



IBM eServer zSeries

VSE Security Concepts

ON DEMAND BUSINESS™





rademarks

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

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

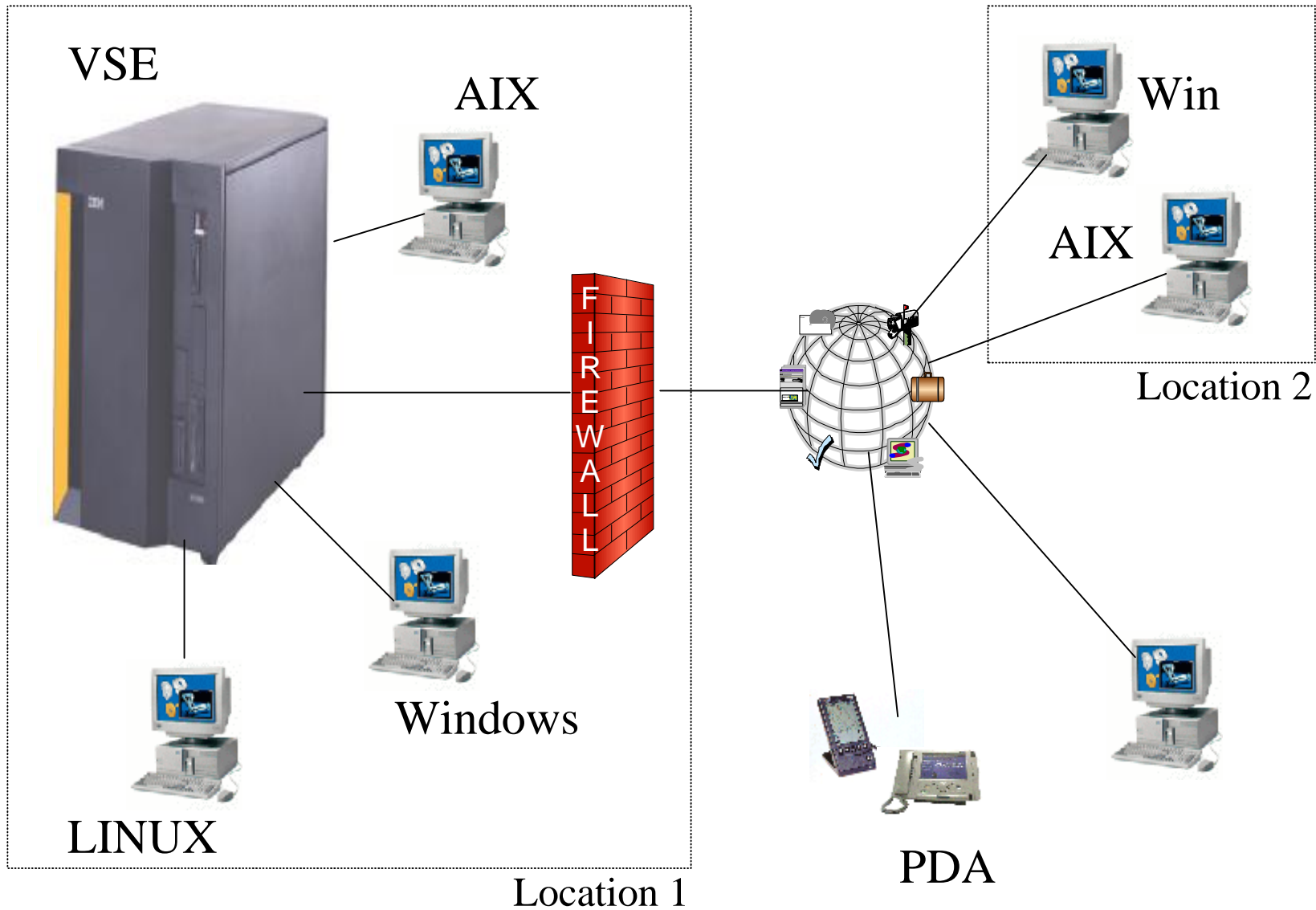
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

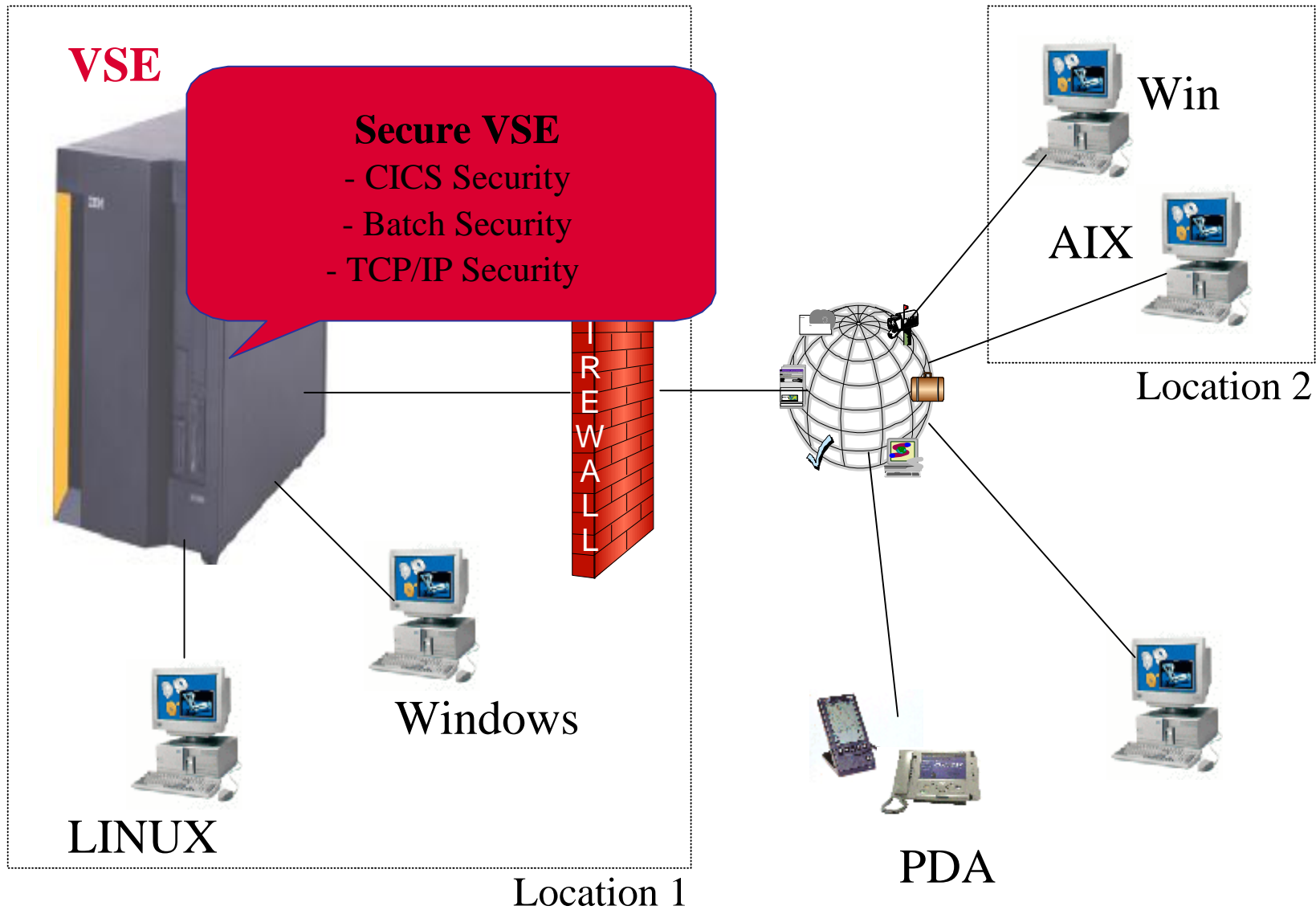
Intel is a registered trademark of Intel Corporation.



Security in a heterogeneous environment

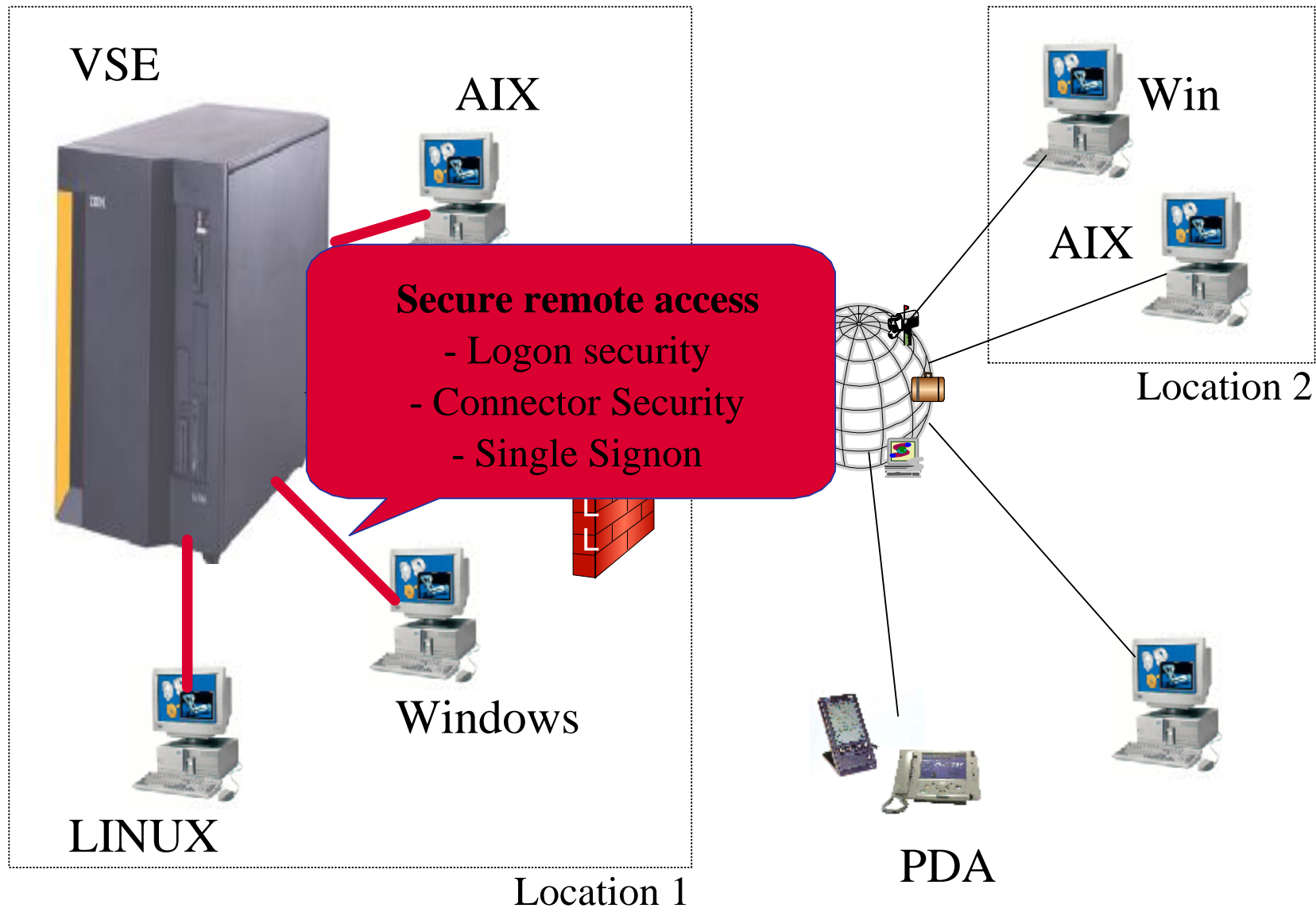


Security in a heterogeneous environment

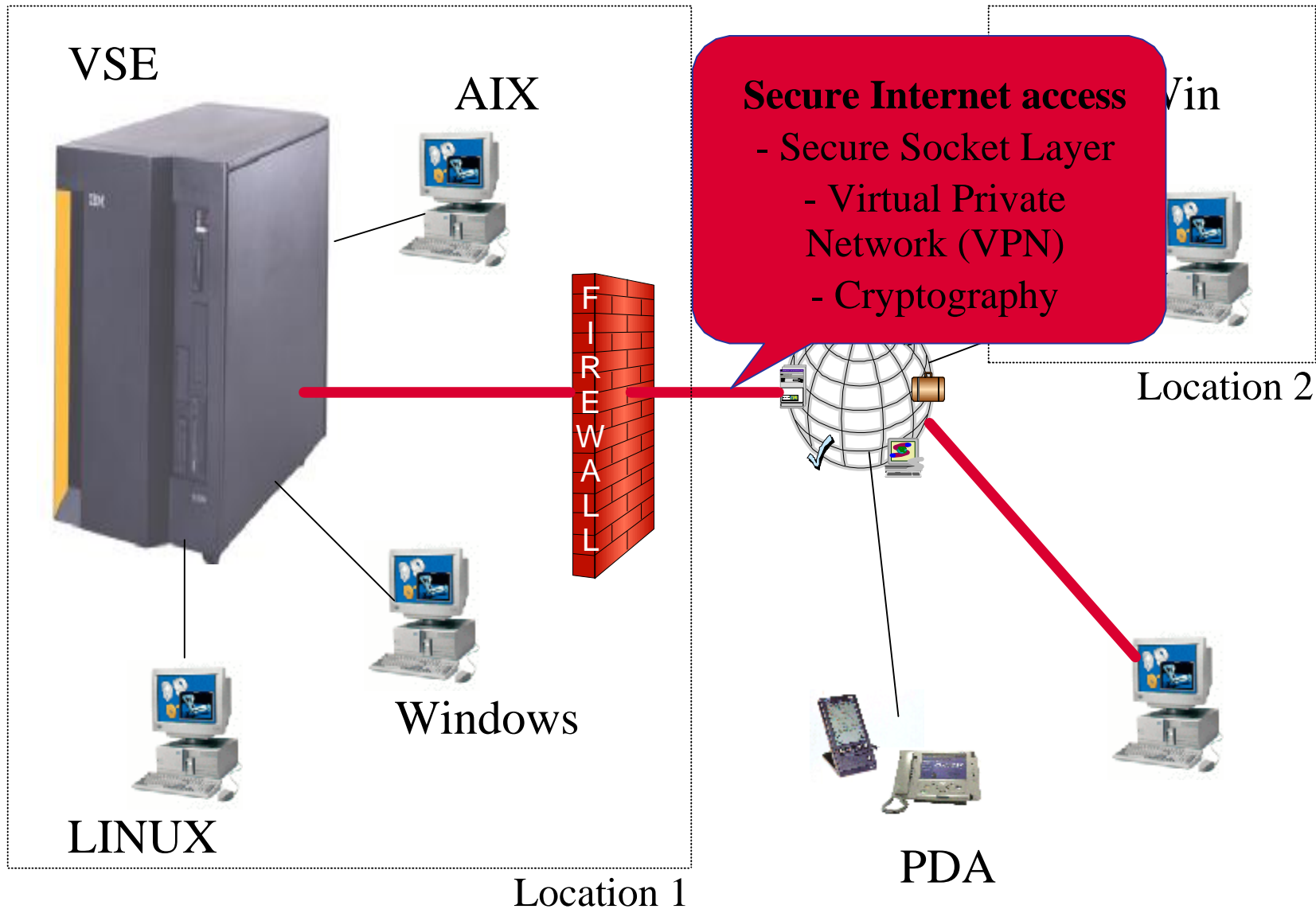




Security in a heterogeneous environment



Security in a heterogeneous environment





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

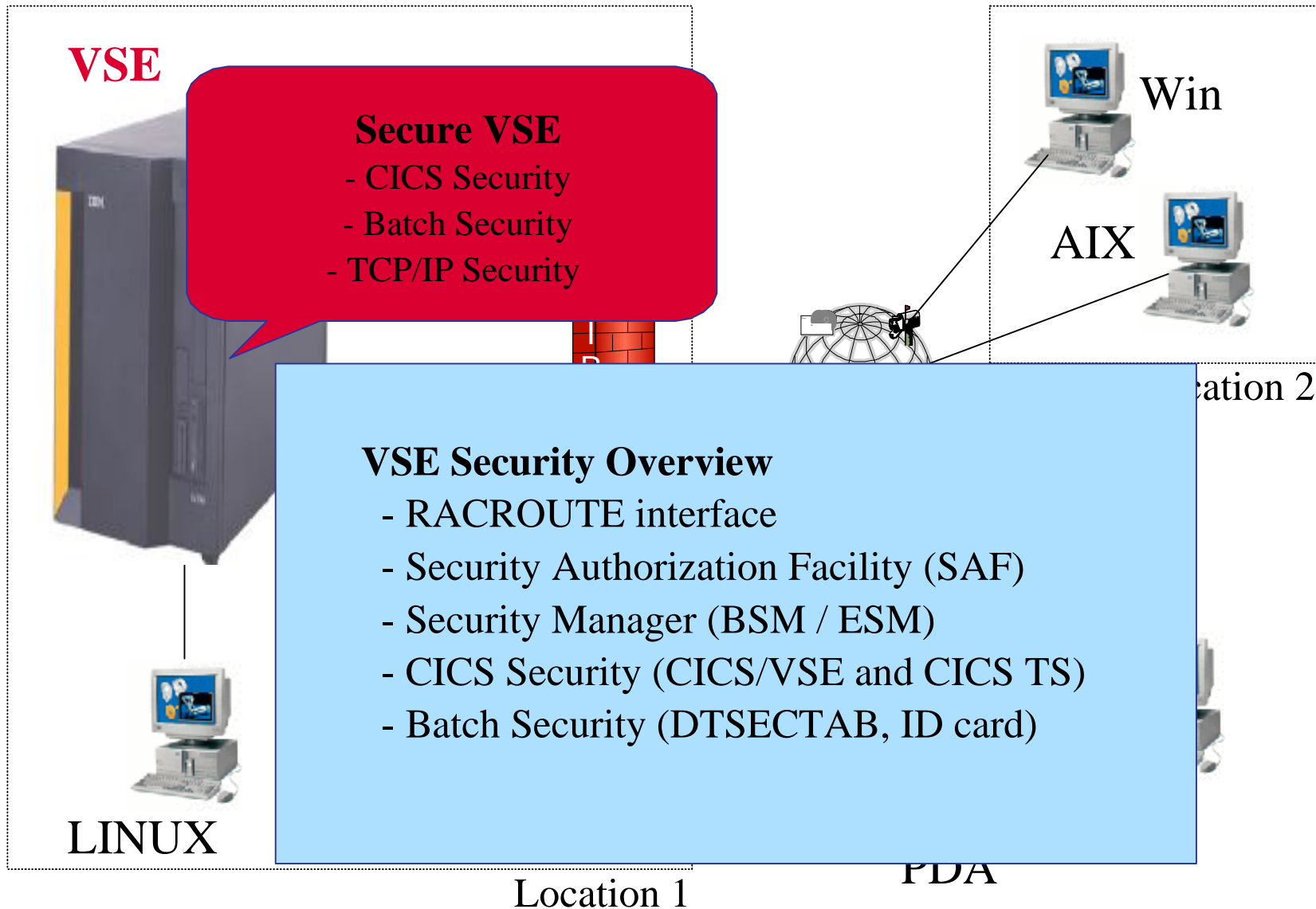
- In an heterogeneous environment
- A lot of different servers and technologies

§ You must know what you are doing !

- Incomplete security setup can be more dangerous
- than NO security



Security in a heterogeneous environment





Why secure VSE ?

§ Prevent unauthorized access to VSE and data

- Keep secret data secret
- Data modification by unauthorized users

§ Prevent users from damaging the VSE system (maybe by accident)

- Deletion of members or entries
- Submission of jobs



SE Security Overview

§ VSE/ESA 2.3 (or below)

- SECHECK macro (DTSECTAB)
- CICS/VSE internal security

§ VSE/ESA 2.4-2.7, z/VSE 3.1

- RACROUTE calls
- Security Server (BSM/ESM)
- Security decisions delegated to Security Manager
- Architected interface (RACROUTE)





ACROUTE

§ Architected interface

§ External interface to the Security Authorization Facility (SAF)

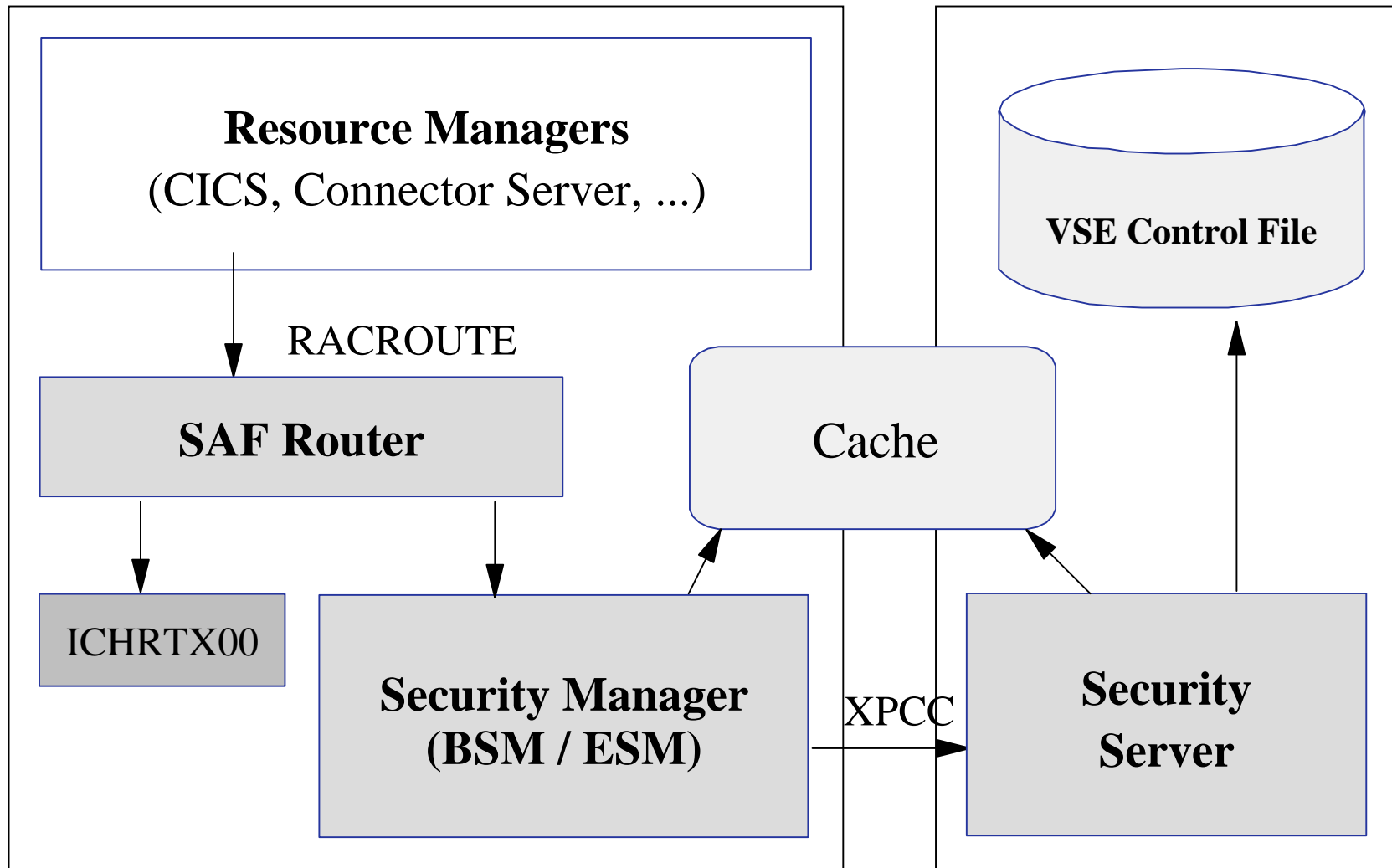
§ To be used by Resource Managers and Subsystems

- CICS TS
- VSE Connector Server
- DITTO/ESA for VSE
- TCP/IP Security Exit
- Interactive Interface Signon





Security Authorization Facility (SAF)



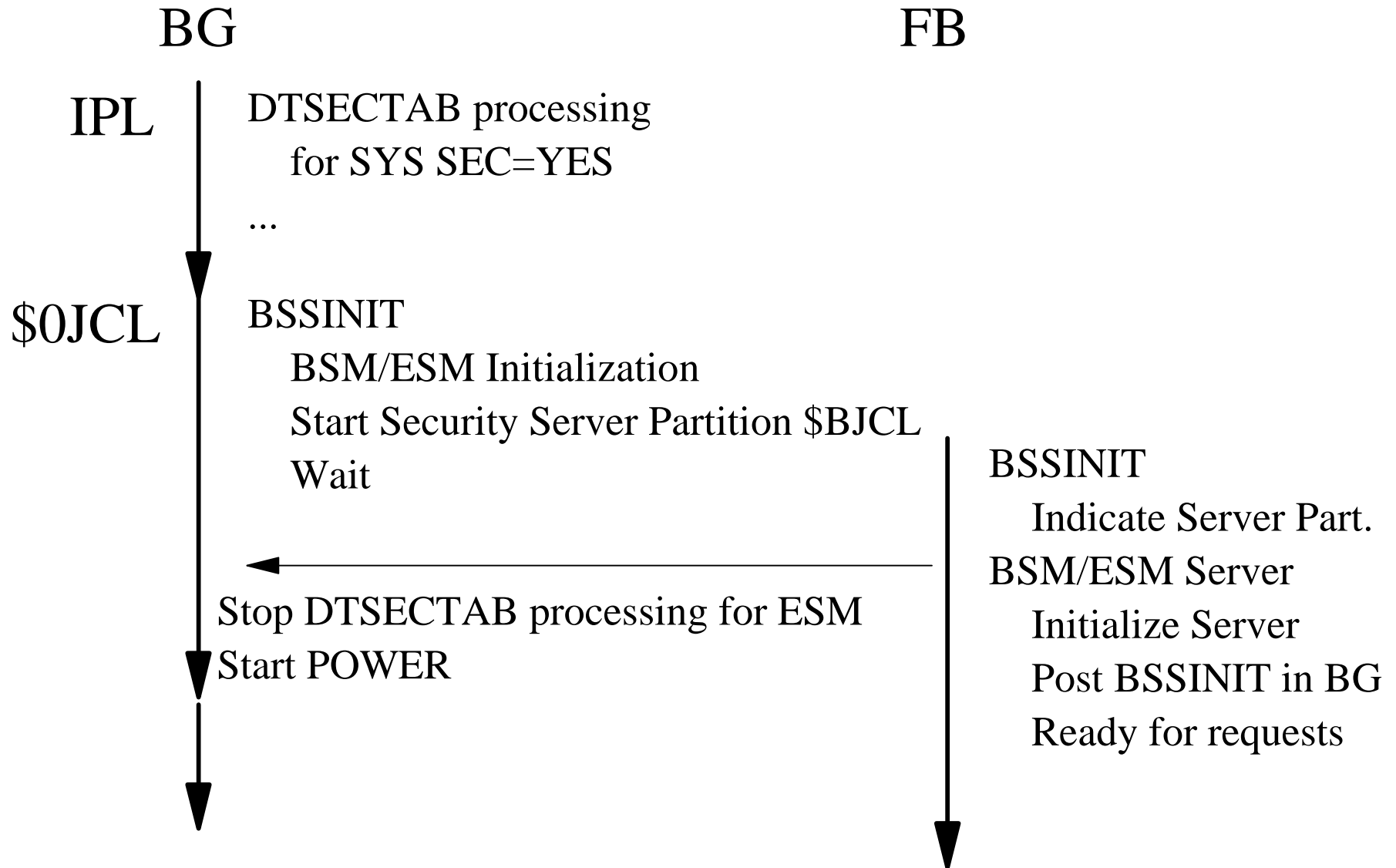
User Partition / SVA

FB





Common Security Startup





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**





Security Managers

§ Basic Security Manager (BSM)

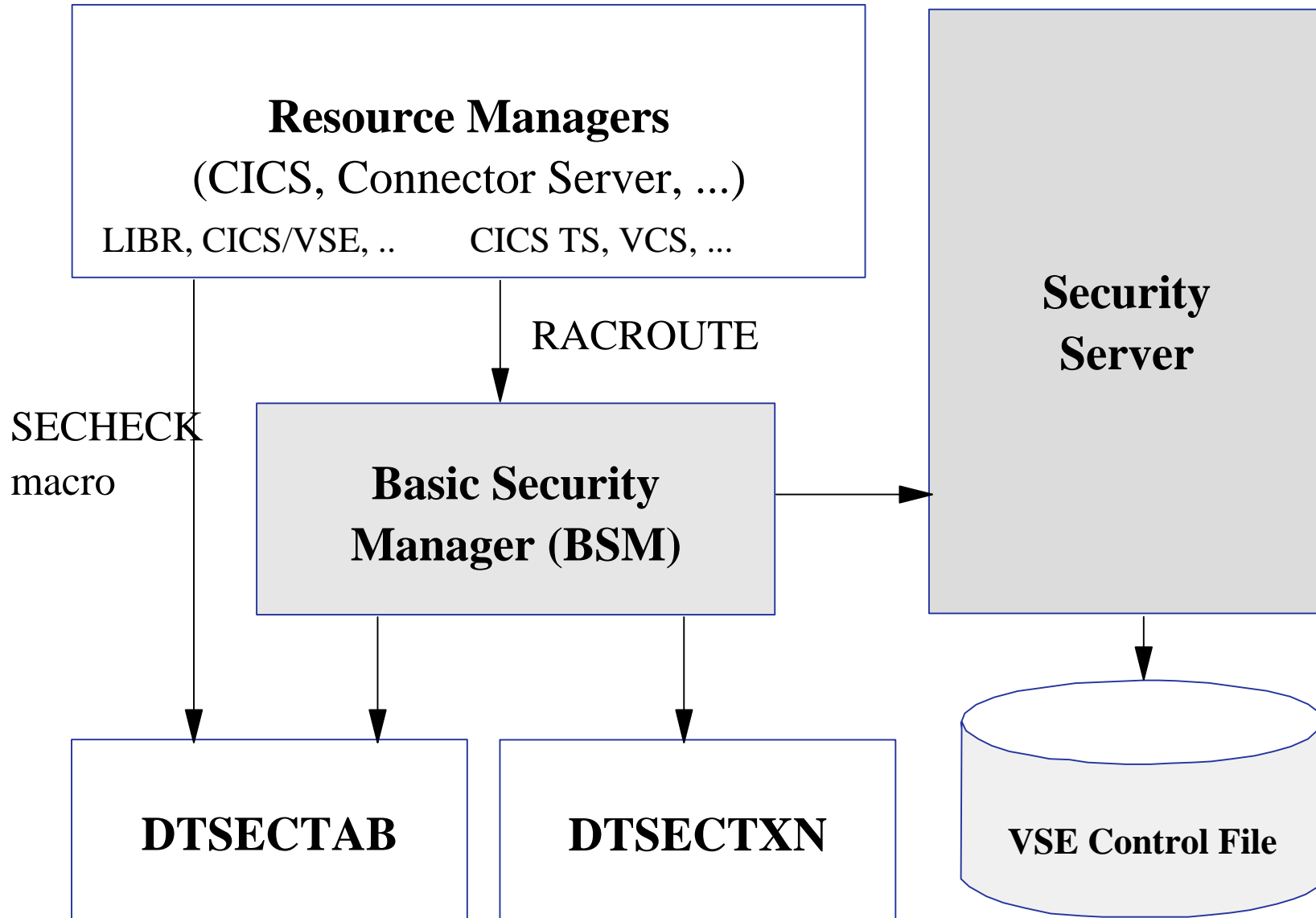
- Part of VSE Central Functions
- Sign on Security
- Transaction Security
- DTSECTAB Security

§ External Security Manager (ESM)

- CA-Top Secret
- BIM Alert
- Vendor



Basic Security Manager





Basic Security Manager (continued)

§ Provides RACROUTE support for

- Sign on (CICS and VSE Connector Server)
- Batch sign on (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 Sign on

§ DTSECTAB

- Contains resources like files, libraries, sub libraries and members
- Only 2 user ids are still needed in DTSECTAB
- (FORSEC, DUMMY)

§ DTSECTXN (new with VSE 2.4)

- Transaction security profiles
- Dialog (28) to define the profiles



Basic Security Manager - Recovery

§ If an active Security Manager does not allow to recover from a problem

- IPL cuu LOADPARAM ..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/ESA 2.4 (or above) Control File records are NOT compatible with previous releases
- Definition
 - User Maintenance Dialog (211)
 - Batch utility IESUPDCF

§ DTSECTAB

- Contains 2 user ids for ASI procedure
- No CICS TS user settings





CICS Security

§ CICS/VSE uses SNT for user verification

- Duplicate user definitions
- SNT users can not change password

§ CICS TS uses RACROUTE calls for

- Sign on
- Resource Security
- Transaction Security





CICS TS Sign on

§ Native CICS TS sign on (CESN)

§ VSE/Interactive Interface sign on (IEGM)

§ Private sign on programs based on CICS SIGNON

§ Sign on 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



ICS TS Sign on - password length

§ New with VSE/ESA 2.7: Minimum password length can be changed

- include the following in USERBG

```
// EXEC IESIRCVT  
  PASSWORD(LENGTH(5))  
  PASSWORD(WARNING(3))  
  PASSWORD(REVOKE(4))  
/*
```

- LENGTH: minimum password length of password
- WARNING: number of days a warning is displayed before password is expired
- REVOKE: number of unsuccessful sign-on attempts before user id is revoked

§ The password-revoke specification via IESIRCVT "wins" against a specification via IESELOGO



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





CICS Security - Prefixing

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





CICS Security - Migration

§ Security related resource to be migrated

- Interactive Interface user profiles from an old VSE control file
- ICCF user records in DTSTFILE
- CICS user profiles from a CICS/VSE sign on table (SNT)
- Transaction definitions from CICS/VSE PCT
- For Batch security users: DTSECTAB
- VSE migration utility IESBLDUP
- migrate user profiles

§ see VSE System Utilities manual





CICS Security - DTSECTXN Macro

§ Macro to support CICS transaction profiles

- CICS-region = user id 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]
```





atch Security

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





Security Checklist for VSE

§ **SYS SEC=YES/NO**

- YES if batch security is required

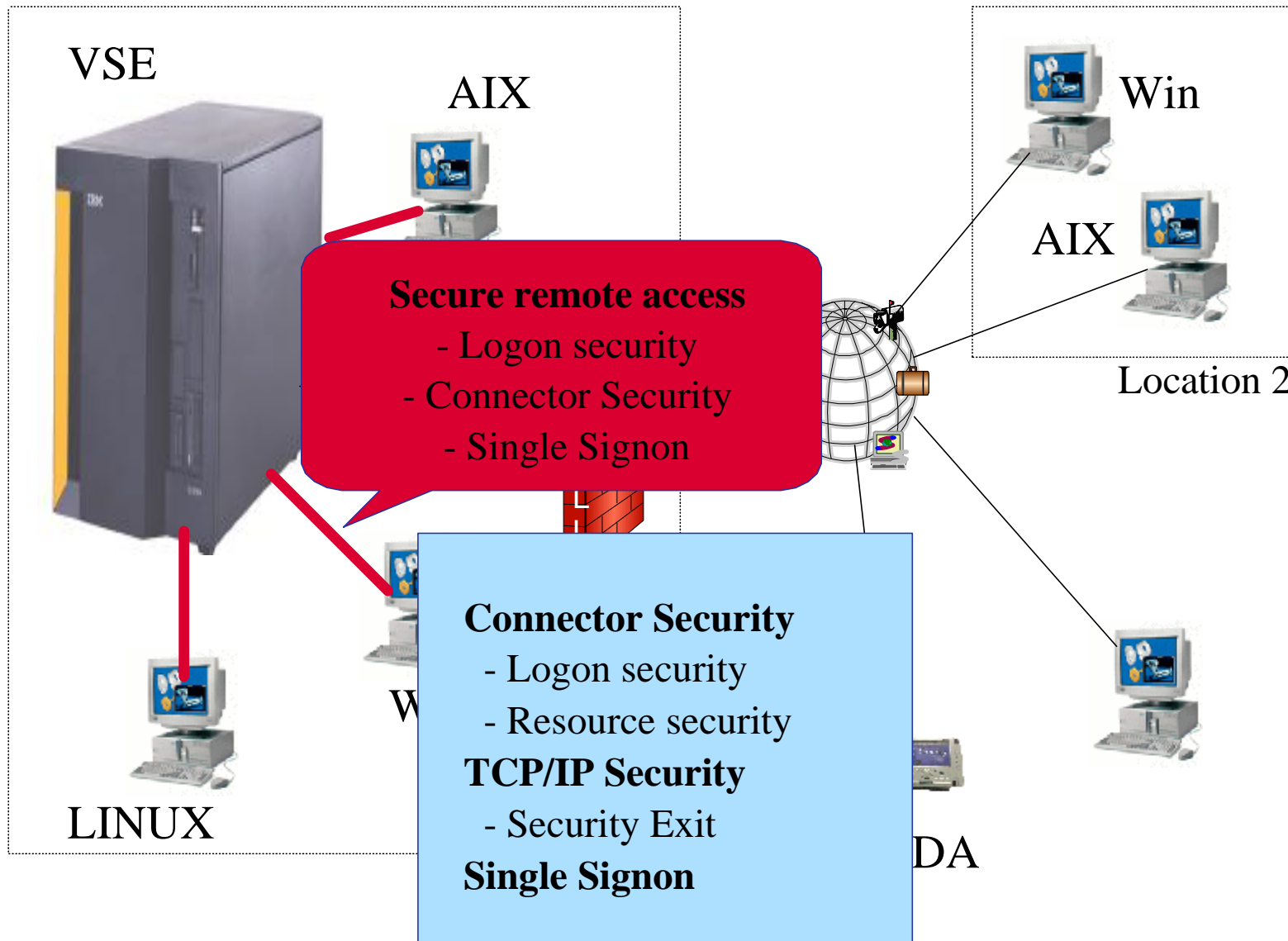
§ **CICS SIT SEC=YES (!)**

- If NO, all users can logon without a password

§ **Change passwords for predefined users**

- POST, PROG, OPER, SYSA, ...

Security in a heterogeneous environment





Why secure remote access ?

§ Today most computers are part of a network

- Can connect to your VSE system

§ Prevent unauthorized access to VSE and data

- Requires to authenticate the user (logon)

§ FTP allows to access production data

- VSAM
- POWER entries (listings)





Connector Security

§ VSE Connector Server acts as a Resource Manager

- Issues RACROUTE calls for
 - User id and password verification
 - Resource security

§ Connector user ids are the same as for CICS TS and Batch

§ No additional user profile setup required

§ But:

- Additional access restriction by user id and/or IP address possible





Connector Security - Logon

- § **VSE Connector Server requires a client to logon with valid user id and password**
- § **User id 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 (VCSRVR)

- 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

- User id
- 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

```

Deactivation of Connector Security

§ Since PTF UQ66736 (VSE/ESA 2.6), UQ66733 (VSE/ESA 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



CP/IP Security

§ In general TCP/IP uses its own user id definitions

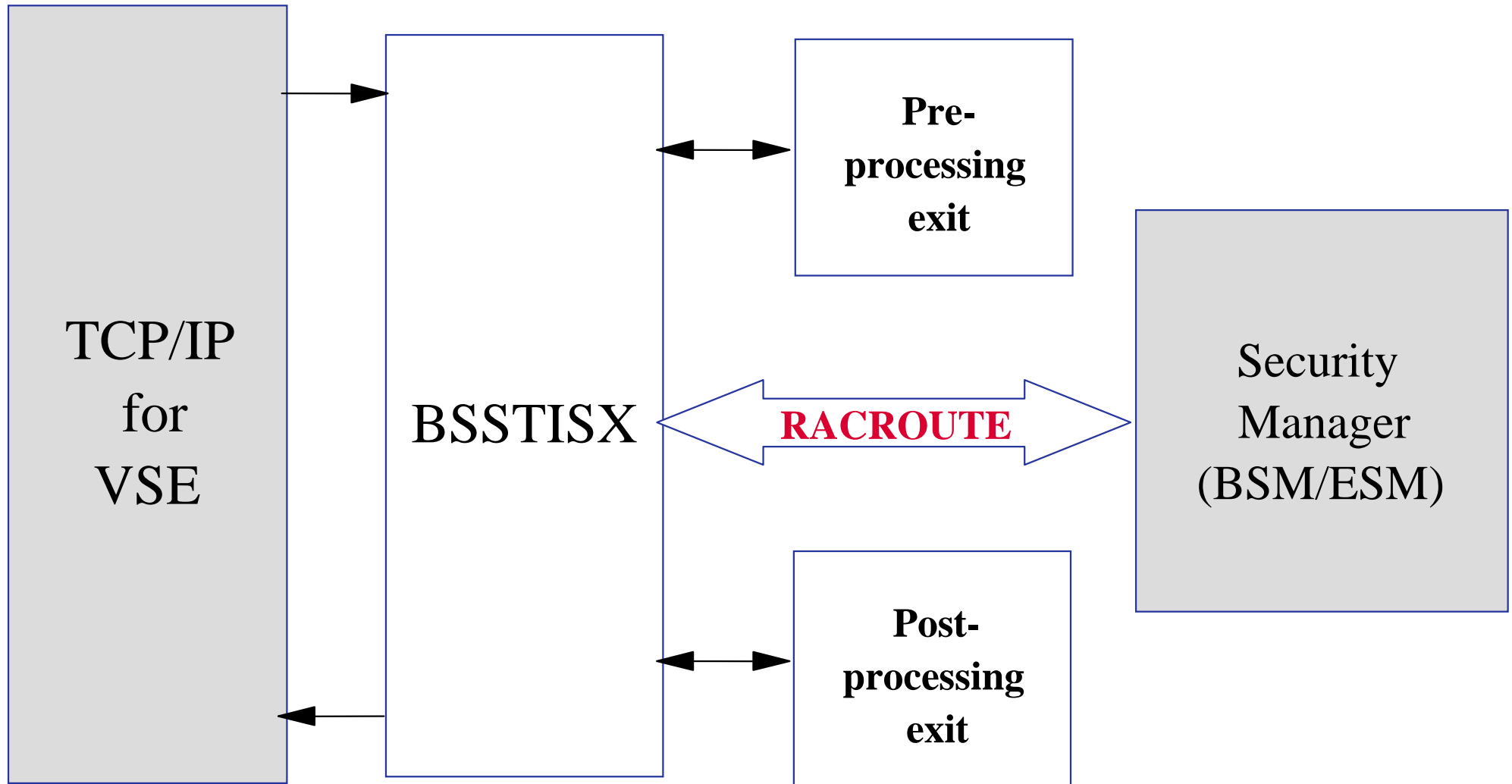
- DEFINE USER, ID=user, PASSWORD=pwd
- Readable in initialization member (IPINITxx.L)
- Duplicate user definitions
- Used for
 - FTP

§ Security Exit available from IBM to check the user ids and resource access via Security Manager

- see next foil



CP/IP Security Exit





CP/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)

CP/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 Sign on Solutions

§ Every server/application requires you to logon

- Different user ids and passwords for each server

§ A single sign on solution

- Requires a user to sign on only once
 - one user id, one password
- Stores sign on information for several servers or applications
- Automatically performs a sign on on each server or application
- Using the stored sign on information

§ Example: LDAP





Security Checklist for TCP/IP

§ Connector Security

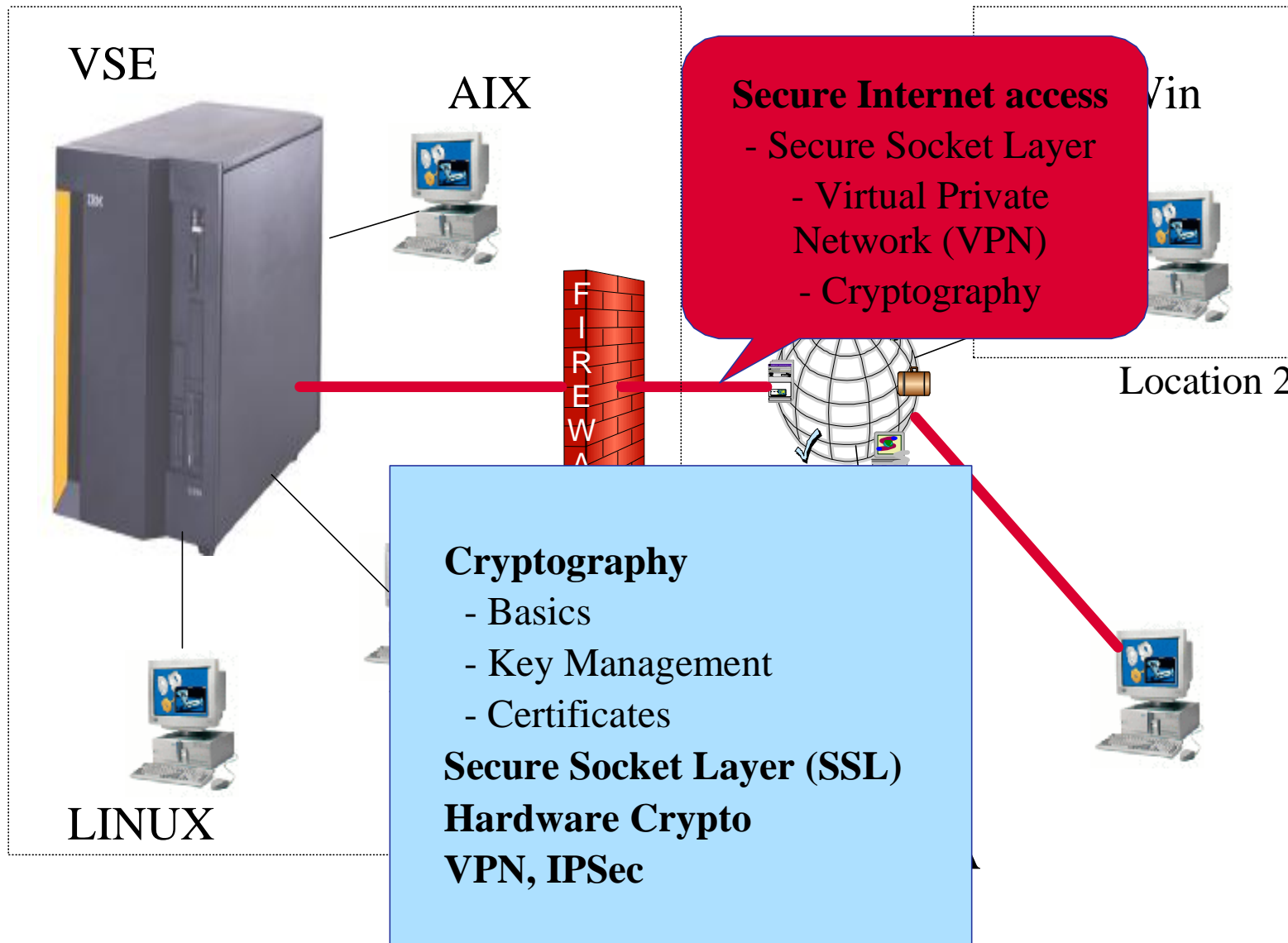
- Set SECURITY=FULL (SKVCSCFG)
- Define resource access rights (BSM/ESM)
- Restrict remote access to specific users and IPs (SKVCSUSR)

§ TCP/IP Security

- SET SECURITY=ON in IPINIT member
- Use Security Exit
- Do not define users in IPINIT member



Security in a heterogeneous environment





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?

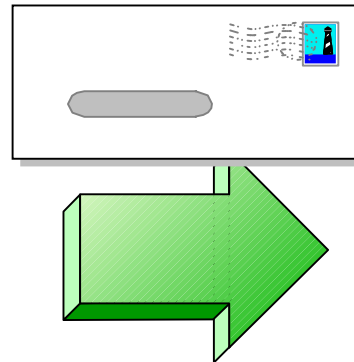
Keeping Secrets



Charly



Alice



Bob

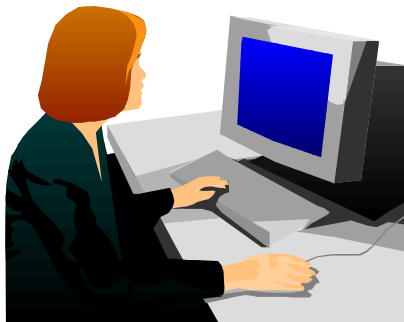
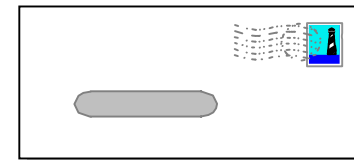


Alice encrypts the message with a secret code that only she and Bob knows

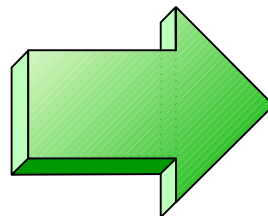
oving Identity



Charly



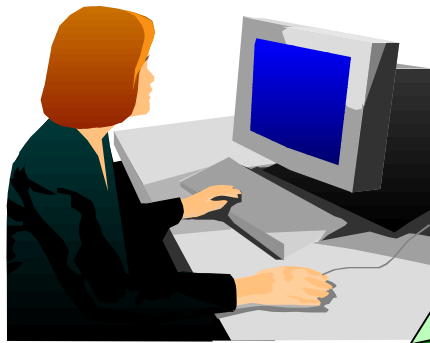
Alice



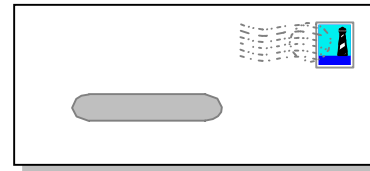
Bob

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

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.

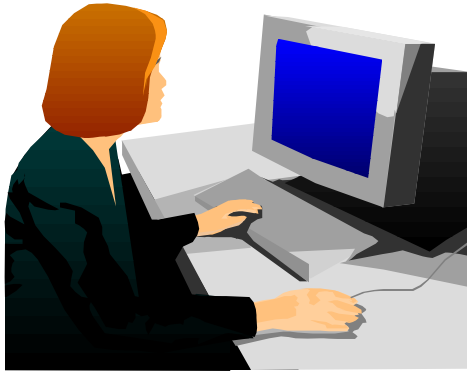


Secret Key Cryptography (symmetric)

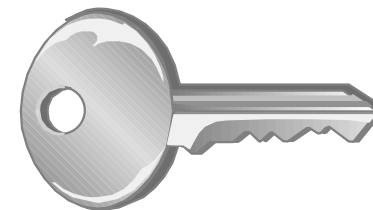
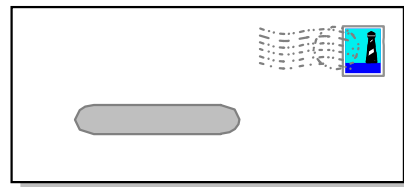
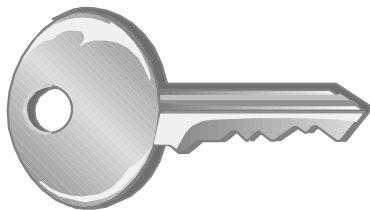
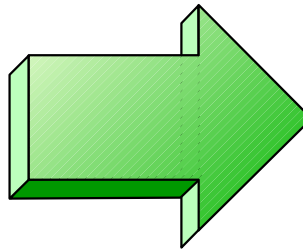
- § **Both parties know the same secret code (key)**
- § **The key must be kept secret**
- § **Encryption algorithm = mathematical transformation of the data with the key**
 - DES Data Encryption standard
 - 3DES Triple strength DES
 - RC2 Rivest Cipher 2
 - RC4 Rivest Cipher 4
- § **Typical key length: 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.



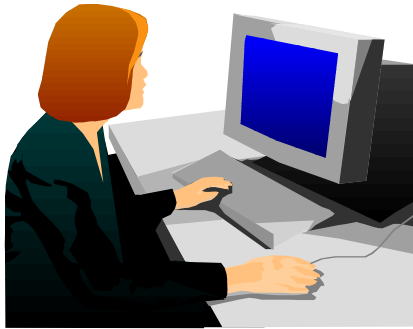
Public Key Cryptography (asymmetric)

- § **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**

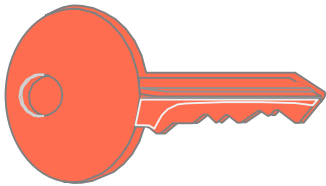
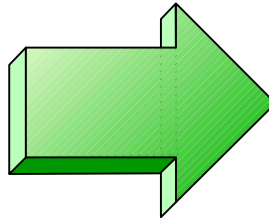


Public Key Cryptography - Encrypting

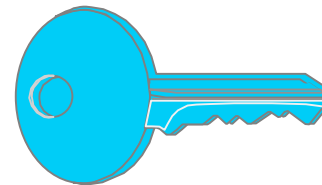
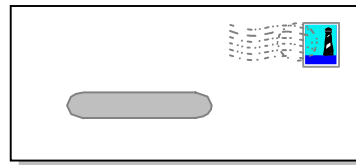
Alice



Bob



Bob's public key

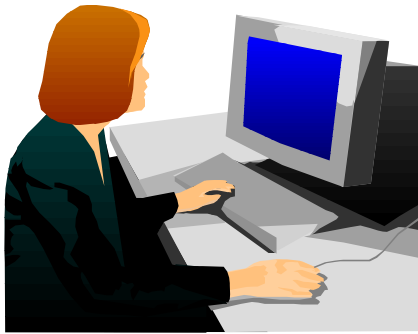


Bob's private key

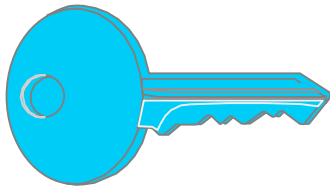
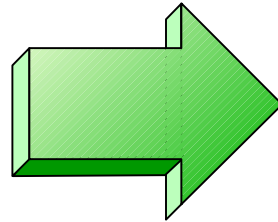
Alice encrypts the message using Bob's public key and sends it to Bob. Bob decrypts it using his private key. Since only Bob knows his private key, only he can read the message.

Public Key Cryptography - Signing

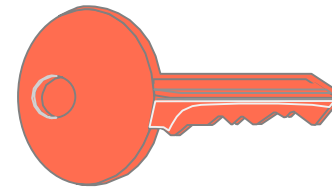
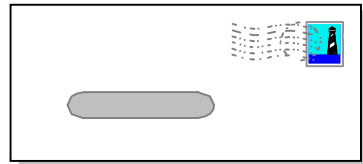
Alice



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

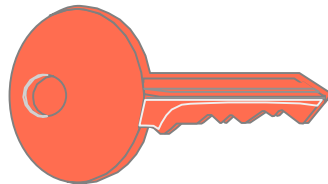
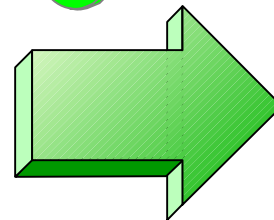
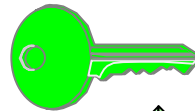
Alice



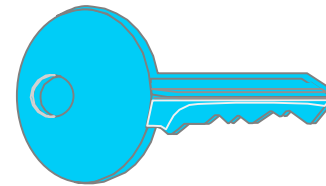
Bob



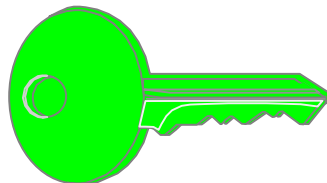
secret key



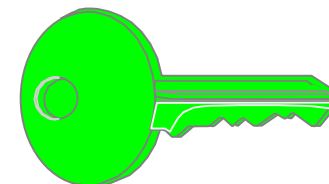
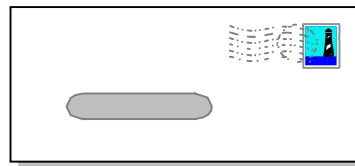
Bob's public key



Bob's private key



secret key

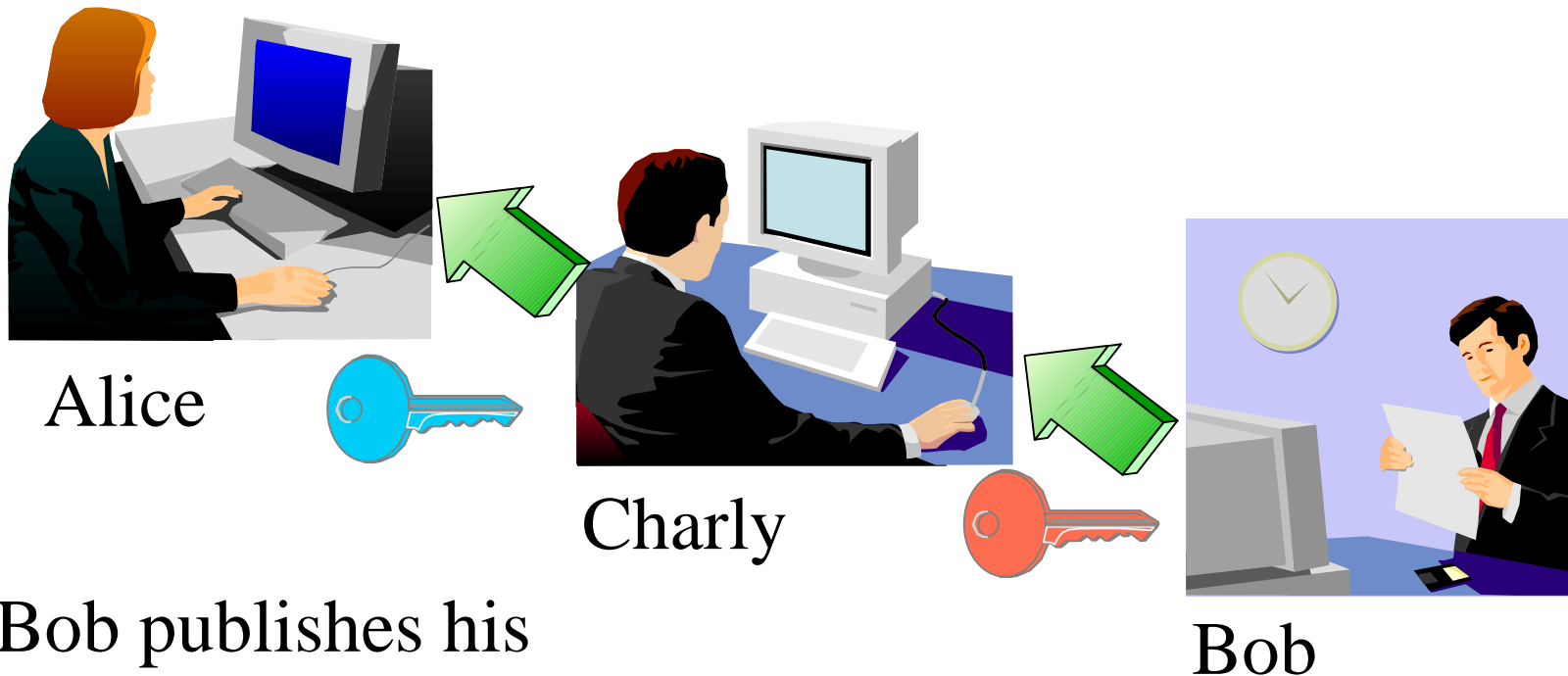


secret key



Key Management

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



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



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



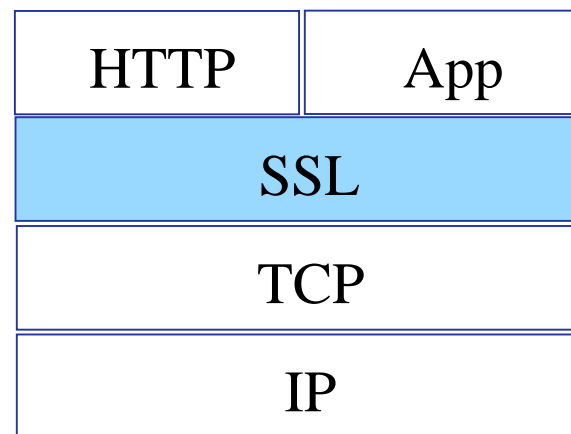
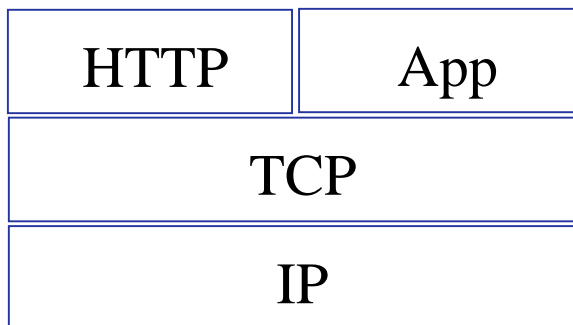
Certificate Authorities

- § **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**



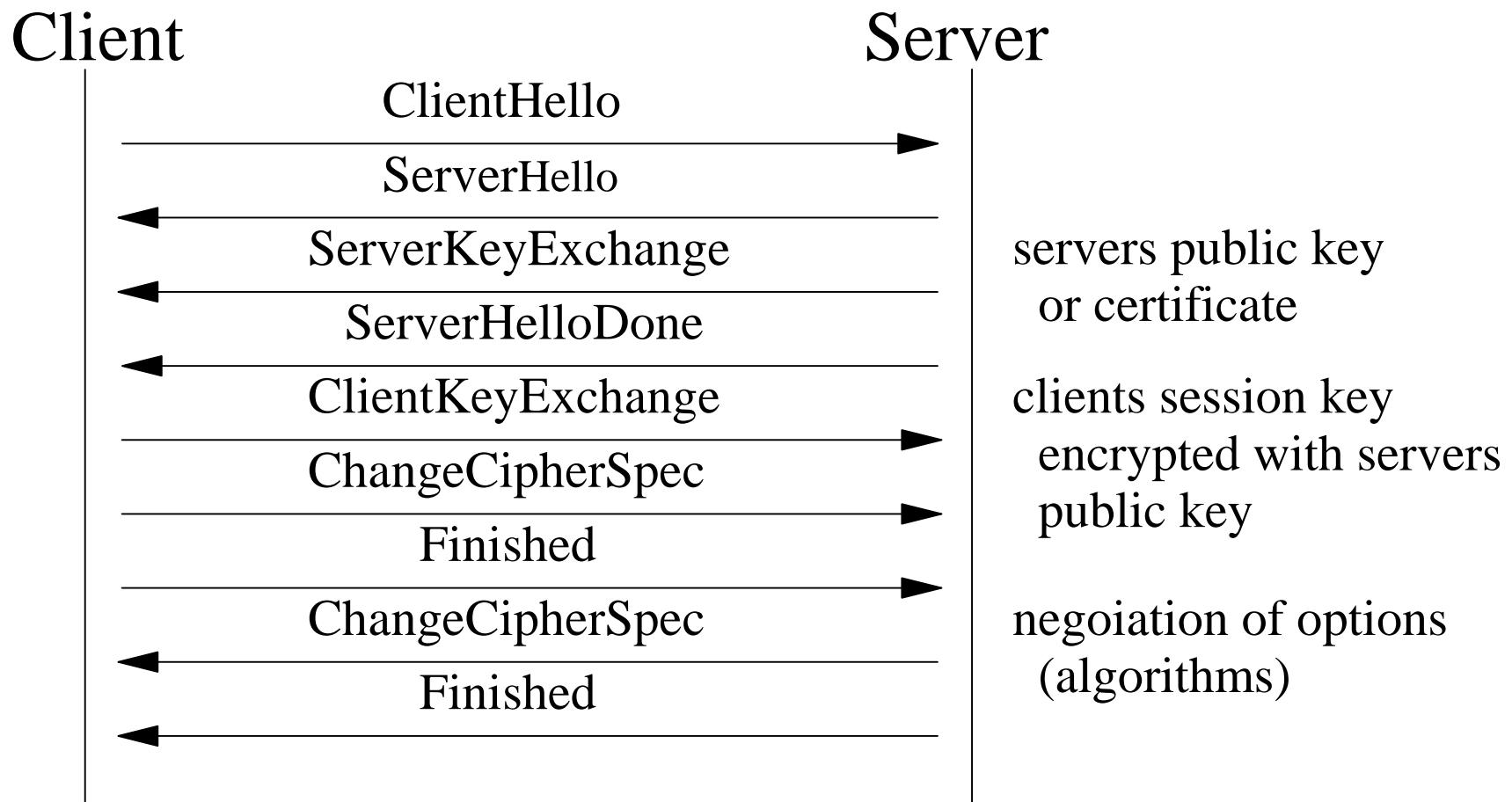
SL (Secure Socket Layer)

- § 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**



SSL Protocol

§ The SSL protocol defines a set of messages





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
- Negotiated algorithms

§ Establishing a SSL Session is a complex and time consuming mechanism

§ Session caching allows to reuse previously negotiated SSL parameters

§ No need of repeating the negotiations 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



SSL for VSE

- § **SSL for VSE is part of the TCP/IP for VSE base**
- § **Enabled with the Application Pak**
- § **Integrated into TCP/IP for VSE**
- § **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**





SL 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,  sub library name
        CERTMEM=MYKEY,    member name
        TYPE=1,           server application
        MINVERS=0300,     SSL 3.0
        DRIVER=SSLD       Driver phase name
```





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





Tools for setting up SSL

§ Keyman/VSE

- Creation of RSA keys and digital certificates
- Upload of keys and certificates to VSE
- Creation of PKCS#12 keyring files (use with Java-based connector or import into a Web browser)

§ Keyman Professional

- Provided by Zurich Research Lab
- No VSE specific functions

→ **Downloadable from VSE Homepage**

<http://www.ibm.com/servers/eserver/zseries/zvse/downloads/#vkeyman>





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





Hardware Crypto Overview

§ Pluggable crypto cards

- PCICA (VSE/ESA 2.7 and higher)
- Crypto Express2 (z/VSE 3.1 with APAR **DY46230**, PTF **UD52721**), a PTF for VSE/ESA 2.7 will follow ...
- Supports RSA operations, i.e. SSL handshaking

§ CPACF (CPU Assist for Cryptographic Function)

- Available on z890 and z990 on board
- z/VSE 3.1 GA level, PTF for 2.7 with DY46230
- Supports symmetric algorithms (DES, SHA-1), i.e. data encryption/decryption

requires TCP/IP 1.5 E



Support on zSeries processors

	z800	z900	z890	z990
PCICA	-	yes	yes	yes
CEX2C	-	-	yes	yes
CPACF	-	-	yes	yes
CEX2A	-	-	-	-

CEX2C = Crypto Express2 in coprocessor mode

CEX2A = Crypto Express2 in accelerator mode



SE Hardware Configuration

§ VSE hardware configuration not necessary for crypto hardware

- No IOCDs definition in VSE
- No device type
- No ADD statement

§ Use of crypto hardware is transparent to end users and even TCP/IP applications

- But use of crypto hardware can be disabled via TCP/IP SOCKOPT phase





W-Crypto related console messages

§ System with crypto hardware

```
FB 0095 1J023I FOUND A CRYPTO EXPRESS2 CARD AT DEVICE INDEX 0
FB 0095 1J023I FOUND A CRYPTO EXPRESS2 CARD AT DEVICE INDEX 1
FB 0095 1J014I FOUND A PCICA CARD AT DEVICE INDEX 6
FB 0095 1J014I FOUND A PCICA CARD AT DEVICE INDEX 7
FB 0095 1J005I HARDWARE CRYPTO ENVIRONMENT INITIALIZED SUCCESSFULLY.
FB 0095 1J006I USING CRYPTO DOMAIN 0
FB 0095 1J022I CPU CRYPTOGRAPHIC ASSIST FEATURE AVAILABLE.
```

§ System without crypto hardware

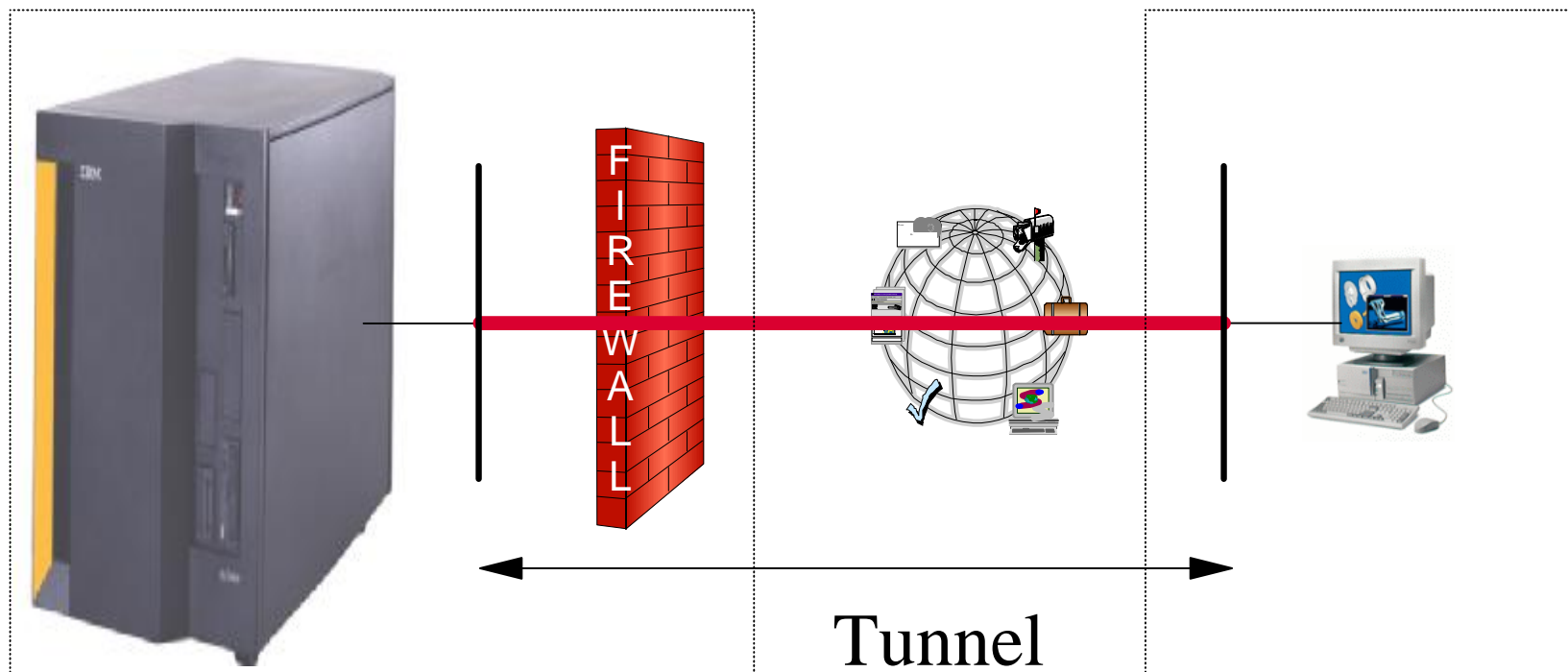
```
FB 0093 1J020W THERE WAS NO PCICA OR CRYPTO EXPRESS2 CARD
FB 0093          FOUND. HARDWARE CRYPTO NOT AVAILABLE.
```



Virtual Private Network, IPSec

§ Can be used to define a secure tunnel between two locations

- Makes use of cryptographic functions





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





Related Documentation

§ Security Homepage

– <http://www.ibm.com/servers/eserver/zseries/zvse/documentation/security.html>

§ z/VSE Planning

§ z/VSE Administration

§ VSE/ESA Software Newsletter No. 17, 18 and 20

§ OS/390 Security Server External Security Interface (RACROUTE) Macro Reference (GC28-1922)

§ OS/390 Security Server (RACF) Data Areas (SY27-2640)

§ z/VSE V3R1.0 e-business Connectors, User's Guide, SC33-8231

– <http://www-1.ibm.com/servers/eserver/zseries/zvse/documentation/>

§ CICS Enhancements Guide, GC34-5763

– <http://www-1.ibm.com/servers/eserver/zseries/zvse/documentation/>

§ VSE/ESA 2.7.3 Release Guide, Chapter 1, section “Hardware Crypto Support”

– <ftp://ftp.software.ibm.com/eserver/zseries/zos/vse/pdf3/vse27/iesrie24.pdf>

§ IBM PCI Cryptographic Accelerator

– <http://www.ibm.com/security/cryptocards/pcica.shtml>

§ zSeries cryptography for highly secure transactions

– <http://www.ibm.com/servers/eserver/zseries/security/cryptography.html>

§ Q & A for Cryptographic Support for z990 servers

– <http://www.ibm.com/security/cryptocards/zservcryptofaq.shtml>





Questions ?

