VSE/ESA Security

in a heterogeneous environment

Ingo Franzki

e-mail: ifranzki@de.ibm.com
VSE/ESA Development

VSESecurity.prz

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and / or other counties.

CICS*	IBM*	Virtual Image Facility
DB2*	IBM logo*	VM/ESA*
DB2 Connect	IMS	VSE/ESA
DB2 Universal Database	Intelligent Miner	VisualAge*
e-business logo*	Multiprise*	VTAM*
Enterprise Storage Server	MQSeries*	WebSphere*
HiperSockets	OS/390*	xSeries
	S/390*	z/Architecture
	SNAP/SHOT*	z/VM
		zSeries

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

LINUX is a registered trademark of Linus Torvalds

Tivoli is a trademark of Tivoli Systems Inc.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.
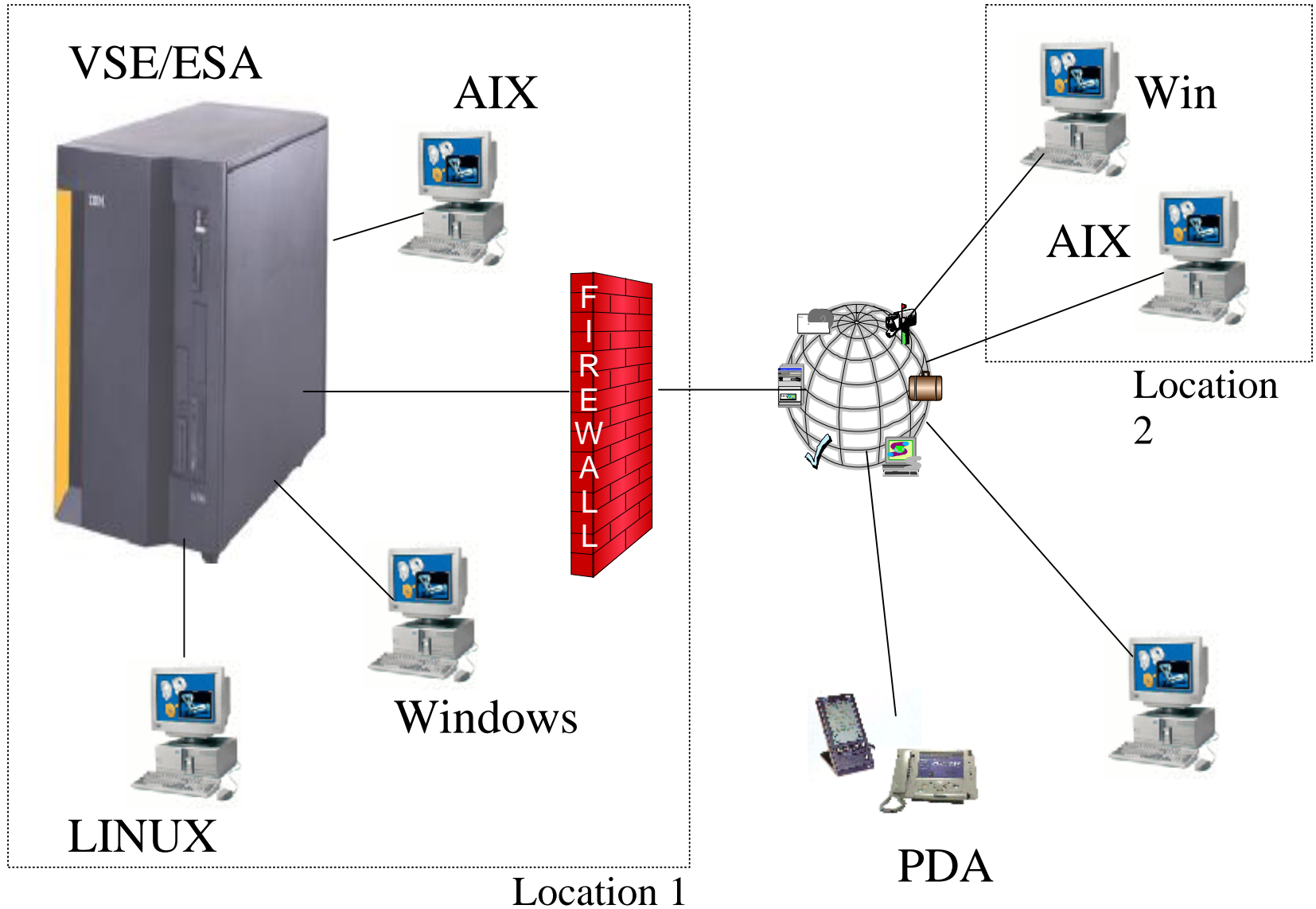
Intel is a registered trademark of Intel Corporation.

Security in a heterogeneous environment

VSE/ESA
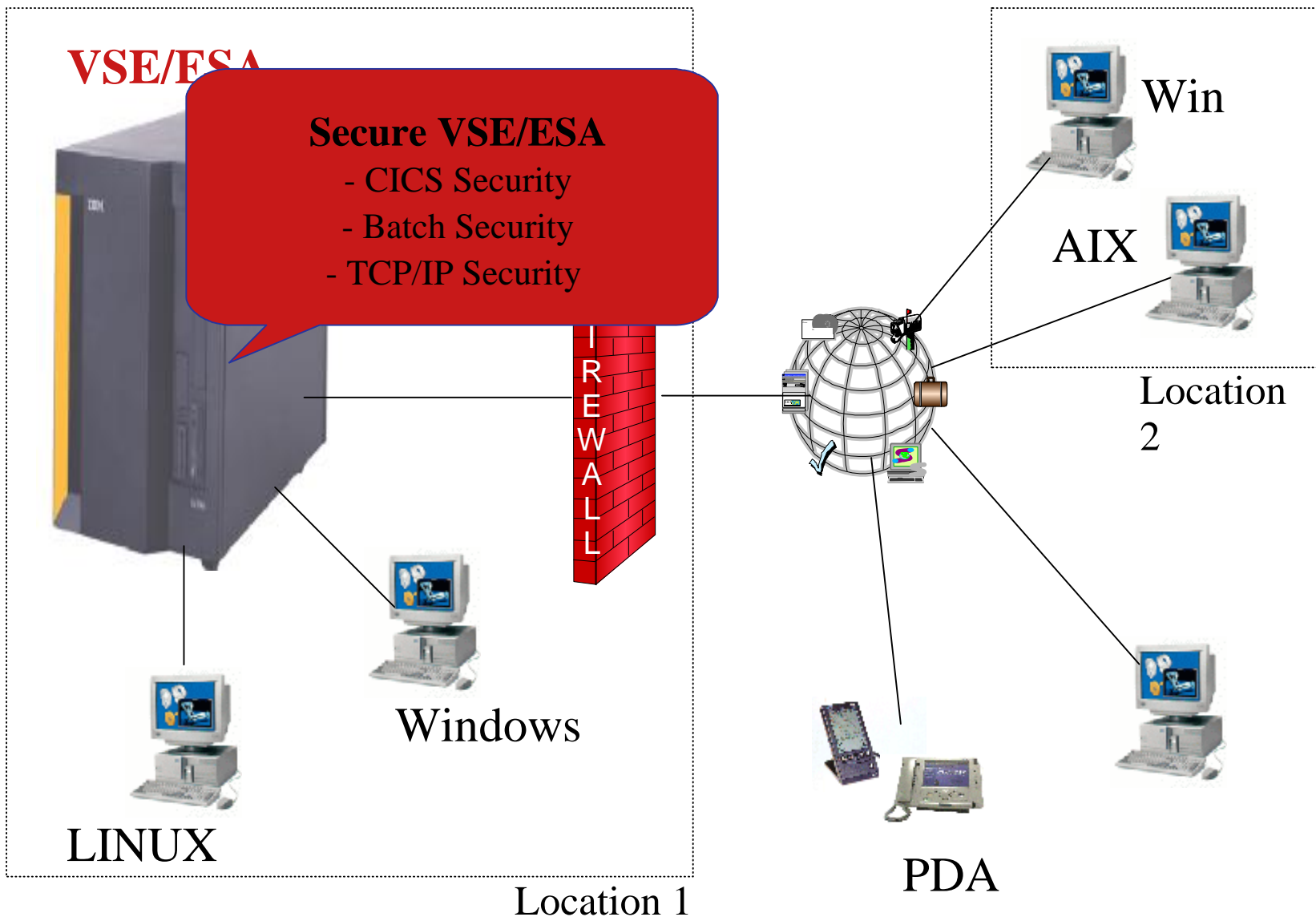
AIX

FIREWALL

Windows

LINUX

Location 1

Win

AIX

Location 2

PDA

VSESecurity.prz

Security in a heterogeneous environment

VSE/ESA

Secure VSE/ESA
- CICS Security
- Batch Security
- TCP/IP Security

FIREWALL

Win

AIX

Location 2

Windows

LINUX

Location 1

PDA

Security in a heterogeneous environment

VSE/ESA

AIX

Win

AIX

Secure remote access
- Logon security
- Connector Security
- Single Signon

Location 2

Windows

LINUX

PDA

Location 1

Security in a heterogeneous environment

VSE/ESA

AIX

Win

Secure Internet access
- Secure Socket Layer
- Virtual Private Network (VPN)
- Cryptography

F
I
R
E
W
A
L
L

Location 2

Windows

LINUX

PDA

Location 1

VSESecurity.prz
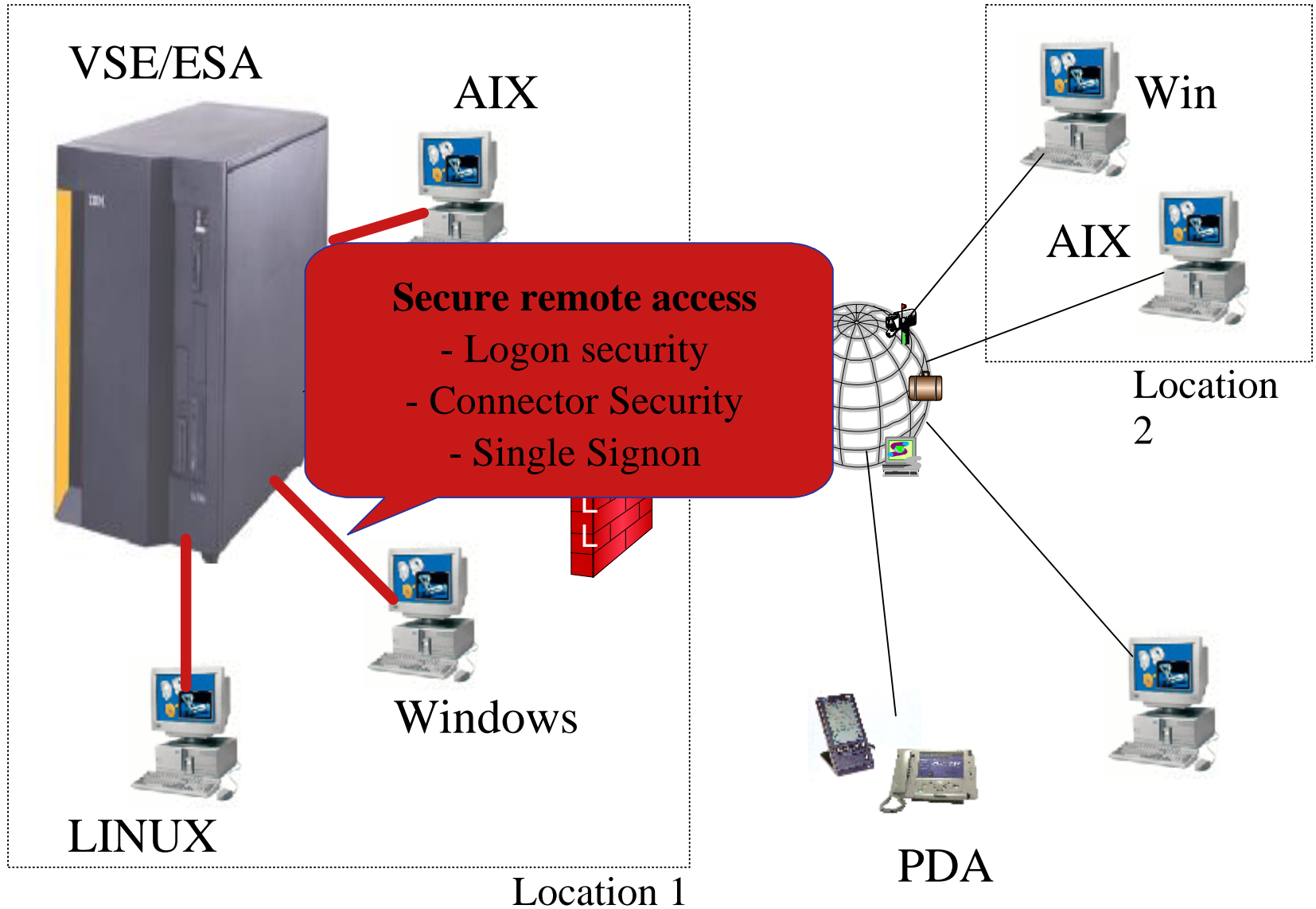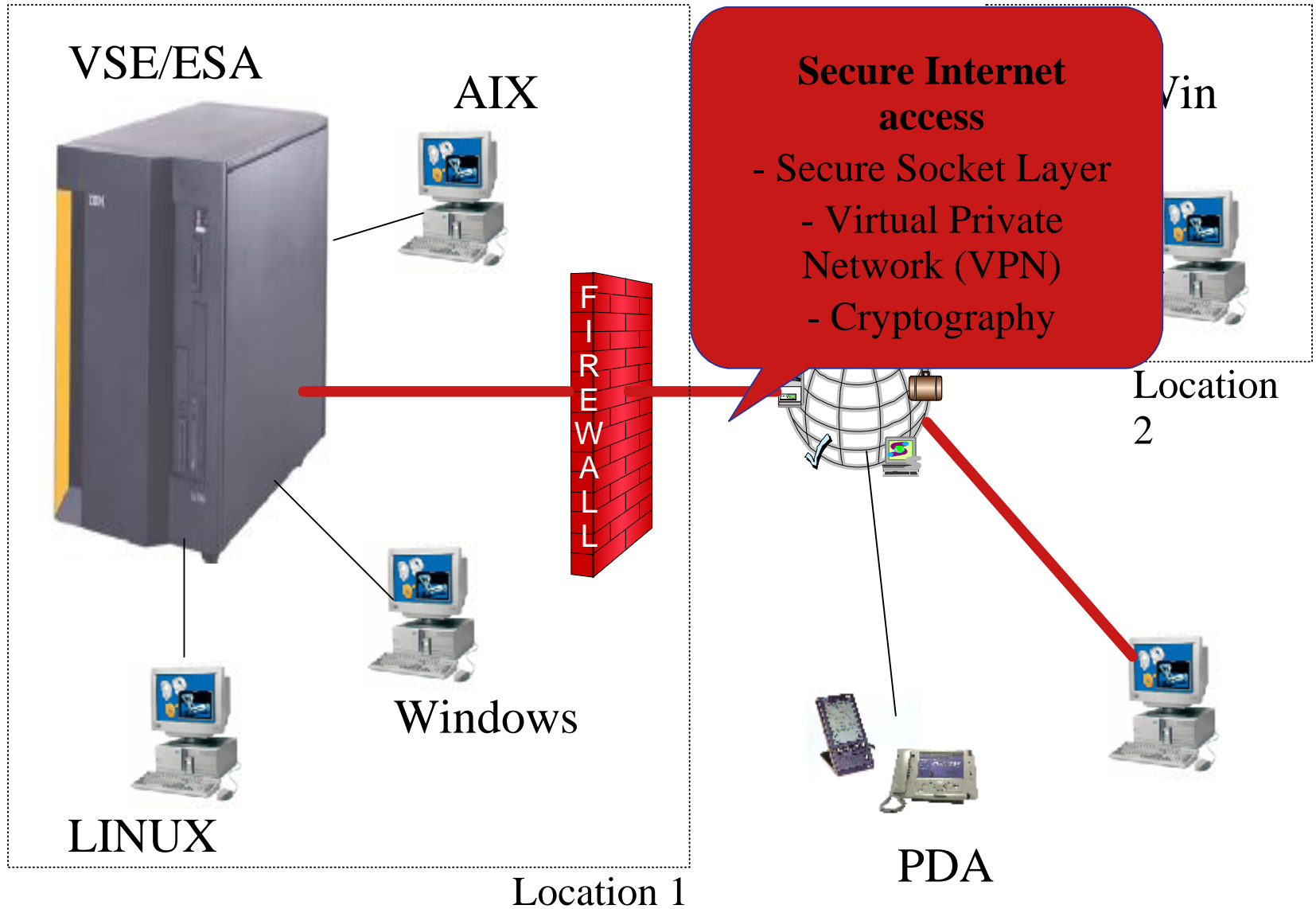
Security in a heterogeneous environment

Security is very important

→ Restrict access to systems

→ Keep secrets

→ Prove identity of users

→ Prevent data modification

Security can be very complex

→ In an heterogeneous environment

→ A lot of different servers and technologies

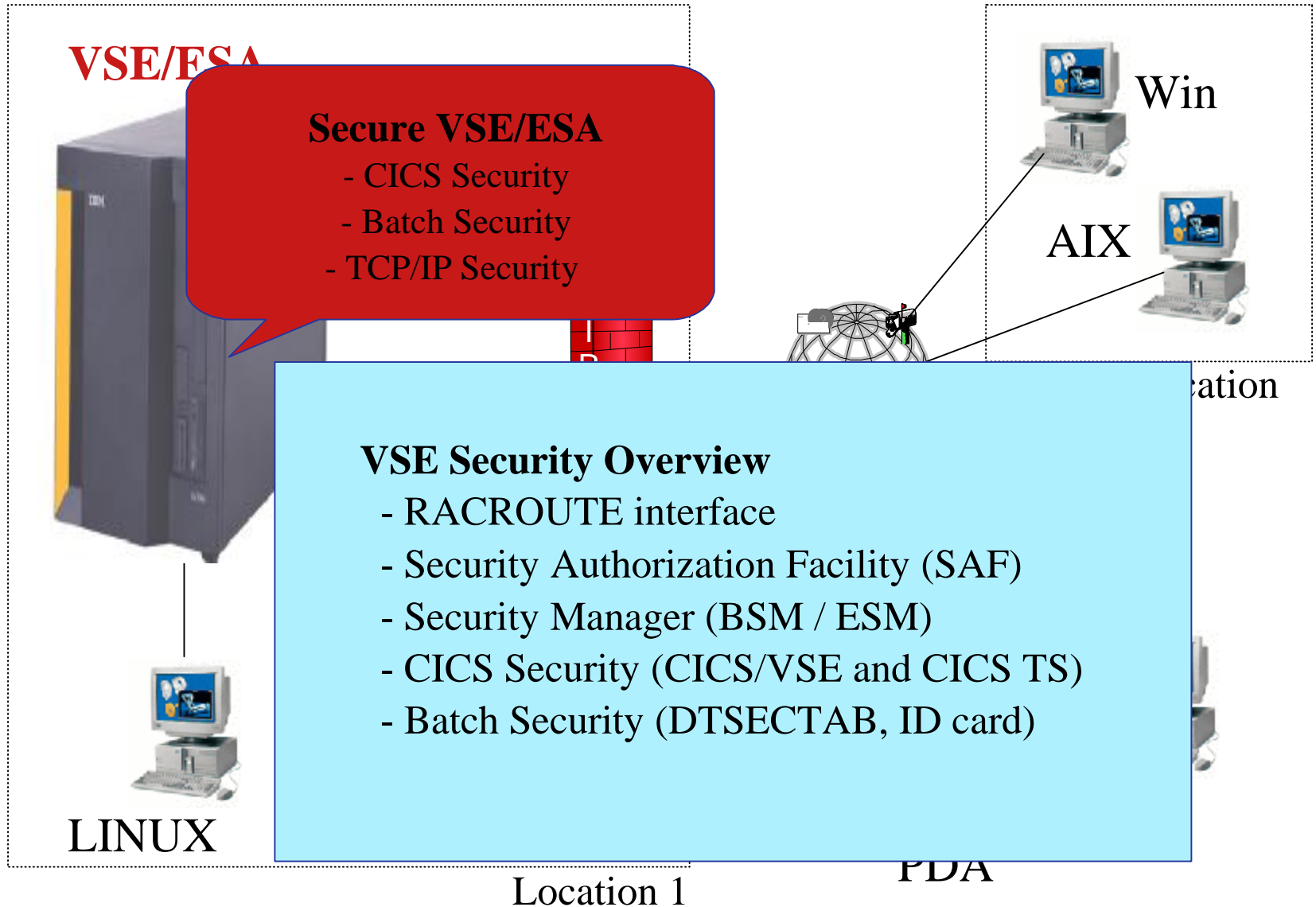## You must know what you are doing !

→ Incomplete security setup can be more dangerous than NO security

Security in a heterogeneous environment

VSE/ESA

Secure VSE/ESA
- CICS Security
- Batch Security
- TCP/IP Security

Win

AIX

VSE Security Overview
- RACROUTE interface
- Security Authorization Facility (SAF)
- Security Manager (BSM / ESM)
- CICS Security (CICS/VSE and CICS TS)
- Batch Security (DTSECTAB, ID card)

LINUX

Location 1

PDA

VS*e*

e-business

IBM

VSESecurity.prz

Why secure VSE/ESA ?

- Prevent unauthorized access to VSE/ESA and data
  - ► Keep secret data secret
  - ► Data modification by unauthorized users
- Prevent users from damaging the VSE/ESA system (maybe by accident)
  - ► Deletion of members or entries
  - ► Submisson of jobs

VSE Security Overview

- VSE/ESA 2.3 (or below)
  - ► SECHECK macro (DTSECTAB)
  - ► CICS/VSE internal security

- VSE/ESA 2.4, 2.5, 2.6 (2.7)
  - ► RACROUTE calls
  - ► Security Server (BSM/ESM)
  - ► Security decisions delegated to Security Manager
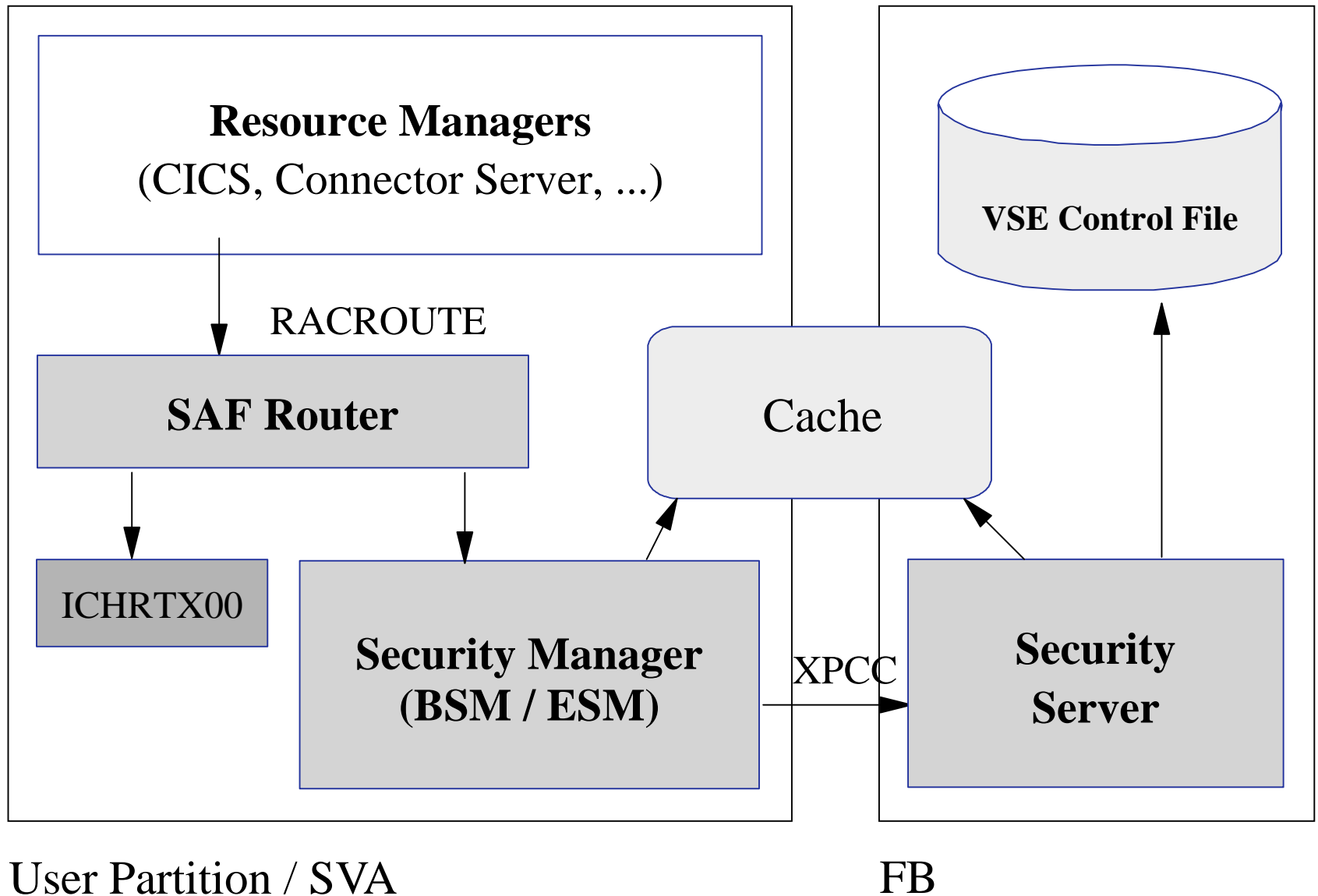 - ► Architectured interface (RACROUTE)

RACROUTE

- Architectured interface
- External interface to the Security Authorization Facility (SAF)
- To be used by Resource Managers and Subsystems
  - ► CICS TS
  - ► VSE Connector Server
  - ► DITTO/ESA for VSE
  - ► TCP/IP Security Exit
  - ► Interactive Interface Signon
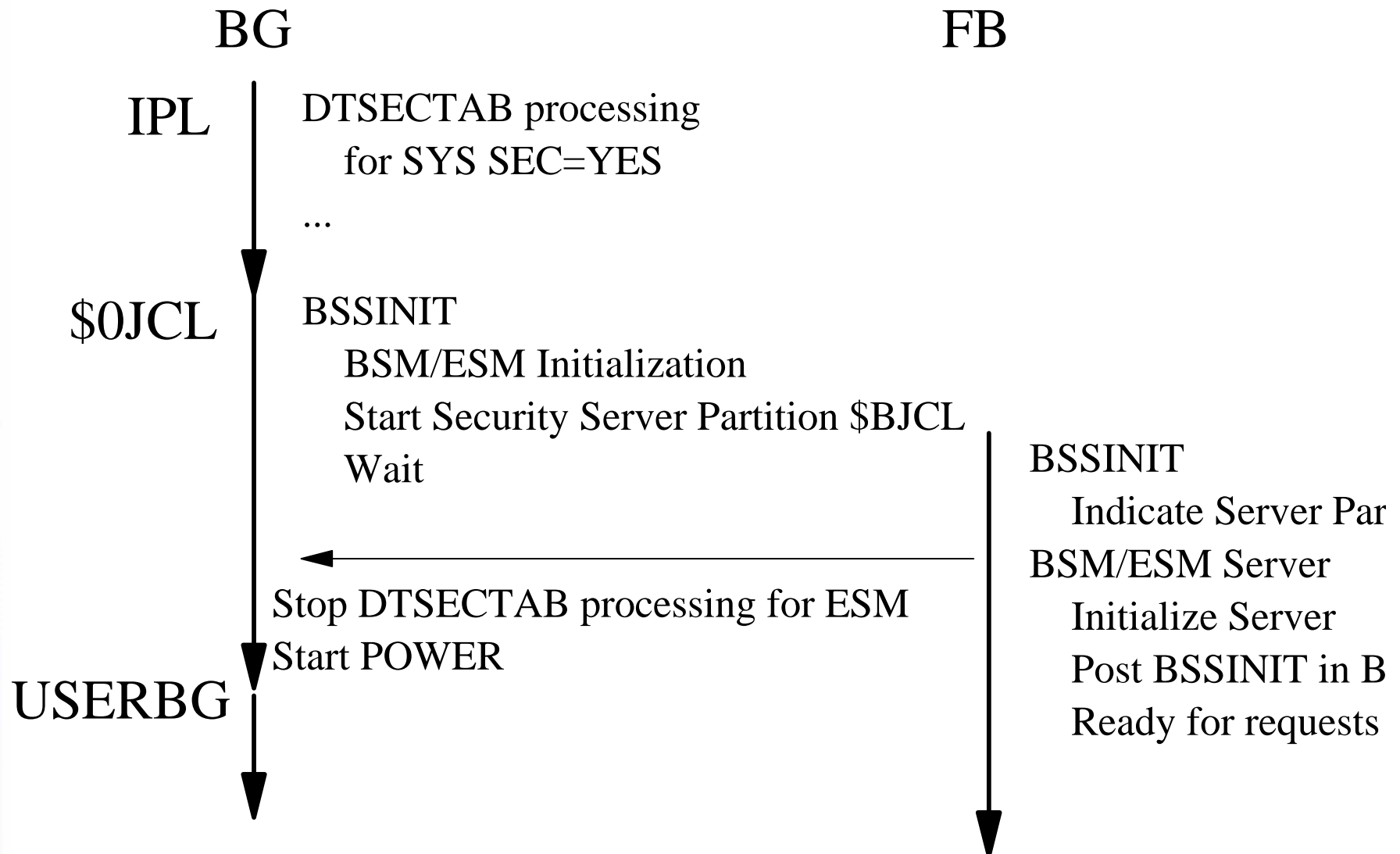
Security Authorization Facility (SAF)

Resource Managers
(CICS, Connector Server, ...)

RACROUTE

SAF Router

ICHRTX00

**Security Manager
(BSM / ESM)**

Cache

VSE Control File

XPCC

**Security
Server**

User Partition / SVA

FB

VSESecurity.prz

Common Security Startup

VS*e*

e-business

BG   FB

IPL   DTSECTAB processing
      for SYS SEC=YES

...

$0JCL   BSSINIT
      BSM/ESM Initialization
      Start Security Server Partition $BJCL
      Wait
                                BSSINIT
                                   Indicate Server Par

Stop DTSECTAB processing for ESM   BSM/ESM Server
Start POWER   Initialize Server
                                   Post BSSINIT in B

USERBG   Ready for requests

Common Security Startup (continued)

- Security manager (BSSINIT) has to initialize before other partition or POWER are active
- BSSINIT will fail, if there are other partition active
- Static partition required for Security Server
- SYS ESM=phasename in IPL proc to start ESM
- If no ESM is started, BSM is activated
- For SYS SEC=YES with ESM a DTSECTAB protection is active until ESM is initialized

VSESecurity.prz

Security Managers

- **Basic Security Manager (BSM)**
  - ► Part of VSE Central Functions
  - ► Signon Security
  - ► Transaction Security
  - ► DTSECTAB Security

- **External Security Manager (ESM)**
  - ► CA-Top Secret
  - ► BIM Alert
  - ► Vendor

VSESecurity.prz

Basic Security Manager

Resource Managers

(CICS, Connector Server, ...)

LIBR, CICS/VSE, ..   CICS TS, VCS, ...

RACROUTE

Security Server

SECHECK macro

Basic Security Manager (BSM)

DTSECTAB

DTSECTXN

VSE Control File

VSESecurity.prz
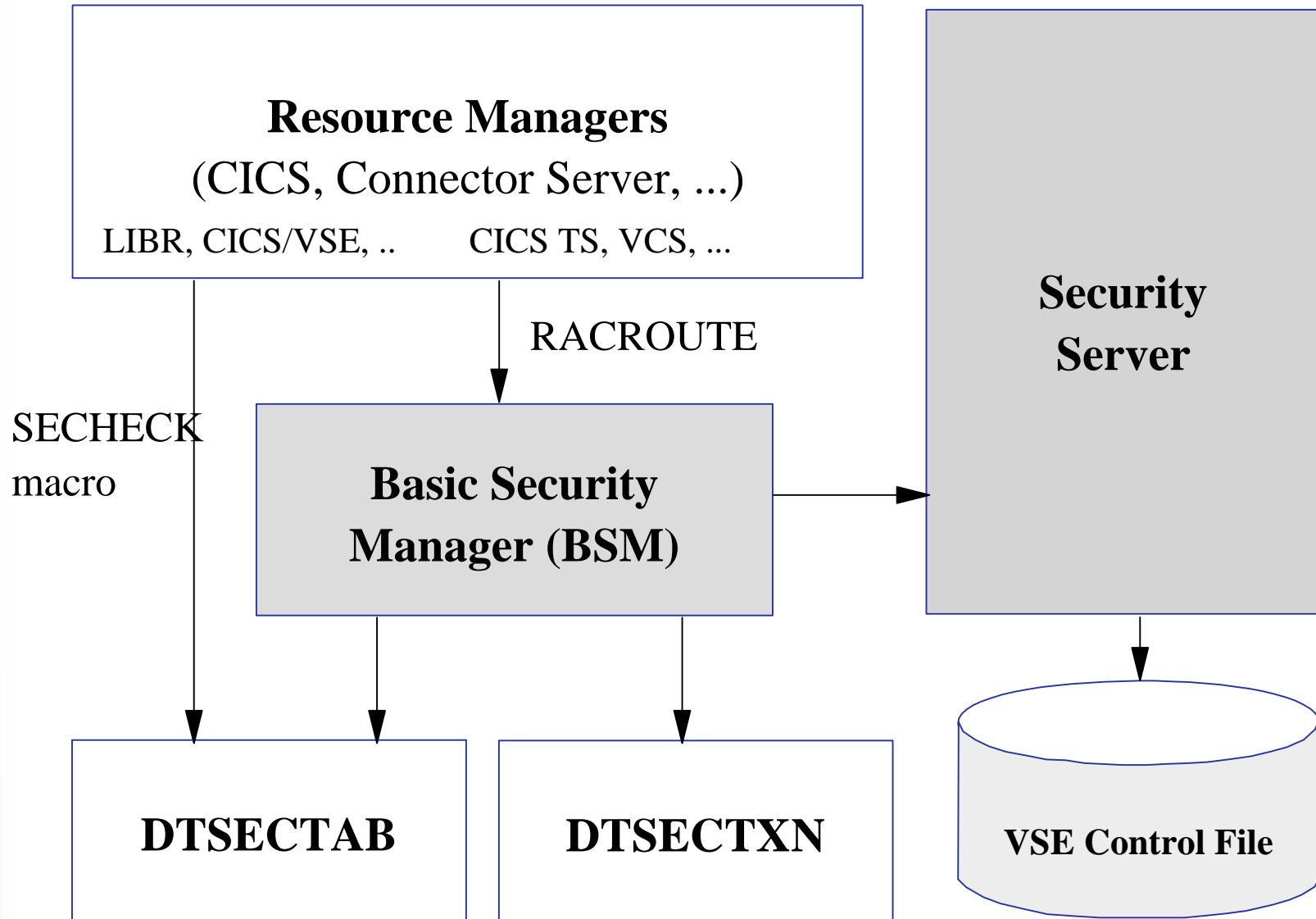
Basic Security Manager (continued)

- Provides RACROUTE support for
  - ► Signon (CICS and VSE Connector Server)
  - ► Batch signon (ID statement)
  - ► Transaction security
- Supports also the SVC-based security calls
  - ► SECHECK
- Resource classes
  - ► USER
  - ► DATASET
  - ► VSELIB, VSESLIB, VSEMEM
  - ► TCICSTRN

Basic Security Manager - Repositories

- VSE Control File (IESCNTL)
  - ►VSAM KSDS file
  - ►Contains all user profiles
  - ►used for CICS, Batch and Connector Signon
- DTSECTAB
  - ►Contains resources like files, libraries, sublibraries and members
  - ►Only 2 userids are still needed in DTSECTAB (FORSEC, DUMMY)
- DTSECTXN (new with VSE 2.4)
  - ►Transaction security profiles
  - ►Dialog (28) to define the profiles

VSESecurity.prz

Basic Security Manager - Recovery

- If an active Security Manager does not allow to recover from a problem
  - ► IPL cuu LOADPARM ..P
  - ► STOP=DPD
  - ► 0 SYS SEC=RECOVER
    - − BSSINIT will not start a Security Manager
    - − Re-IPL required to start Security Manager again

Basic Security Manager - User Profiles

- ■ VSE Control File (IESCNTL)
  - ► All Users must be defined here (SNT no longer supported by CICS TS)
  - ► VSE 2.4 (or above) Control File records are NOT compatible with previous releases
  - ► Definition
    - − User Maintenance Dialog (211)
    - − Batch utility IESUPDCF
- ■ DTSECTAB
  - ► Contains 2 userids for ASI procedure
  - ► No CICS TS user settings

VSESecurity.prz

CICS Security

- CICS/VSE uses SNT for user verification
  - ► Duplicate user definitions
  - ► SNT users can not change password

- CICS TS uses RACROUTE calls for
  - ► Signon
  - ► Resource Security
  - ► Transaction Security

VSESecurity.prz

CICS TS Signon

- Native CICS TS signon (CESN)
- VSE/Interactive Interface signon (IEGM)
- Private signon programs based on CICS SIGNON
- Signon characteristics
  - ► Inherit user identification and password verification by Security Manager
  - ► CICS TS and Interactive Interface extracts subsystem specific user settings
    - – CICS: Operator ID, Operator classes, ...
    - – II: User type, Initial panel, access flags, ...
  - ► No user definitions to subsystems necessary

CICS Security - Coexistence

- Exit program for CICS/VSE to do user verification against BSM user profiles
- DFHXSE and DFHXSSCO in PRD1.BASE
  - ► Requires RACROUTE macro from GENLIB
- Requires default user entry in SNT
- Activate ESM in CICS/VSE
  - ► EXTSEC=YES in SIT

VSESecurity.prz

CICS Security - Prefixing

- CICS Prefixing can be used to differentiate between two or more CICS TS running on the same VSE/ESA system
- CICS Prefix is identical with the userid of the CICS startup job
  - ► SECPRFX=YES in SIT
  - ► SYS SEC=YES: userid in * $$ JOB or ID statement is used
  - ► SYS SEC=NO: userid in ID statement is used
  - ► When no userid is given: FORSEC is used

VSESecurity.prz

CICS Security - Migration

- Security related resource to be migrated
  - ▶ Interactive Interface user profiles from an old VSE control file
  - ▶ ICCF user records in DTSFILE
  - ▶ CICS user profiles from a CICS/VSE signon table (SNT)
  - ▶ Transaction definitions from CICS/VSE PCT
  - ▶ For Batch security users: DTSECTAB
- VSE migration utility IESBLDUP
  - ▶ migrate user profiles
  - ▶ see VSE/ESA System Utilities

CICS Security - DTSECTXN Macro

- Macro to support CICS transaction profiles
  - ► CICS-region = userid in CICS startup job
  - ► transid = up to 4 characters
  - ► class = 1-64
    - – 1 = public transactions
    - – 64 = interactive interface transactions

DTSECTXN NAME={CICS-region.}transid,
    TRANSEC=(class)
    [,SUBTYPE={INITIAL | FINAL}]
    [,TYPE=GENERIC]

VSESecurity.prz

Batch Security

- ID statement or * $$ JOB specifies userid and password for a job
- Userid and password are verified against
  - ▶ DTSECTAB
  - ▶ Security Manager (RACROUTE)
- Subsystems (LIBR, VSAM, ...) uses this userid to verify access rights against DTSECTAB

VSESecurity.prz

Security Checklist for VSE/ESA

- SYS SEC=YES/NO
  - ► YES if batch security is required

- CICS SIT SEC=YES (!)
  - ► If NO, all users can logon without a password

- Change passwords for predefined users
  - ► POST, PROG, OPER, SYSA, ...

VSESecurity.prz

Security in a heterogeneous environment

VSE/ESA

AIX

Win

AIX

Secure remote access
- Logon security
- Connector Security
- Single Signon

Location 2

Connector Security
- Logon security
- Resource security

TCP/IP Security
- Security Exit

Single Signon

VSESecurity.prz

LINUX
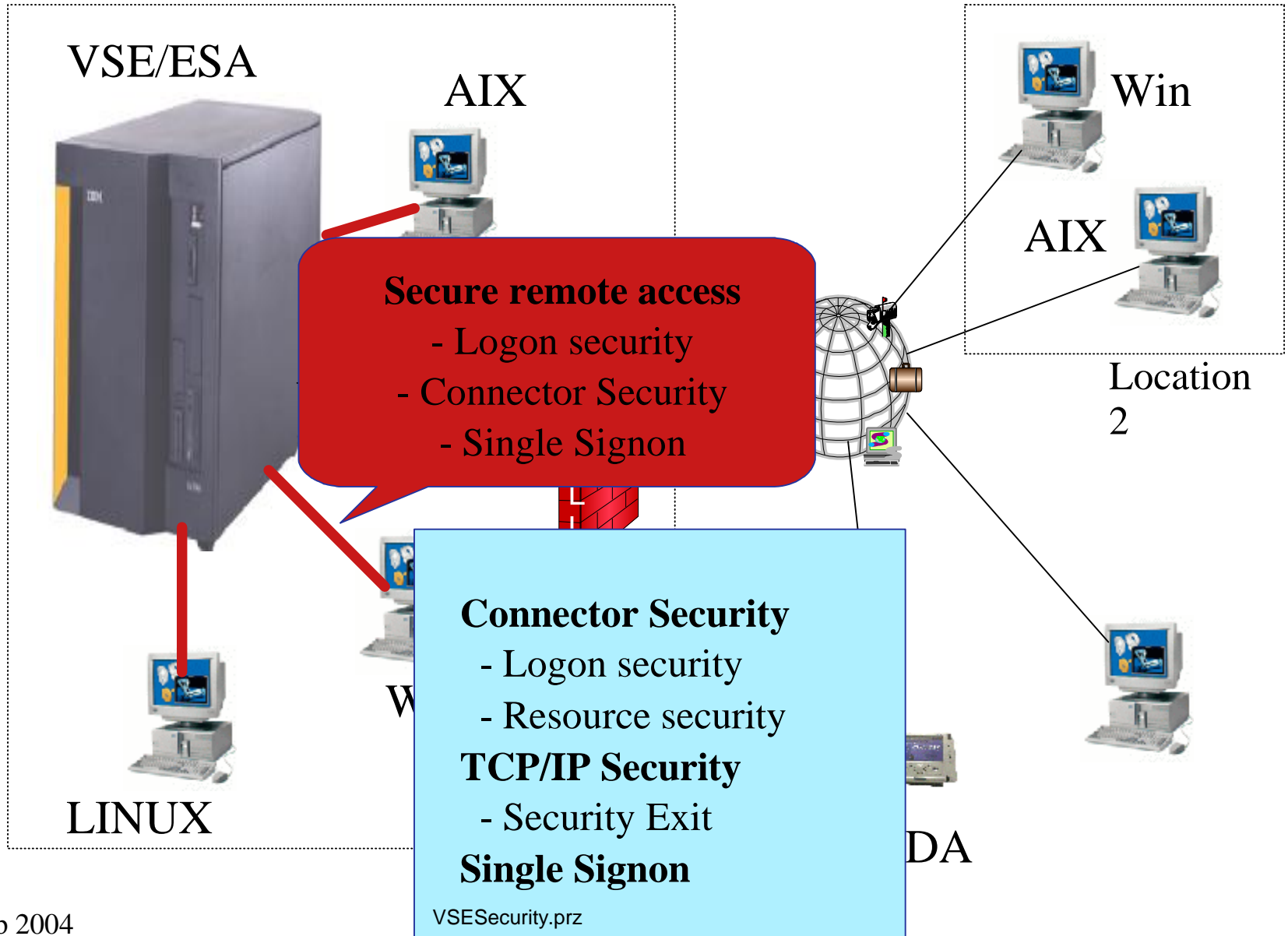
Why secure remote access ?

- Today most computers are part of a network
  - ► Can connect to your VSE/ESA system
- Prevent unauthorized access to VSE/ESA and data
  - ► Requires to authenticate the user (logon)
- FTP allowes to access production data
  - ► VSAM
  - ► POWER entries (listings)

VSESecurity.prz

Connector Security

- VSE Connector Server acts as a Resource Manager
  - ► Issues RACROUTE calls for
    - – Userid and password verification
    - – Resource security
- Connector userids are the same as for CICS TS and Batch
- No additional user profile setup required
- But:
  - ► Additional access restriction by userid and/or IP address possible

VSESecurity.prz

Connector Security - Logon

- VSE Connector Server requires a client to logon with valid userid and password
- Userid and password is checked via RACROUTE calls
- Additional information is extracted from ACEE and IUI or AF segment
  - ►User type, access flags, ...
- The user's ACEE is kept during the whole session
  - ►Used to do resource access checking
- Multiple logon attempts with same userid is possible

Connector Security - Resource Security

- When a client issues a resource access request
  - ▶ The server does RACROUTE calls to check if the user is allowed to access the resource
  - ▶ Access is done only if user is allowed to access the resource
- VSE Connector Server runs under a special userid (VCSRV)
  - ▶ specified in ID statement in startup job
  - ▶ should be allowed to access all resources

Connector Security - Internals

■ Logon processing

▶ RACROUTE VERIFY CREATE

▶ RACROUTE EXTRACT (user type checking)

  – AF segment, if this fails (e.g. CA-TopSecret) IUI segment

▶ Flags used in AF segment

  – AFADMIN   user is a administrator = type 1
  – AFMCONS   user is allowed to open a console

▶ Flags used in IUI segment

  – IESISUTP   user type (1,2 or 3)
  – IESISFL1   user flag byte 1
  – IESISFL2   user flag byte 2

Connector Security - User types

- Type 1 (Administrator)
  - ► read and write access for all resources
- Type 2 (Programmer)
  - ► read only access for all resources
  - ► allowed to submit jobs
- Type 3 (Application User)
  - ► read only access for selected resources

Connector Security - Resource classes

- The following Resource class are used
  - ▶ VSELIB, VSESLIB, VSEMEM (LIBR)
  - ▶ DATASET (VSAM)
- Resource not protected by Security Manager
  - ▶ POWER queue entries
    - protected by user type and access flag
  - ▶ Console
    - protected by user type and access flag
    - If user is allowed to access the console, he can issue all console commands,
      even REIPL NOPROMPT (!)
  - ▶ ICCF Libraries and Members
  - ▶ VSAM Record Mappings

Connector Security - Additional Security

- Configuration member allows to restrict logon (connect) by
  - ►Userid
  - ►IP address
- See skeleton SKVCSUSR in ICCF library 59

```
* ****************************************************************
* USERS FROM THIS IP'S ARE ALLOWED TO LOGON
* ****************************************************************
IP   = *,          LOGON = ALLOWED
* IP = 9.164.123.456, LOGON = DENIED
* IP = 9.165.*      , LOGON = DENIED
* IP = 10.0.0.*     , LOGON = ALLOWED
* ****************************************************************
* THIS USERS ARE ALLOWED TO LOGON
* ****************************************************************
USER = *,          LOGON = ALLOWED
* USER = BOBY,       LOGON = ALLOWED
* USER = SYS*,       LOGON = DENIED
```

VSESecurity.prz

Deactivation of Connector Security

- Since PTF UQ66736 (VSE 2.6), UQ66733 (VSE 2.5) Connector Security can be deactivated
- New keyword SECURITY in main configuration member:
  - ► SECURITY = FULL (default, as before)
  - ► SECURITY = RESOURCE (no user type checking)
  - ► SECURITY = LOGON (no resource, only logon)
  - ► SECURITY = NO (no security at all)
- Access restriction (previous foil) is still active, even if SECURITY = NO

VSESecurity.prz

TCP/IP Security

- In general TCP/IP uses its own userid definitions
  - ► DEFINE USER,ID=user,PASSWORD=pwd
  - ► Readable in initialization member (IPINITxx.L)
  - ► Duplicate user definitions
  - ► Used for
    - – FTP
- Security Exit available from IBM to check the userids and resource access via Security Manager
  - ► see next foil
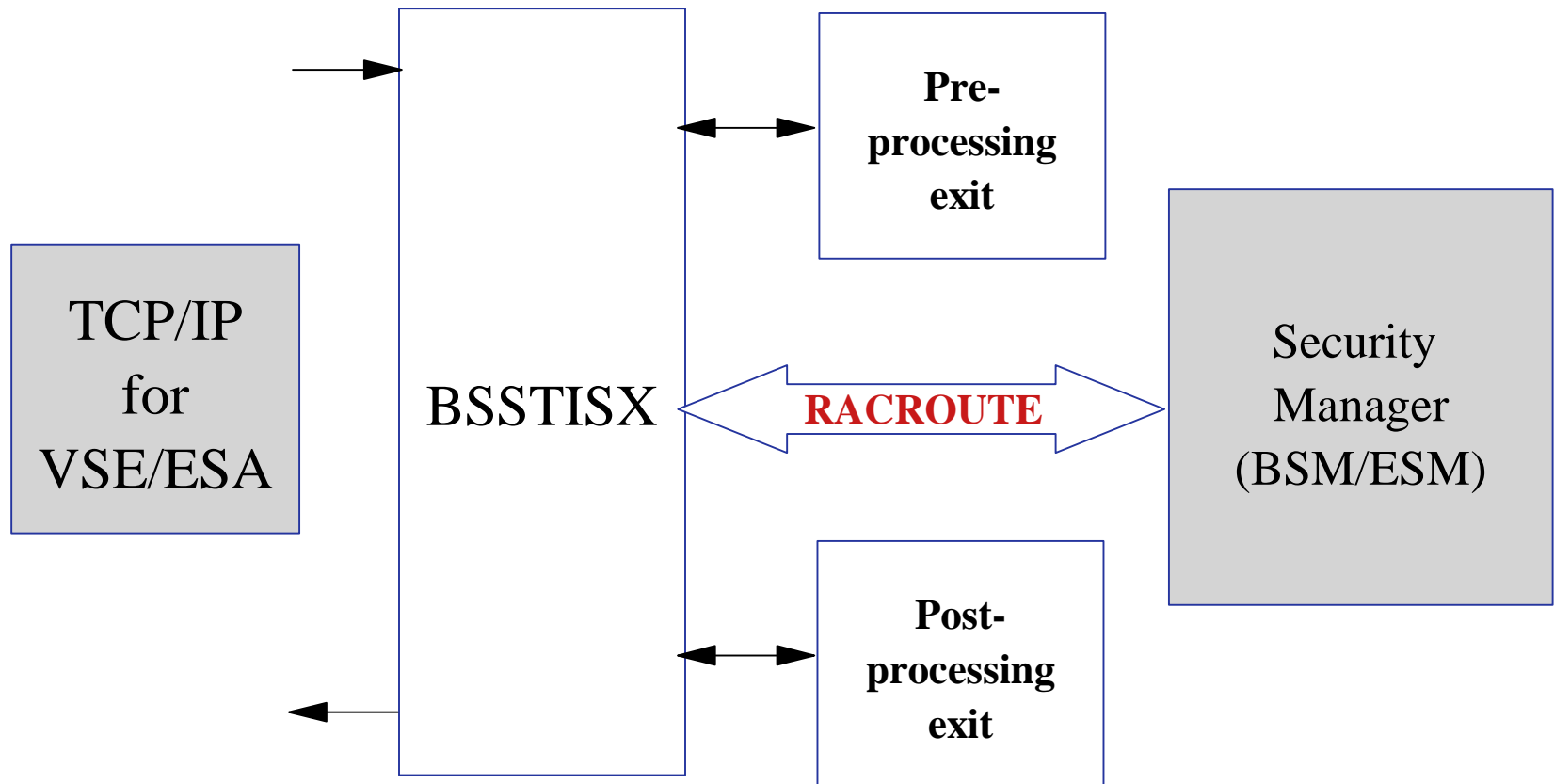
TCP/IP Security Exit

TCP/IP Security Exit

- Issues RACROUTE calls for
  - ► User identification and verification
  - ► Resource access control
    - – VSE files, libraries, members
    - – POWER entries
    - – SITE commands
- Provides a pre- and post-processing exit interface
- Activation
  - ► DEFINE SECURITY,DRIVER=BSSTISX[,DATA=data]
    - – DATA=anonym_uid,anonym_pwd,preproc,postproc, mode
  - ► SET SECURITY=ON
  - ► For details see VSE/ESA Software Newsletter #20 (First/Second Quarter, 2000)

VSESecurity.prz

TCP/IP Security - HTTPHACK.L

- Typical hacker attacks are normally no problem for VSE, only for Windows
- Rejects hacker attacks
  - ►by filtering known URL prefixes
- HTTPHACK.L:

```
* Example:
*
* "SCRIPTS/" will cover...
*       GET /SCRIPTS/ROOT.EXE?C+D
*       GET /SCRIPTS/ROOT.EXE?CAT+PASSWD
*       etc...
* =========================================================
SCRIPTS/
MSADC/
_VTI_BIN/
_MEM_BIN/
C/WINNT/SYSTEM32/CMD.EXE
D/WINNT/SYSTEM32/CMD.EXE
CGI-BIN/
```

Single Signon Solutions

- Every server/application requires you to logon
  - ► Different user ids and passwords for each server
- A single signon solution
  - ► Requires a user to signon only once
    - − one user id, one password
  - ► Stores signon information for several servers or applications
  - ► Automatically performs a signon on each server or application
    - − Using the stored signon information
- Example: LDAP

VSESecurity.prz

Security Checklist for TCP/IP

■ Connector Security
  ► Set SECURITY=FULL (SKVCSCFG)
  ► Define resource access rights (BSM/ESM)
  ► Restrict remote access to specific users and IPs (SKVCSUSR)

■ TCP/IP Security
  ► SET SECURITY=ON in IPINIT member
  ► Use Security Exit
  ► Do not define users in IPINIT member

VSESecurity.prz

Security in a heterogeneous environment

VSE/ESA

AIX

Win

F
I
R
E
W
A

Secure Internet access
- Secure Socket Layer
- Virtual Private Network (VPN)
- Cryptography

Location 2

Cryptography
- Basics
- Key Management
- Certificates

Secure Socket Layer (SSL)

Hardware Crypto

VPN, IPSec

VSESecurity.prz

LINUX
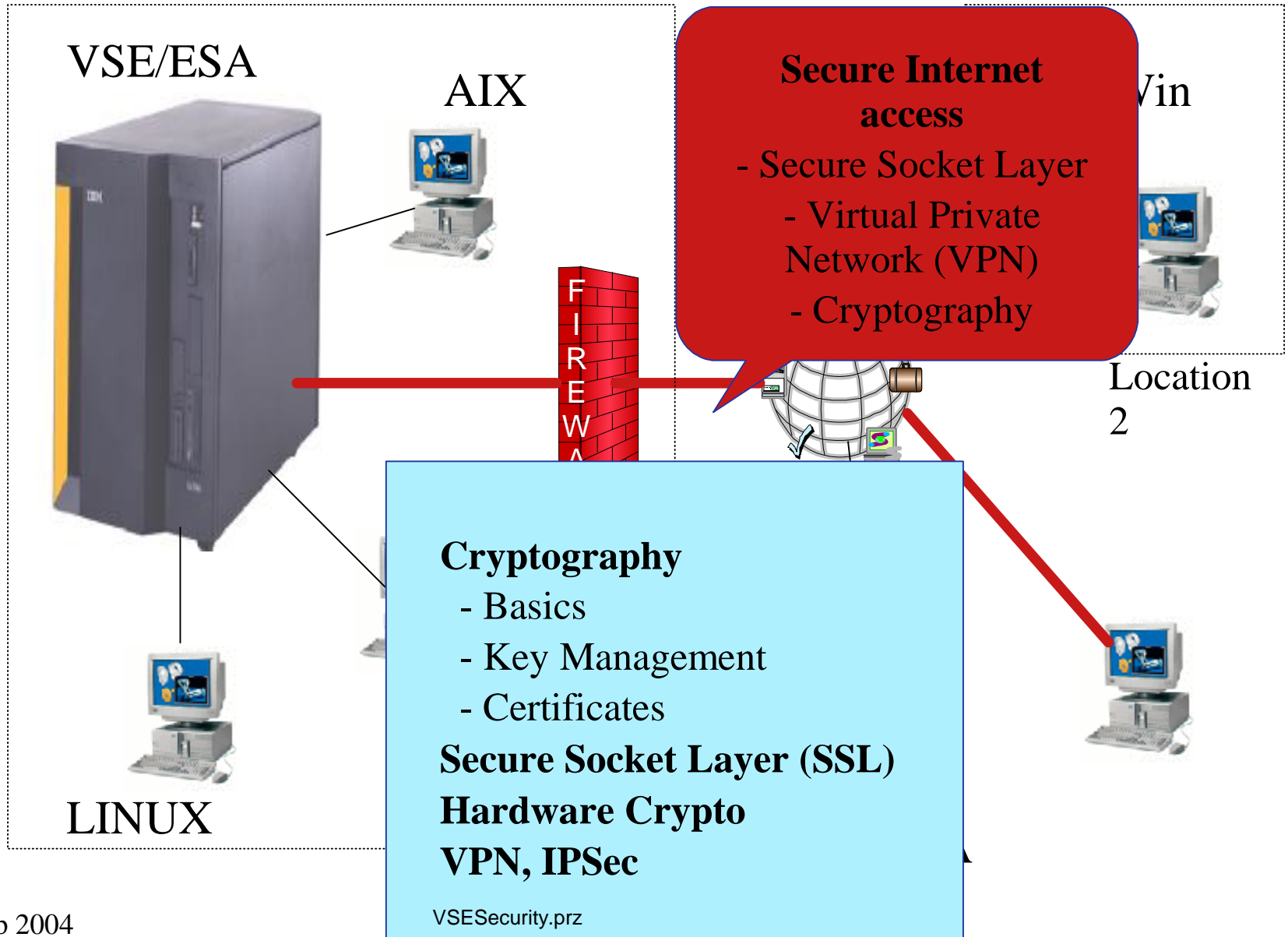
Why Cryptography ?

- **Keeping secrets**
  - ► Alice wants to send Bob confidential information, Charly should not be able to read it.
- **Proving identity**
  - ► Bob receives a message from Alice. How he can be sure that it is really from Alice?
- **Verifying information**
  - ► Bob receives a message from Alice. How he can be sure that the content has not been modified?

Keeping Secrets

Charly

Alice

Bob
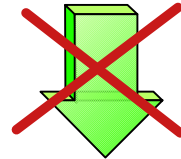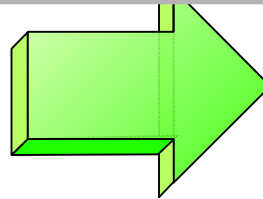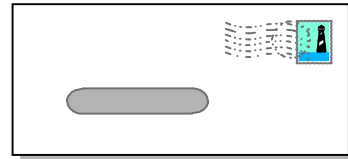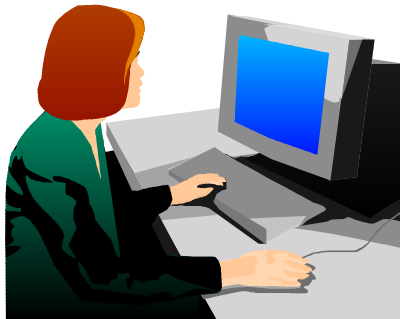
Alice encrypts the message with a secret code that only she and Bob knows

Proving Identity

Charly
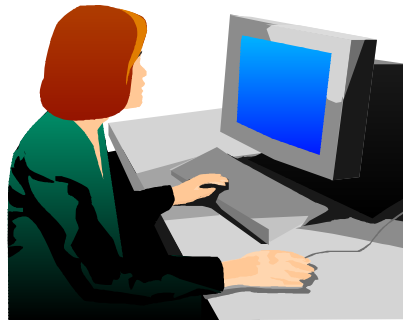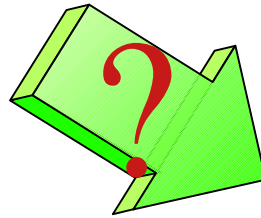
Alice
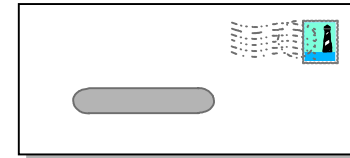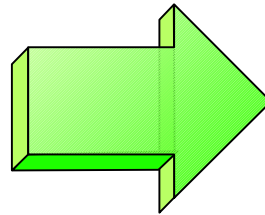
Bob

Alice "signs" the message by attaching a secret phrase that only she and Bob knows

VSESecurity.prz

Verifying Information
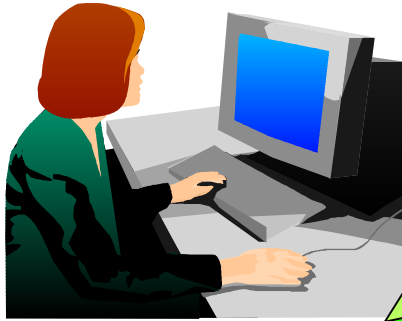
Alice
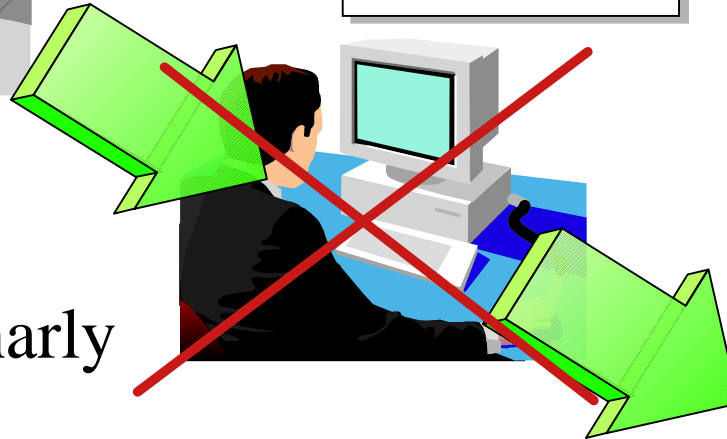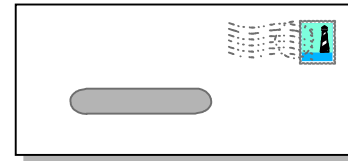
Charly
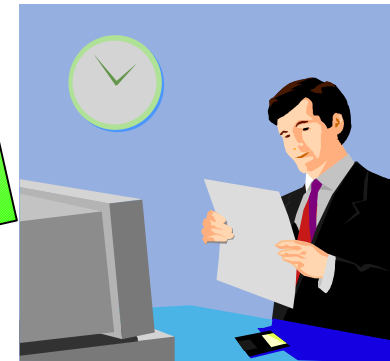
Bob

Alice generates a "hash" from the
message using a secret code and attaches
it to the message. Bob also generates the hash
from the received message and compares it.

VSESecurity.prz

Secret Key Cryptography (symmetric)

- Both parties know the same secret code (key)
- The key must be kept secret
- Encryption algorithm = mathematical transformation of the data with the key

➤ DES   Data Encryption standard

➤ 3DES   Triple strength DES

➤ RC2   Rivest Cipher 2

➤ RC4   Rivest Cipher 4

Typical key length: 40, 56 or 128 bit

Secret Key Cryptography - continued

Alice   Bob
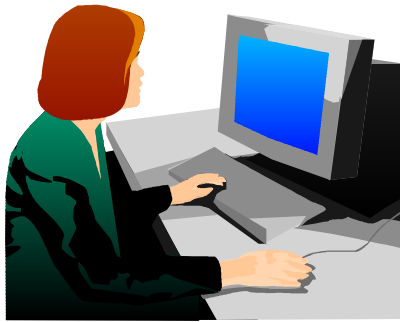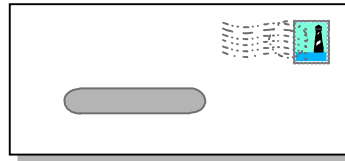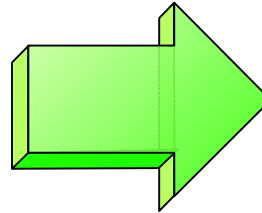
Alice encrypts the message with the secret key and sends it to Bob. Bob decrypts the message with the secret key.

Public Key Cryptography (asymmetric)

- One "public key" and one "private key"
- "Private key" is kept secret (private)
- "Public key" is published
- Asymmetric cryptography is based on mathematical problems, that are much easier to create than to solve

  ➤ RSA   Rivest Shamir Adleman
  ➤ DSA   Digital Signature Algorithm
  ➤ DHE   Diffie Hellman Algorithm

Typical key length: 512 or 1024 bit

Public Key Cryptography - Encrypting

Alice   Bob
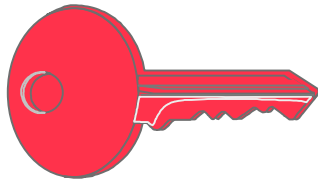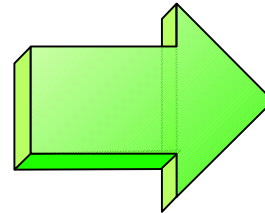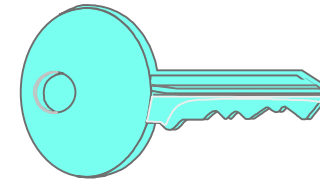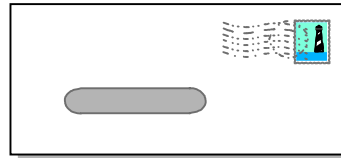
Bob's public key   Bob's private key

Alice encrypts the message using Bobs public key and ser
it to Bob. Bob decrypts it using his private key.
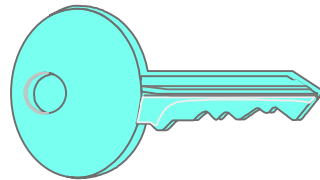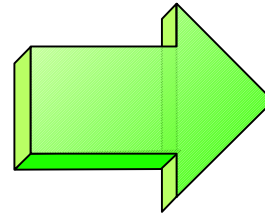Since only Bob knows his private key, only he can read
the message.

VSESecurity.prz
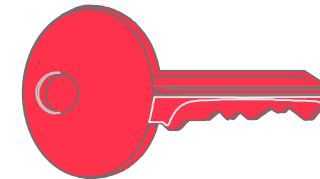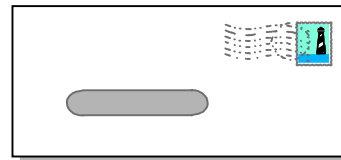
Public Key Cryptography - Signing

Alice

Bob

Alice's private key

Alice's public key

Alice encrypts the message using her private key and sends it to Bob. Bob decrypts it using Alice's public key. The message is "signed" by Alice since it can only be decrypted using **her** public key.

Combined Symmetric and Asymmetric Cryptography

- Asymmetric cryptography is very CPU-time consuming
- Use asymmetric cryptography only for secret key exchange
- Data encryption uses symmetric cryptography
- Secret key is generated by random
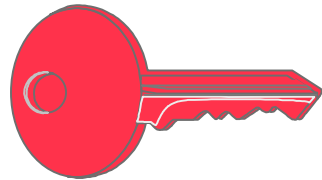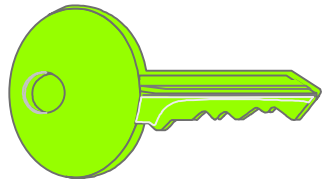- SSL also uses this mechanism

Combined Symmetric and Asymmetric Cryptography - continued

Alice

Bob

secret key

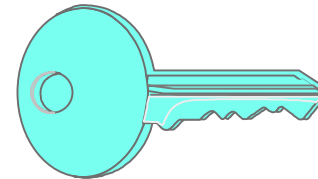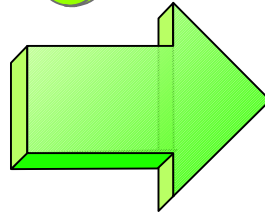Bob's public key

Bob's private key

secret key

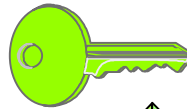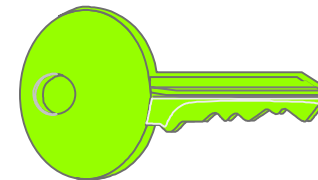secret key

VSESecurity.prz
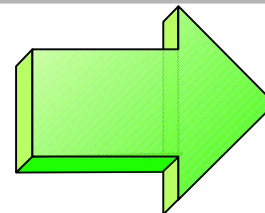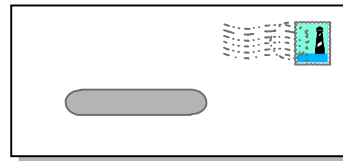
Key Management

- Key management is not trivial:
  - ► Is the public key really from the right person?

Alice

Charly

Bob
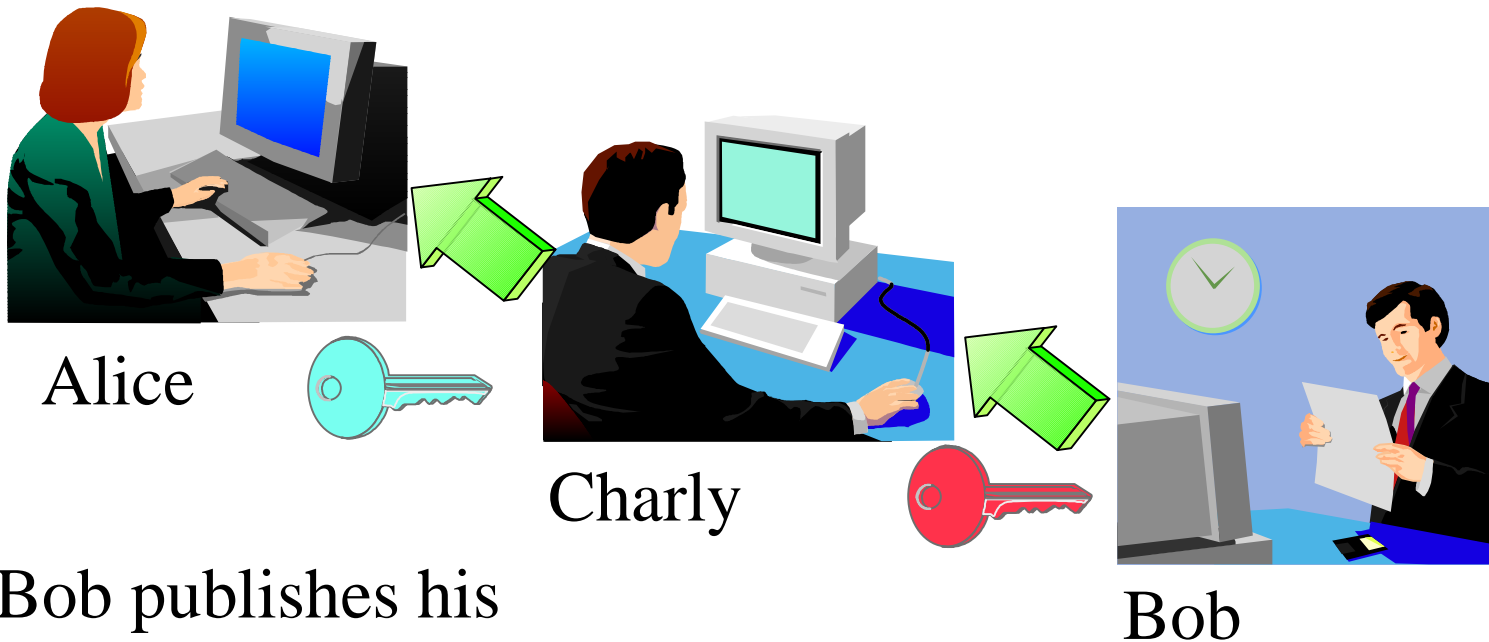
Bob publishes his
public key, but Charly intercepts
this and instead sends his public key to Alice.

VSESecurity.prz

Certificates

- A certificate contains the following items
  - ► The subject (name of the person)
  - ► The subject's public key
  - ► Period of validity
  - ► The issuer
  - ► Issuers signature
- The issuer "signs" the certificate by encrypting a hash of the certificate content with his private key
- Everyone can check the sign by decrypting it with the issuers public key

Certificate Authorities

- A certificate is issued by a certificate authority (CA)
- If a user trusts the certificate authority, he can trust the certificates issued by this CA
- CAs identify itself with a "self signed certificate":
  - ▶ The public key in the certificate is also the public key used to decrypt the signature
  - ▶ Subject and issuer are the same
- It is possible to build certificate hierarchies
- Certificate revocation lists are used to mark certificates that have been issued by error

VSESecurity.prz
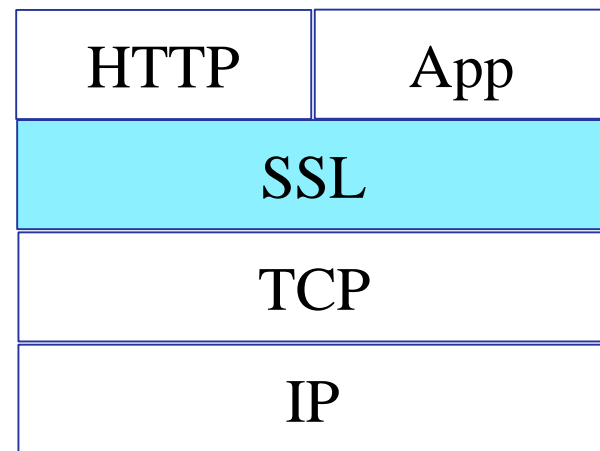
SSL (Secure Socket Layer)

- As the name implies, SSL is a layer on top of TCP
- SSL uses a TCP connection to transfer encrypted messages
- Uses asymmetric cryptography for session initiating
- Uses symmetric cryptography for data encryption

HTTP	App
TCP	
IP	

HTTP	App
SSL	
TCP	
IP	

VSESecurity.prz
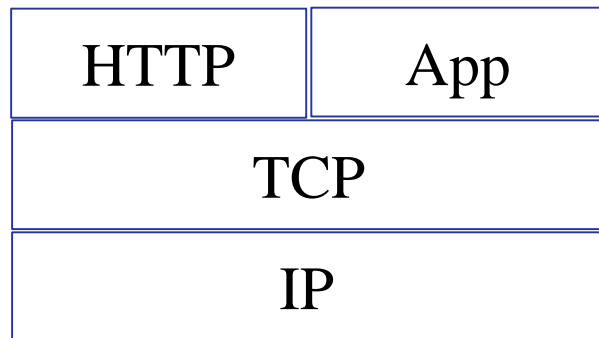
SSL Protocol

■ The SSL protocol defines a set of messages

Client   Server

ClientHello
→

ServerHello
←

ServerKeyExchange   servers public key
←   or certificate

ServerHelloDone
←

ClientKeyExchange   clients session key
→   encrypted with servers

ChangeCipherSpec   public key
→

Finished
→

ChangeCipherSpec   negoiation of options
←   (algorithms)

Finished
←
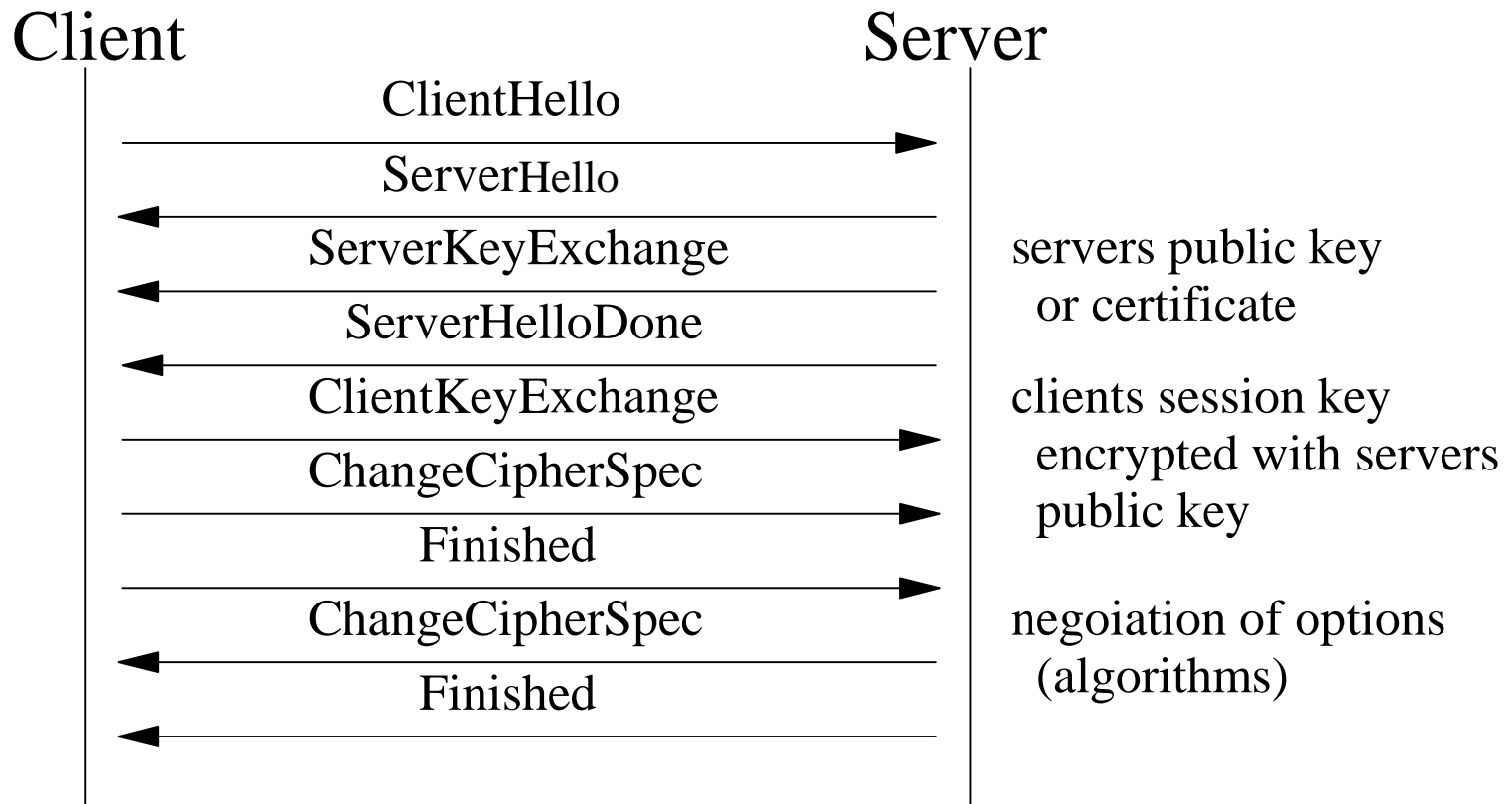
Cipher Suites

- Cipher suites defines the algorithms used:
  - ► For key exchange
  - ► For encryption
  - ► For hash algorithm

SSL_RSA_WITH_DES_CBC_SHA

Key exchange

Encryption

Hash algorithm

Server / Client Authentication

- Server authentication means:
  - ► The server sends his certificate to the client
  - ► Client checks the certificate (using the signature)
  - ► Client does not authenticate itself
- Client authentication means:
  - ► The client sends his certificate to the server
  - ► Server checks the certificate
  - ► Optionally associates access rights or privileges

Session Caching

- "SSL Session" means
  - ► Secret key used for data encryption
  - ► Negoiated algorithms
- Establishing a SSL Session is a complex and time consuming mechanism
- Session caching allows to reuse previously negoated SSL parameters
- No need of repeating the negoations or authentications
  - ► The same symmetric key is used
- The connection becomes more unsecured
- A SSL Session time-out defines how long a session is kept alive

VSESecurity.prz

SSL for VSE/ESA

- SSL for VSE is part of the TCP/IP for VSE/ESA base
- Enabled with the Application Pak
- Integrated into TCP/IP for VSE/ESA
- Supports SSL 3.0 and TLS 1.0
- Key exchange: RSA
- Data Encryption: DES and Triple DES
- Hash algorithm: MD5, SHA
- Supports X.509v3 PKI Certificates
- SSL daemon implementation for HTTPS, Telnet
- SSL API compatible with the OS/390 SSL API

VSESecurity.prz

SSL Daemon (SSLD)

- Define a SSL daemon for each TCP port that you want to secure:

  - ► DEFINE TLSD,ID=MYSSLD,

PORT=443,	HTTPS port
PASSPORT=443,	
CIPHER=0A096208,	Cipher suites
CERTLIB=CRYPTO,	library name
CERTSUB=KEYRING,	sublibrary name
CERTMEM=MYKEY,	member name
TYPE=1,	server application
MINVERS=0300,	SSL 3.0
DRIVER=SSLD	Driver phase name

VSESecurity.prz

VS*e*

e-business

Secure Socket Layer API

- Compatible to OS/390 SSL API
- Functions available for
  - ▶ Session initiating
  - ▶ Sending/receiving data
  - ▶ Ending a session
- SSL API is based on Socket API
- SSL API can be called from
  - ▶ LE-C programs
  - ▶ Assembler programs

VSESecurity.prz

CryptoVSE API

- Native cryptographic API (not available though LE)
- Provides cryptographic services:
  - ► Data encryption
    - − DES
    - − Triple DES
    - − RSA PKCS #1
  - ► Message Digest
    - − MD5
    - − SHA-1
  - ► Digital Signatures
    - − RSA PKCS #1 with SHA1 or MD5
  - ► Message Authentication
    - − HMAC

VSESecurity.prz

Hardware Crypto Overview

- Requires VSE/ESA 2.7 and TCP/IP for VSE/ESA 1.5
- Supported crypto cards
  - ▶ PCI Cryptographic Accelerator (PCICA)
    - – Feature code 0862
    - – Available for zSeries (z800, z900)
- The crypty card is plugged into the Adjunct Processor
- Currently only RSA (asymmetric) is supported
  - ▶ Of benefit for Session initiation (SSL-Handshake)
- Also supported with
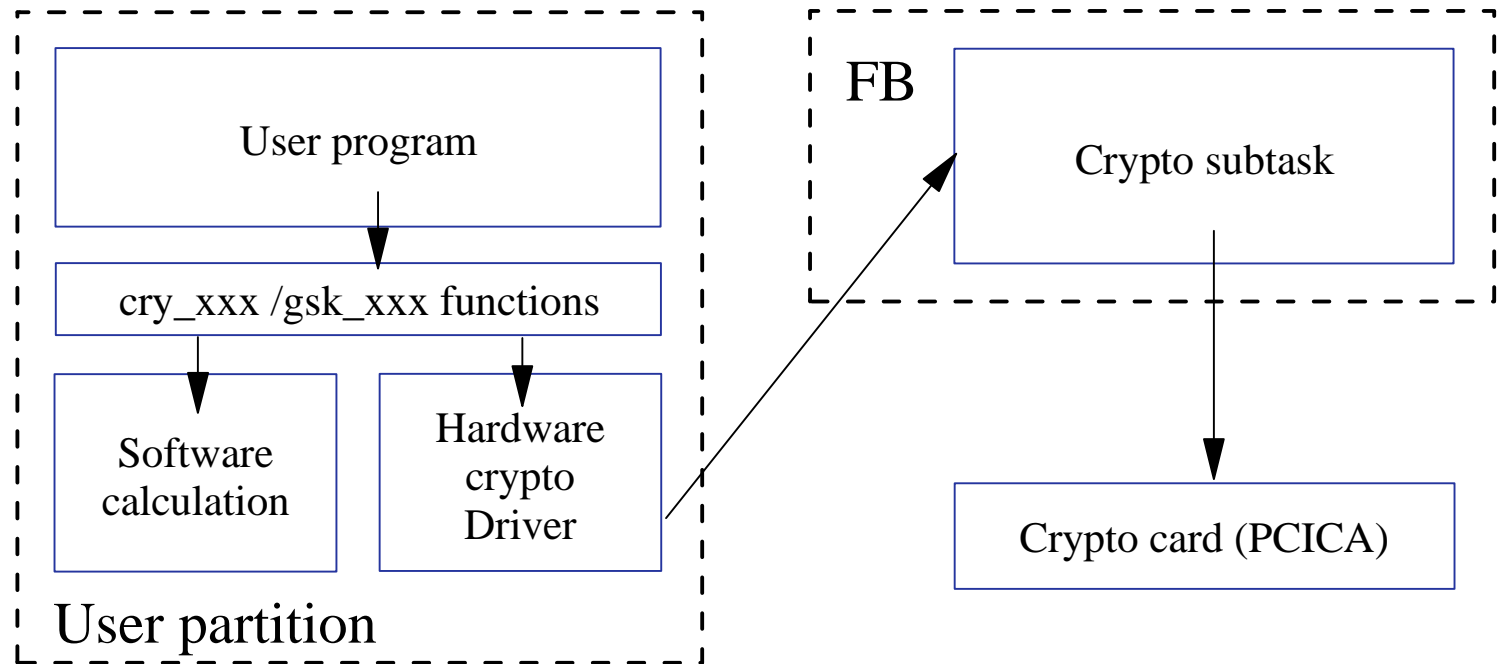 - ▶ z/VM 4.2 + APAR VM62905
  - ▶ z/VM 4.3

VSESecurity.prz

Hardware Crypto Overview - continued

- New crypto subtask in Security Server (SECSERV) running in FB
  - ► Or as separate job if no SECSERV is running
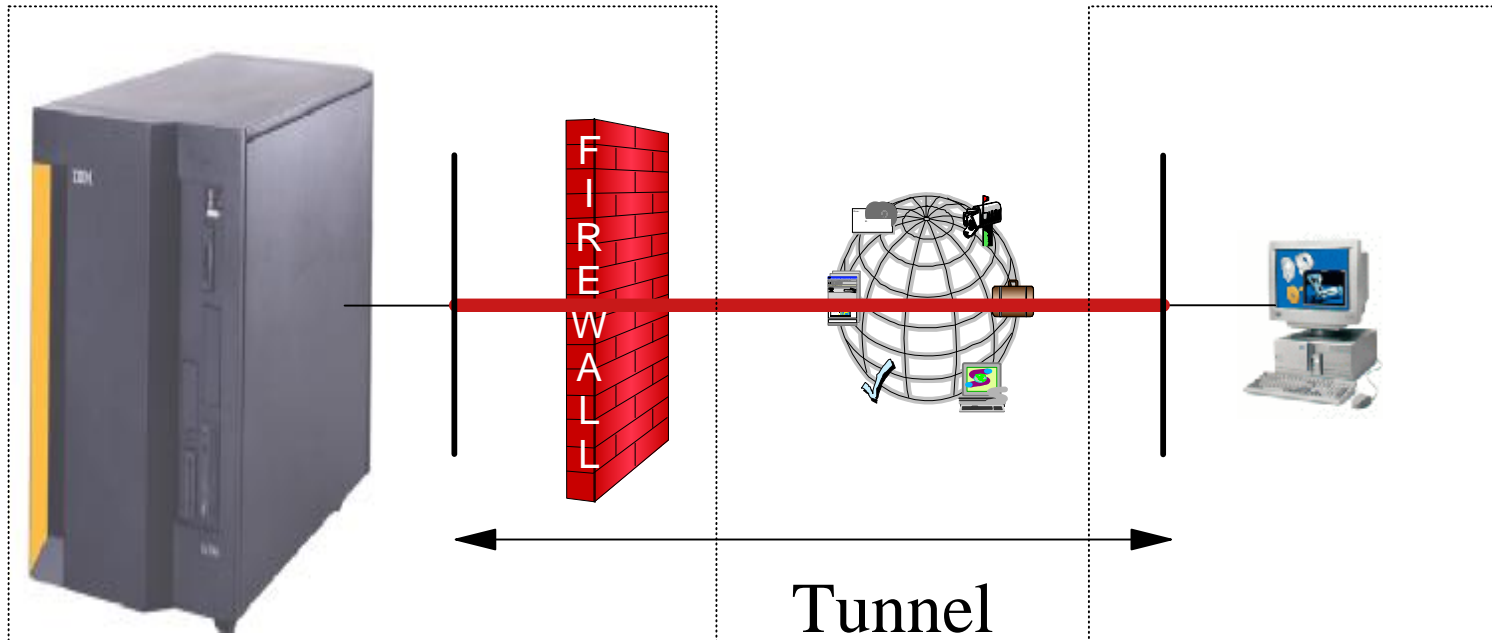 - ► Crypto card is polled by crypto task

User program
cry_xxx /gsk_xxx functions

Software calculation	Hardware crypto Driver

User partition

FB

Crypto subtask

Crypto card (PCICA)

Virtual Private Network, IPSec

- Can be used to define a secure tunnel between two locations
  - ▶ Makes use of cryptographic functions

Tunnel

Security Checklist for SSL / Crypto

- Secure Socket Layer
  - ► Use SSL if you have a need for
    - Keeping secrets
    - Proving identity
    - Verifying information
  - ► Use suitable cipher suite
    - E.g. RSA512_NULL_MD5: No data encryption
  - ► Use Hardware Crypto if available

- VPN, IPSec
  - ► Use if you dial in from outside the company
  - ► Encrypt data before sending through the Internet

VSESecurity.prz

Related Documentation

- **New:** VSE Security Homepage:
  http://www-1.ibm.com/servers/eserver/zseries/os/vse/support/vsesec.htm
- VSE/ESA Planning (SC33-6703)
- VSE Administration (SC33-6705)
- VSE/ESA Software Newsletter No. 17 (G225-4508-17)
- VSE/ESA Software Newsletter No. 18 (G225-4508-18)
- VSE/ESA Software Newsletter No. 20 (G225-4508-20)
- OS/390 Security Server External Security Interface (RACROUTE) Macro Reference (GC28-1922)
- OS/390 Security Server (RACF) Data Areas (SY27-2640)

Questions

VS_e_

e-business

IBM

VSESecurity.prz