDSCACHE(64)

APAR OA08172

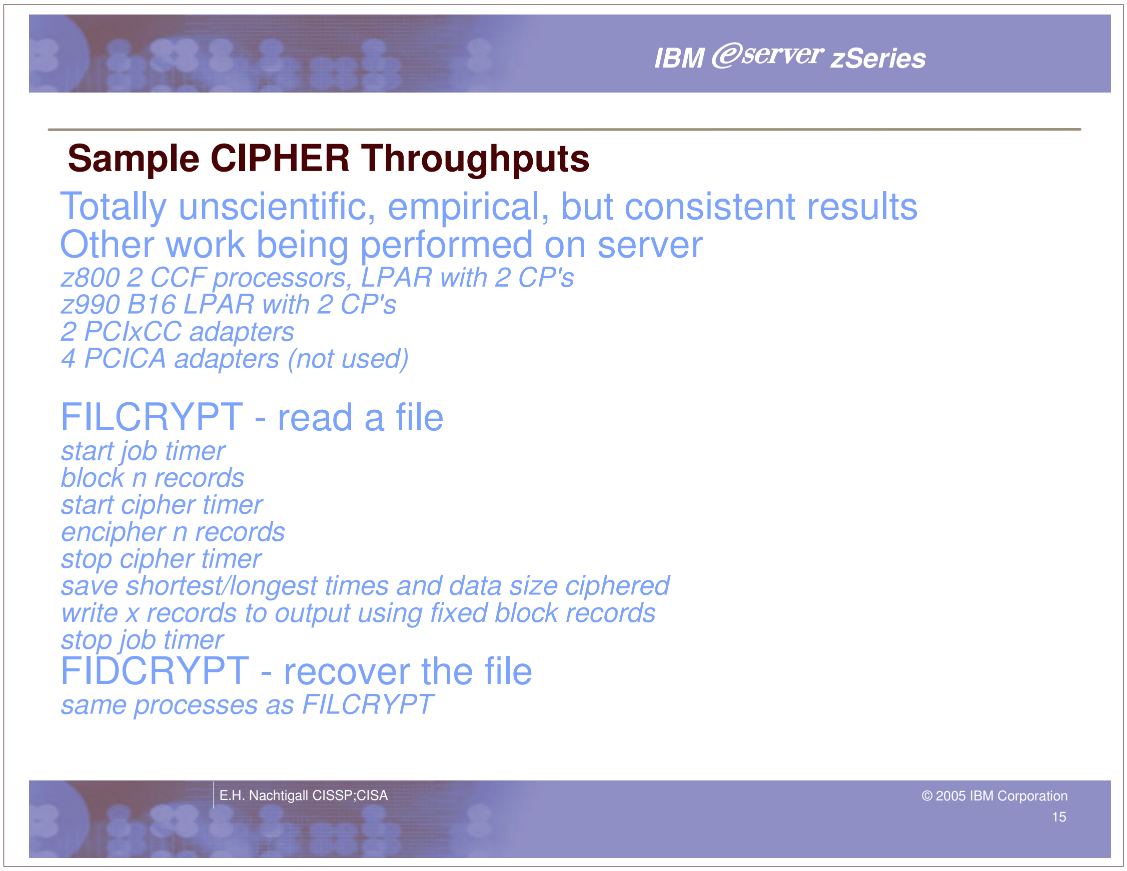
- Clear key tokens in the ICSF CKDS
  - Key Token Build can build tokens
  - Key Record Write can write tokens
  - Key Record Read can not read tokens (unless SUP/KEY0)
  - CSNBSYD - Symmetric Decipher
  - CSNBSYE - Symmetric Encipher
- KGUP support
- Designed for fast DES CPACF access (via ICSF API CSNBSYD andCSNBSYE)
  - Updates to IBM Data Encryption for IMS and DB2 Databases
- Allows centralized storage of CPACF tokens within the ICSF CKDS
- Recommend RACF protection of Key Label especially for shared CKDS with other systems

Data Encryption for IMS DB2

- ISPF front end utility
- IMS Segment Edit/Compression exit.
  - segement level encryption
- DB2 EDITPROC exit
  - table level encryption
- Define encryption keys
- Provide key label to exit routine
  - SAF CSFKEYS
  - SAF CSFSERV
    - secure key uses CSFENC/CSFDEC
    - clear key (z9/z990/z890) uses CSFSYE/CSFSYD
- Unload data sets
- Install exit
- Reload data

Bulk File Encryption

- Read a sequential file
  - automated key management
    - user provided keys; dynamically generated keys
    - PKA protection
    - JAVA reference code
  - encrypt data
    - DES/TDES/AES
  - write encrypted file
- Read encrypted archive
  - decrypt data
  - write re-constructed file



Sample CIPHER Throughputs

Totally unscientific, empirical, but consistent results

Other work being performed on server

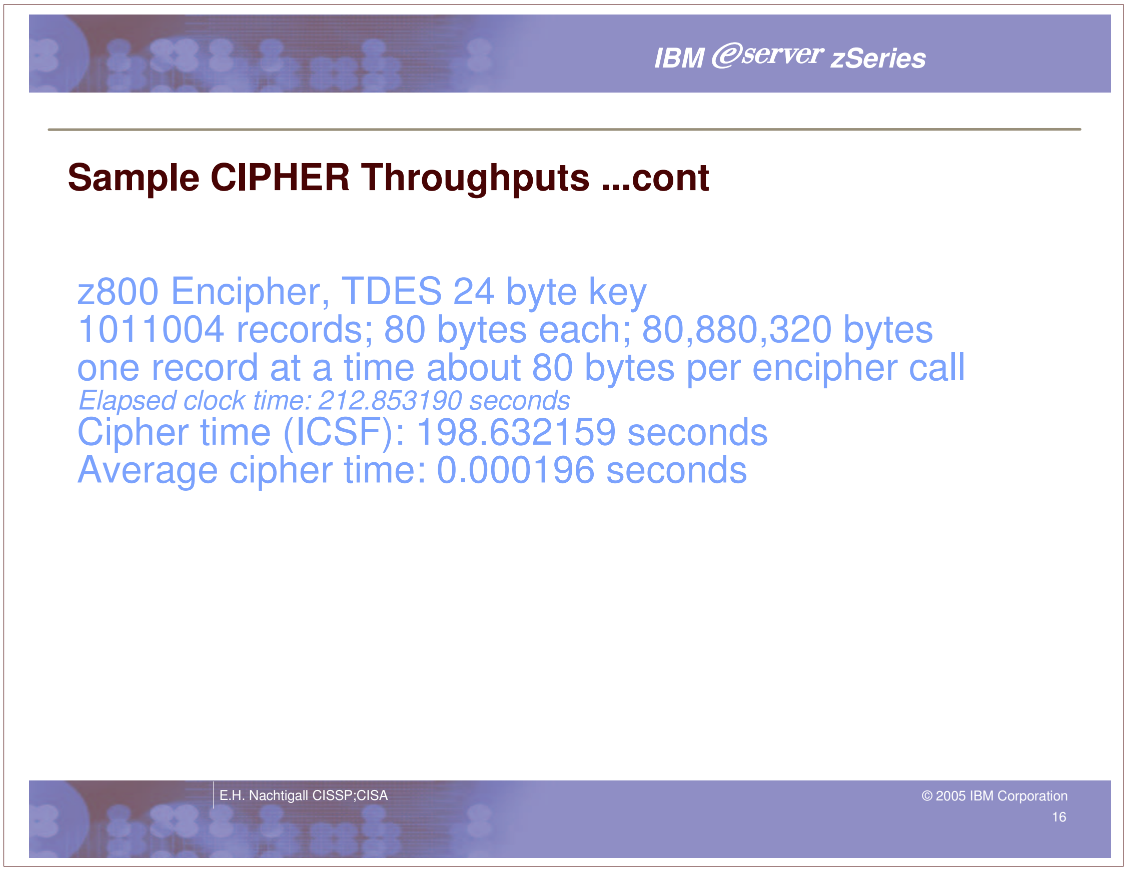
z800 2 CCF processors, LPAR with 2 CP's
z990 B16 LPAR with 2 CP's
2 PCIxCC adapters
4 PCICA adapters (not used)

FILCRYPT - read a file

start job timer
block n records
start cipher timer
encipher n records
stop cipher timer
save shortest/longest times and data size ciphered
write x records to output using fixed block records
stop job timer

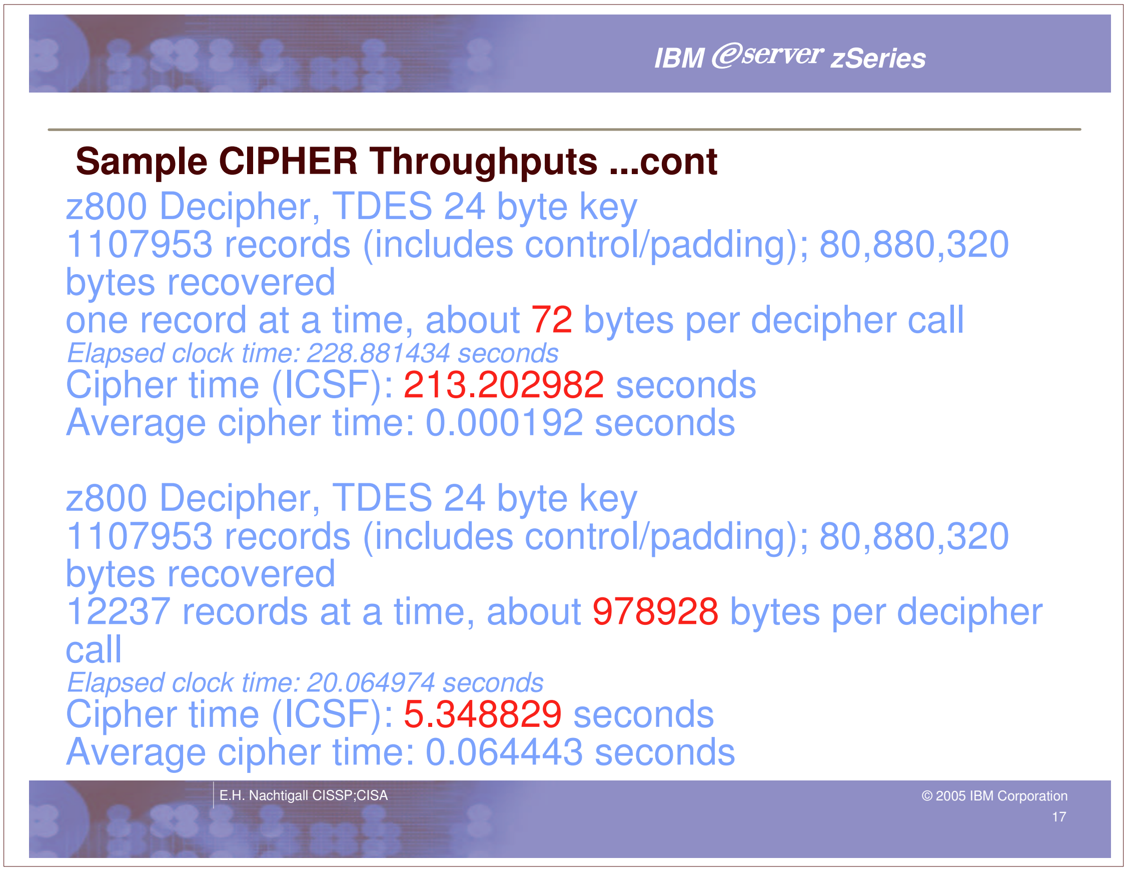
FIDCRYPT - recover the file

same processes as FILCRYPT



Sample CIPHER Throughputs ...cont

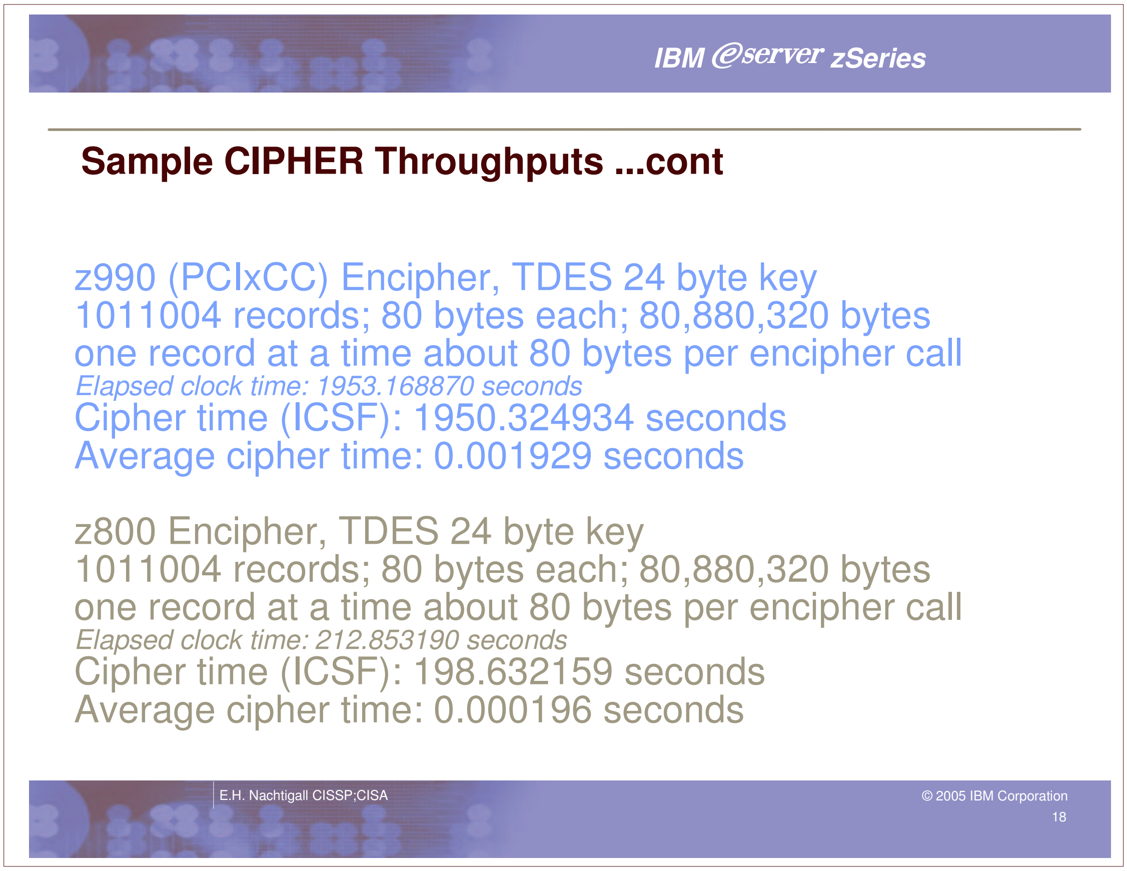
z800 Encipher, TDES 24 byte key
1011004 records; 80 bytes each; 80,880,320 bytes
one record at a time about 80 bytes per encipher call
Elapsed clock time: 212.853190 seconds
Cipher time (ICSF): 198.632159 seconds
Average cipher time: 0.000196 seconds



Sample CIPHER Throughputs ...cont

z800 Decipher, TDES 24 byte key
1107953 records (includes control/padding); 80,880,320 bytes recovered
one record at a time, about 72 bytes per decipher call
Elapsed clock time: 228.881434 seconds
Cipher time (ICSF): 213.202982 seconds
Average cipher time: 0.000192 seconds

z800 Decipher, TDES 24 byte key
1107953 records (includes control/padding); 80,880,320 bytes recovered
12237 records at a time, about 978928 bytes per decipher call
Elapsed clock time: 20.064974 seconds
Cipher time (ICSF): 5.348829 seconds
Average cipher time: 0.064443 seconds



Sample CIPHER Throughputs ...cont

z990 (PCIxCC) Encipher, TDES 24 byte key
1011004 records; 80 bytes each; 80,880,320 bytes
one record at a time about 80 bytes per encipher call
Elapsed clock time: 1953.168870 seconds
Cipher time (ICSF): 1950.324934 seconds
Average cipher time: 0.001929 seconds

z800 Encipher, TDES 24 byte key
1011004 records; 80 bytes each; 80,880,320 bytes
one record at a time about 80 bytes per encipher call
Elapsed clock time: 212.853190 seconds
Cipher time (ICSF): 198.632159 seconds
Average cipher time: 0.000196 seconds

Sample CIPHER Throughputs ...cont

z990 (PCIxCC) Decipher, TDES 24 byte key
1107953 records (includes control/padding); 80,880,320 bytes recovered
one record at a time, about 72 bytes per decipher call
Elapsed clock time: 2138.434733 seconds
Cipher time (ICSF): 2133.420225 seconds
Average cipher time: 0.001925 seconds

z990 (PCIxCC) Decipher, TDES 24 byte key
1107953 records (includes control/padding); 80,880,320 bytes recovered
12237 records at a time, about 978928 bytes per decipher call
Elapsed clock time: 44.783924 seconds
Cipher time (ICSF): 32.060308 seconds
Average cipher time: 0.386268 seconds

Sample CIPHER Throughputs ...cont

z990 (ICSF Clear key) Encipher, TDES 24 byte key
1011004 records; 80,880,320 bytes
one record at a time, about 80 bytes per encipher call
Elapsed clock time: 19.151577 seconds
Cipher time (ICSF): 12.598216 seconds
Average cipher time: 0.000012 seconds

z990 (ICSF Clear key) Encipher, TDES 24 byte key
1011004 records; 80,880,320 bytes
12237 records at a time, about 986960 bytes per encipher call
Elapsed clock time: 15.384605 seconds
Cipher time (ICSF): 0.509520 seconds
Average cipher time: 0.006213 seconds

Sample CIPHER Throughputs ...cont

z990 (native CPACF Clear key) Encipher, TDES 24 byte key

1011004 records; 80,880,320 bytes

one record at a time, about 80 bytes per encipher call

Elapsed clock time: 11.020412 seconds

Cipher time (CP): 1.105856 seconds

z990 (Native CPACF Clear key) Encipher, TDES 24 byte key

1011004 records; 80,880,320 bytes

12237 records at a time, about 986960 bytes per encipher call

Elapsed clock time: 13.656816 seconds

Cipher time (CP): 0.545292 seconds

	z800 Cipher Time (seconds)	z800 Clock Time (seconds)	z990 Cipher Time (seconds)	z990 Clock Time (seconds)	CPACF Clear Key Cipher Time	CPACF Clear Key Clock Time
72 Byte Records	**198.632159**	**212.853190**	**1950.324934**	**1953.168870**	**1.105856** **523 Mbyte 8.765327 seconds**	**11.020412** **523 Mbyte 114.764597 seconds**
980,000 Byte Records	5.348829	20.064974	32.060308 523 Mbyte 210 seconds	44.783924 523 Mbyte 318 seconds	0.545292 523 Mbyte 3.269381 seconds	13.656816 523 Mbyte 106.562428 seconds

CPACF Operation Codes

- KM (ECB Encrypt, key is X'0123456789ABCDEFFEDCBA9876543210' and looks like X'0123456789ABCDEFFEDCBA9876543210' in memory)
- KMC (CBC Encrypt, same key exposure)
- KMAC (Message Authentication, same key exposure)
- KIMD (Intermediate SHA-1, no keys)
- KLMD (Last SHA-1, no keys)

- SSL can use KIMD/KLMD/KMC
  - also uses exponentiating arithmetic which is requires special hardware or significant CPU mathematic instructions

- CSNBENC (CBC secure encrypt, key value appears as X'010000000000C000F8775639CD49A68CBB84F2D693BF510800000...' in memory)

Clear Key Crypto (CPACF)

High Speed Symmetric Algorithms imbedded in each CP

available via ICSF as API's (CSNBSYD/CSNBSYE) or as new operation codes (OP CODES)

"SOFTWARE ENCRYPTION" with algorithm code in hardware

DES TDES SHA-1 AES (MD5 and AES via ICSF)

Encryption/Decryption keys are clear (not encrypted) in user address space

typically not appropriate or allowed for sensitive processing such as VISA, MasterCard, INTERAC, LINK

can be mitigated to offer certain in-house functions

file archive to tape

ICSF user defined functions, keys in clear in the ICSF address space only

Specifically designed for WEB (SSL/TLS/TN3270/FIREWALL) type applications, short duration applications, throw-away key values

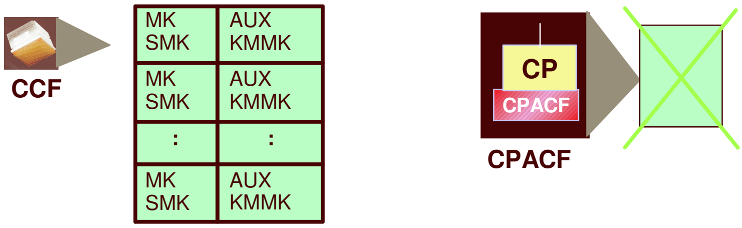
z890/z990/z9 vs z900 Base Crypto: Clear vs Secure

z890/z990/z9 Base Crypto

Central Processor Assist for Cryptographic Function (CPACF)
Requires hardware setup, configuration data load, ICSF active
Does Not require Master Key Loading

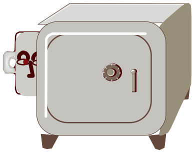
z800/z900 Base Crypto

Cryptographic Coprocessor Facility (CCF)
Requires hardware setup, configuration data load, ICSF active
Requires Master Key Loading



MK SMK	AUX KMMK
MK SMK	AUX KMMK
:	:
MK SMK	AUX KMMK

Secure Key Operations & Clear Key Operations



Secure operation implies that the interruption of the activity will not expose any unprotected key value.

Previous IBM crypto products including software require secure key usage

Key Value Protection?

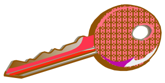
Actual value used in cryptographic algorithm is also encrypted or securely protected from view
Level of protection

Actual key value is not exposed to view or copy once imported into a cryptographic system

Actual key value is protected until use is required.

Actual key value is restored and used for cryptographic operation within crypto system

z990/z9 Cryptographic Hardware



Base Crypto - Central Processor Assist for Cryptographic Function (CPACF)

Performs clear key encrypt/decrypt, MAC and SHA-1 hashing
Feature Code 3863 Required to obtain Configuration Data
No feature code to indicate crypto hardware

Accelerator for SSL - PCI Cryptographic Accelerator (PCICA)

Performs decrypt/encrypt of pre-master secret during handshake
Handles same throughput rate per card as on z900, approx. (2100 handshakes per second)
CPACF and Feature Code 3863 required

May Require Software updates

For SSL users - no exploitation of crypto hardware for SSL handshake performance improvements without ordering PCICA, PCIXCC or CEX2C!!!!

Secure Key vs. CLear Key

- Processing paranoia
  - "...just 'cause I'm paranoid, doesn't mean they're not out to get me..."
- Internally (tape backup)
  - dual key custody or keys kept out of general access
  - exposure is our customers and reputation only
- Externally (VISA CISP)
  - dual custody, perhaps even between partners
    - no possibility of internal exposure
  - protects ME if I process YOUR clients information (ATM/POS)
  - limited exposure, if any

Clear vs Secure Has Meaning on z890/z990/z9

With CPACF

- No protected key values used with API
- CSNBSYE/SYD
  - for data privacy
  - DES / TDES / (AES)
- CSNBECO/CSNBDCO
  - for data privacy
  - DES - ECB; no chaining
- SHA-1 and (MD5)
- Utility based functions
- No Performance Support with Handshake

Accelerator

- CSNDPKE/PKD
  - to provide acceleration during SSL handshake
- No Hardware Acceleration for Client Authentication

no retail or banking function support

SSL Performance may change

Clear vs Secure Has Meaning on z890/z990/z9 . . .

- TKE Required to comply with dual key part entry and no exposure of key parts within network during entry
  - TKE 4.1+ required for application key entry
- PCIXCC/CEX2C Required to support
  - DUKPT, PIN processing applications
  - Retained Keys
  - Secure Application Keys in CKDS or application storage
  - Any old crypto applications that might be running production
    - PCF/CUSP
    - IDCAMS Repro using ENCIPHER/DECIPHER

Clear vs Secure Has Meaning on z890/z990 . . .

SSL based applications may not have the same performance as on z900/z800 when migrated to z990

Same throughput for the decrypt of premaster secret

Impact may be felt if requiring the SSL optional actions

Server requires temporary RSA key because certificate key length or purpose cannot be used in session

Client Authentication required

Impact is also due to CPACFs not supporting the RSA functions supported on CCFs

Plan for Performance

Expectations from previous benchmarks on non-z990 or on service level objectives

PCICA/CEX2C features may be required to offset CCF loss

PassPhrase

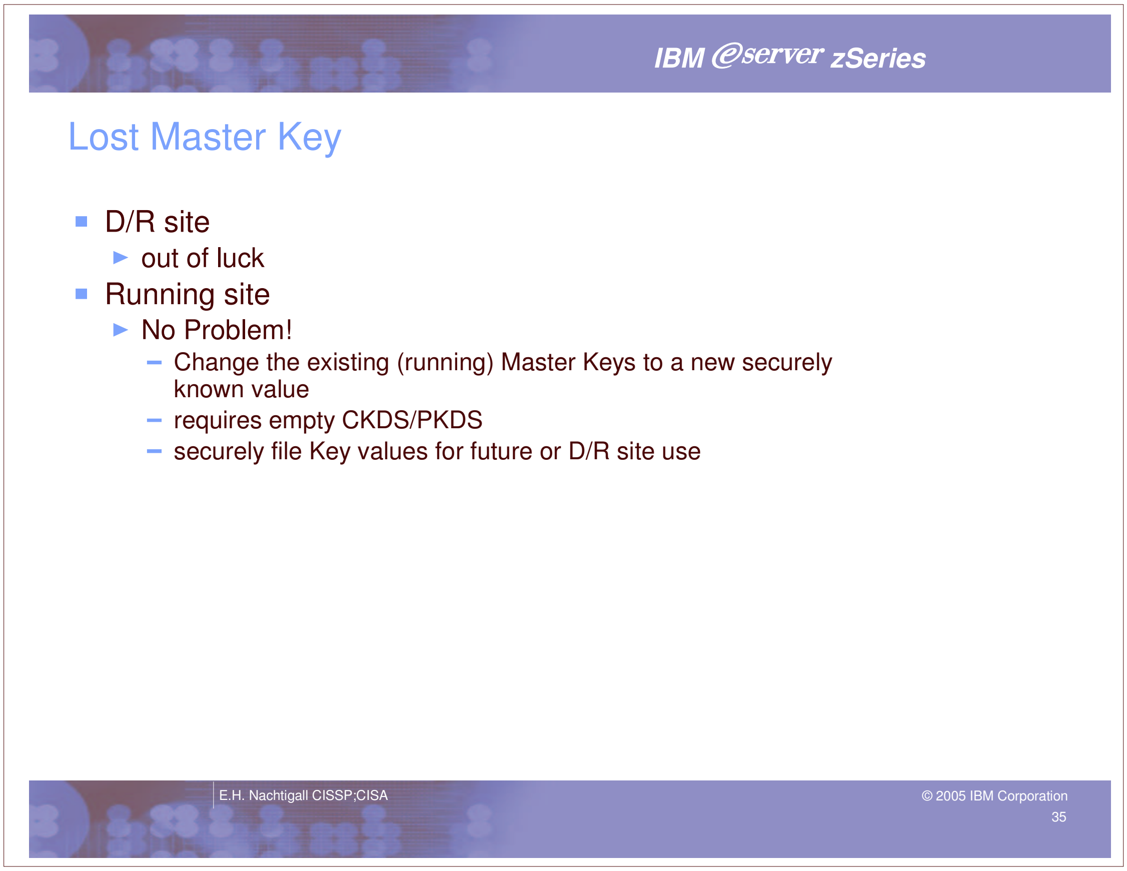
- Ideally Master Keys created from random numbers under multiple custody
- ICSF can create random numbers if a Master Key has been loaded (z9xx)
  - can you say "Catch-22"?
  - z9 has CPACF pseudo random number generation
- Z900 could load Mkey of zeroes to initialize crypto
- z990/z9 do not tolerate weak keys
- PassPhrase via ISPF or utility (CSFEUTIL)

PassPhrase

- Single entry
  - one person "owns" the keys
  - content dependant
    - blanks, capitals
  - Guessable?
- Requires "Clean" crypto module
  - no existing key values
  - can not be used to change Master Key
  - may not be enterable at D/R site
    - shared LPAR, keys loaded by prior user
- Ideal for ShopzSeries prior to z/OS 1.7 and z9

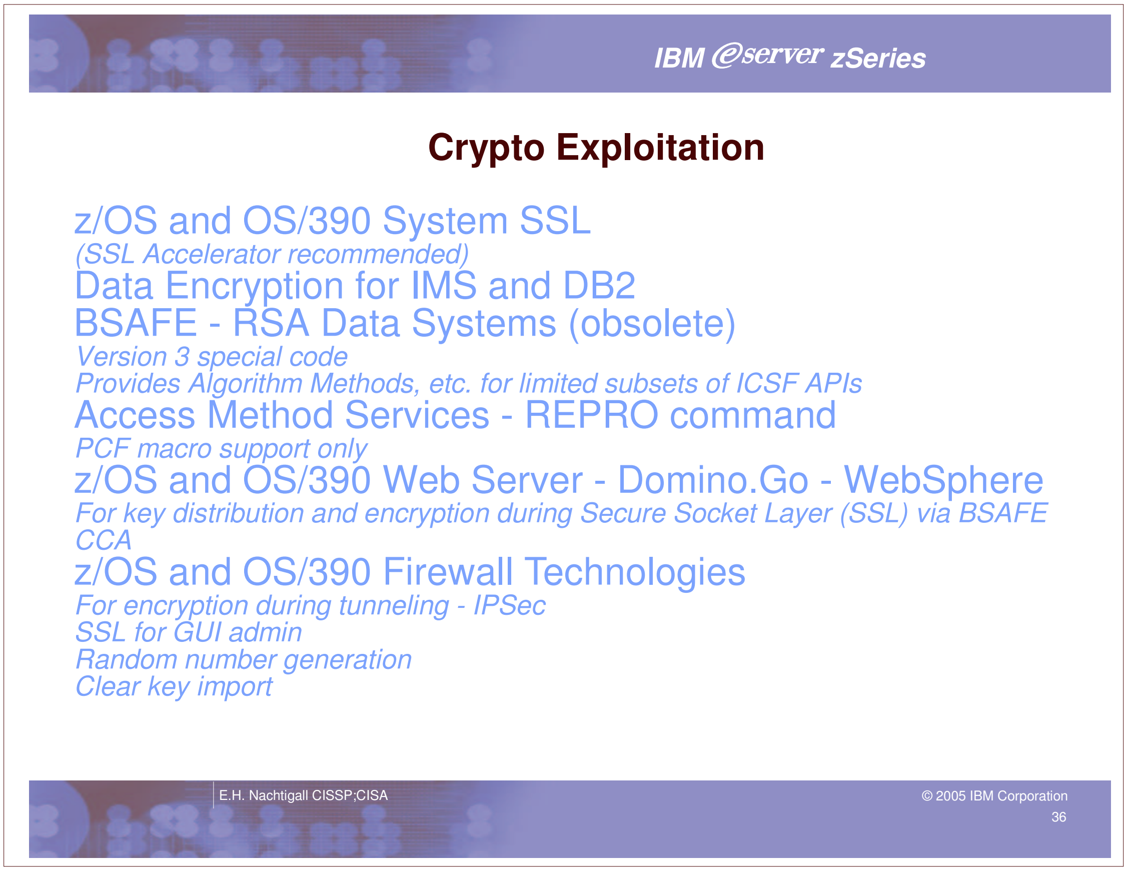
PassPhrase

- Invoke Pass Phrase Initialization, then:
  1. Invoke the ISPF Random Number Generator (each Key custodian does this)
  2. Generate the associated Check Sum
  3. Enter the Key parts
  4. Select Change Master Key. This will require a new, empty CKDS
  5. Update the ICSF parameter file to point to this new CKDS
  6. Enter production keys as required.
  7. Generate and enter/change PKA Master Keys
- Above can now be a standard procedure for later Master Key change



Lost Master Key

- D/R site
  - out of luck
- Running site
  - No Problem!
    - Change the existing (running) Master Keys to a new securely known value
    - requires empty CKDS/PKDS
    - securely file Key values for future or D/R site use



Crypto Exploitation

z/OS and OS/390 System SSL
(SSL Accelerator recommended)
Data Encryption for IMS and DB2
BSAFE - RSA Data Systems (obsolete)
Version 3 special code
Provides Algorithm Methods, etc. for limited subsets of ICSF APIs
Access Method Services - REPRO command
PCF macro support only
z/OS and OS/390 Web Server - Domino.Go - WebSphere
For key distribution and encryption during Secure Socket Layer (SSL) via BSAFE
CCA
z/OS and OS/390 Firewall Technologies
For encryption during tunneling - IPSec
SSL for GUI admin
Random number generation
Clear key import

S/390 Crypto Exploitation . . .

DCE Security Server - RPC

For encryption during remote procedure call
Random number generation
Clear key import

VTAM

Session Level Encryption - V3R4.1

Encryption

Message Authentication - V4R4

MAC

Communication Server/2

S/390 Payment Gateway

Extract public key info - modulus
Build a PKA token
Add keys to PKDS via PKDS import
SET processing using OAEP decompose