z/OS V1R10 Communications Server: TCP/IP Cryptography Demystified

Chris Meyer
meyerchr@us.ibm.com

Trademarks and notices

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM (logo)®
- IBM®
- IMS
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x
- System z
- System z9
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Refer to www.ibm.com/legal/us for further legal information.

Why this presentation?

To answer the question…

"What hardware crypto facilities get used when,

(and who gets charged for the cycles)?"

Agenda

- **Review of basic cryptographic operations**
  - Symmetric cryptography
  - Asymmetric cryptography
  - Message digests and Message Authentication Codes
  - Digital certificates
- **Relevant System z and z/OS cryptographic componentry**
  - Hardware components
  - Software components
- **z/OS TCP/IP cryptography and how it uses z components**
  - SSL/TLS
  - AT-TLS
  - IPSec and IKE
- **Conclusion**

Agenda

- **Review of basic cryptographic operations**
  - Symmetric cryptography
  - Asymmetric cryptography
  - Message digests and Message Authentication Codes
  - Digital certificates
- **Relevant System z and z/OS cryptographic componentry**
  - Hardware components
  - Software components
- **z/OS TCP/IP cryptography and how it uses z components**
  - SSL/TLS
  - AT-TLS
  - IPSec and IKE
- **Conclusion**

Cryptographic Basics

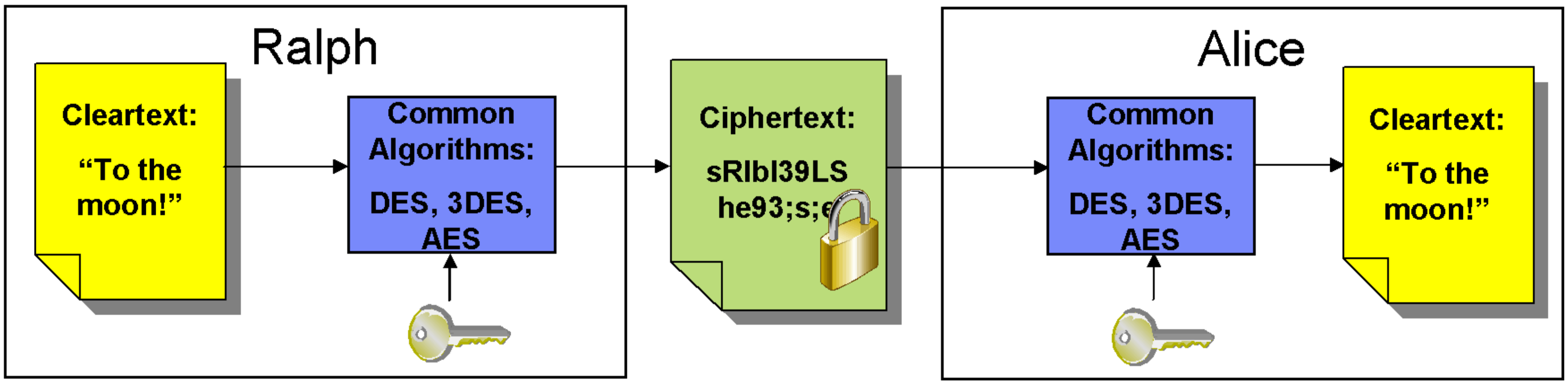
- **Cryptography is the use of mathematical algorithms to transform data for the purposes of ensuring:**
  - Data privacy – hiding the data (encryption/decryption)
  - Data integrity – proving the data hasn't been modified since it was sent (message digests)
  - Data authentication – proving the data's origin (Message Authentication Codes (MACs) and digital signatures)
- **Cryptographic operations are compute intensive, hence the need for hardware assist technologies**
- **General rule: *For a given algorithm*,**
  longer keys == stronger security == more intensive computation
  - For example, AES-128 vs. AES-256
  - Increases the amount of work an attacker needs to do to crack the code

Glossary

- AES – Advanced Encryption Standard (symmetric encryption, 128/192/256/512 bit keys)
- AH – Authentication Header (IPsec data authentication protocol)
- DES – Digital Encryption Standards (symmetric encryption, 56 bit keys)
- 3DES – Triple-DES (symmetric encryption, 168 bit keys)
- Diffie-Hellman – secure key exchange algorithm
- DSA – Digital Signature Algorithm (asymmetric encryption, 512/1024 bits*)
- ESP – Encapsulating Security Payload (IPsec data privacy and authentication protocol)
- IKE – Internet Key Exchange (protocol used for setting up dynamic IPsec tunnels)
- IPsec – IP security (secure networking protocol, consists of AH and ESP)
- MD5 – Message Digest 5 (message digest, 128 bits)
- SHA-1 – Secure Hash Algorithm-1 (message digest, 160 bits)
- SHA-2 – Secure Hash Algorithm-2 (message digest, 224/256/384/512 bits)
- SSL – Secure Sockets Layer (secure networking protocol for authentication and privacy)
- RSA – Rivest, Shamir, Adleman (asymmetric encryption, 1024/2048/4096 bit keys*)
- TLS – Transport Layer Security (IETF-adopted form of SSL)

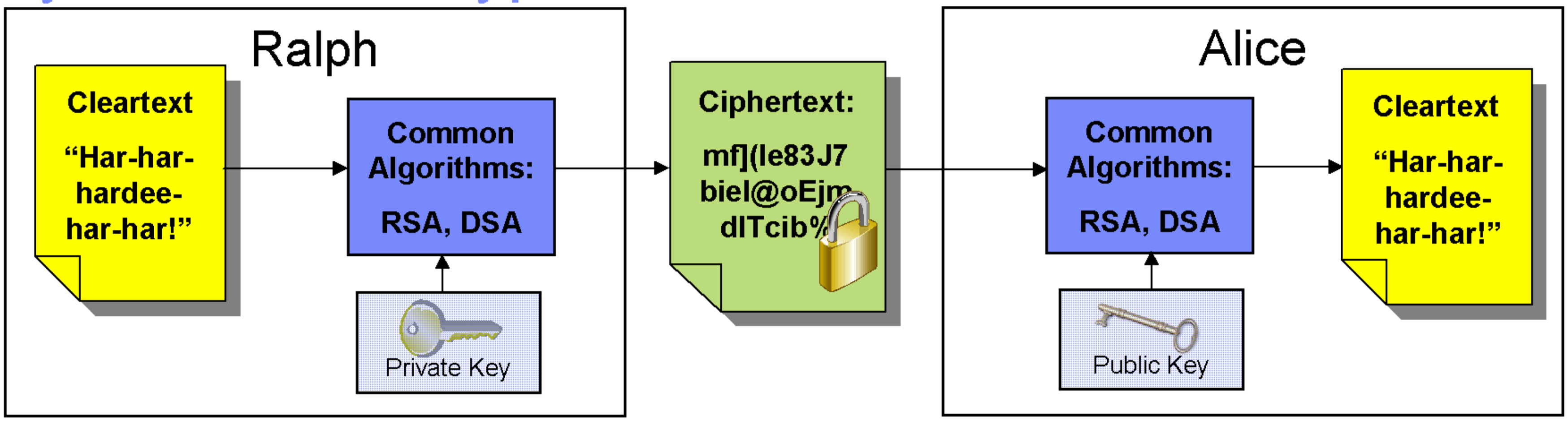
* other sizes allowed in between

Symmetric encryption



- **Only one key value - "shared secret" between both parties**
  - Used for both encryption and decryption
  - Hence, the symmetry – each side has the same key
- **Much faster than asymmetric crypto – great for bulk encryption**
- **Securely sharing the key between both parties is a major issue**
- **Also known as…**
  - "secret key encryption"
  - "private key encryption" (easily confused with asymmetric)

Asymmetric encryption

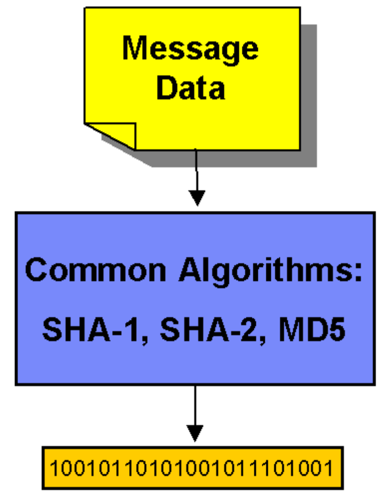


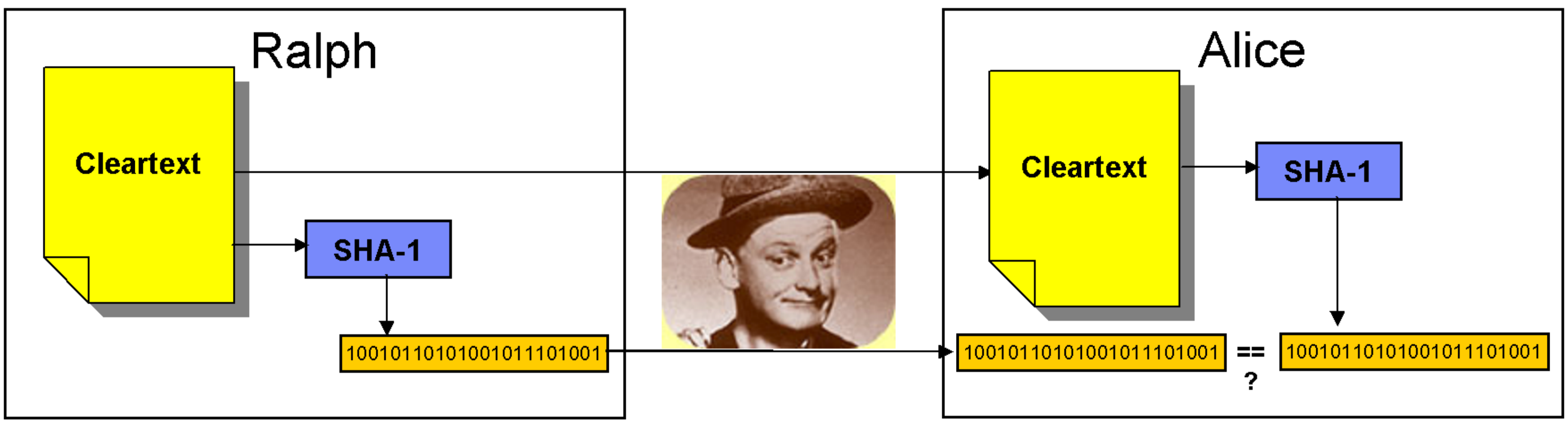
- **Two different key values – no shared secrets!**
  - Private key is known only to owner
  - Public key is freely distributed to others
  - Data encrypted with private key can only be decrypted with public key and vice versa
  - No way to derive one key value from the other
- **Great for authentication and non-repudiation**
  - "digital signatures" - signing with private key
- **Very expensive computationally**
  - Not so great for bulk encryption
  - Usually used to encrypt small data objects like message digests or symmetric keys
- **Also known as "public key cryptography"**

Message digests

A message digest is…

- not based on a secret key
- a fixed-length value generated from variable-length data
- unique:
  - the same input data always generates the same digest value
  - small change in data generates a very different hash value
  - extremely difficult (and time consuming) to find two different data values that result in the same hash value
- one-way: can't reverse a digest value back to the original data
- also known as a "one-way hash"
- very good at proving data integrity (but not origin)

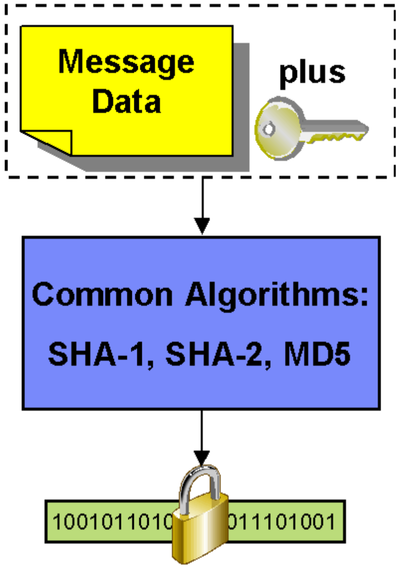


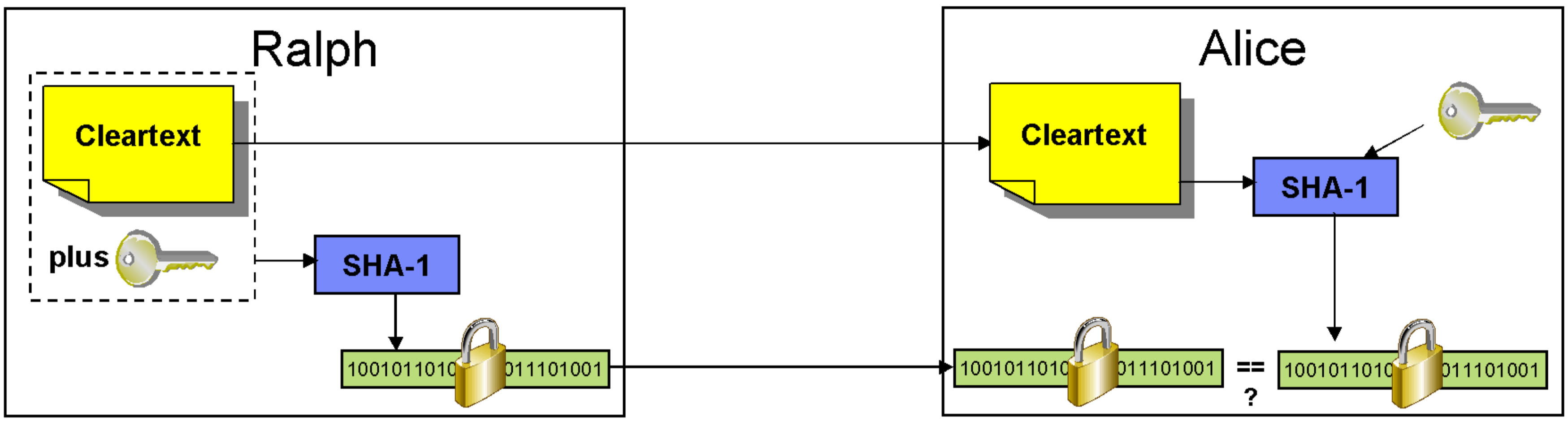


Message Authentication Codes (MACs)

A Hashed Message Authentication Code (HMAC) is…

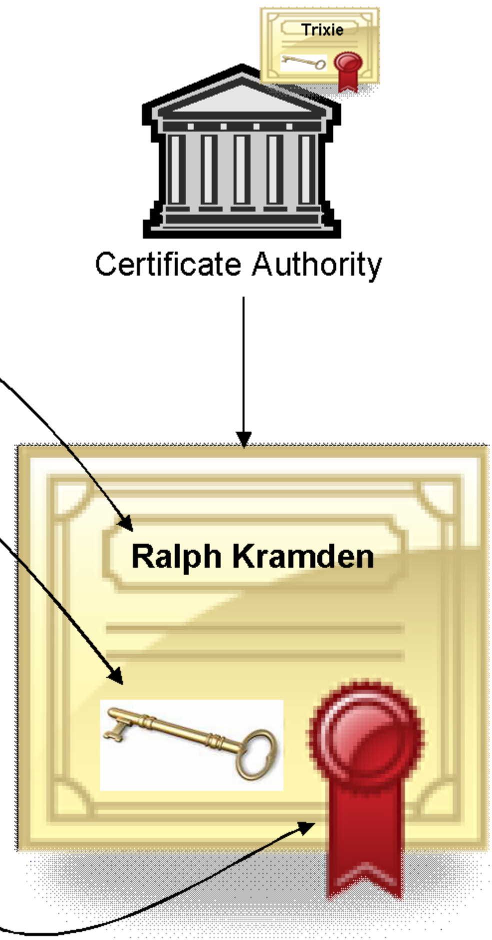
- **a message digest**
- **generated on a concatentation of**
  - **variable-length data**
  - **a secret (symmetric) key**
- **very good for proving data integrity as well as authenticating data origin**





Digital certificates

- **A digital document that…**
  - Is issued by a trusted third party called a Certificate Authority (CA)
  - Identifies a subject
  - Contains:
    - cleartext (issuer, serial number, etc.)
    - the subject's public key
    - a signed hash of the cleartext (signed by the issuer) – proves certificate's validity
- **…and is used to…**
  - Distribute a public key
  - Prove the subject's identity
- **Part of a trust hierarchy**
  - CA's (issuer) have their own certificates. (well-known CA certs are distributed with web browsers).
  - Subjects usually only need to store their own and the CA's cert in their keyring (exception is self-signed certs)
  - Part of a Public Key Infrastructure (PKI)



Agenda

- Review of basic cryptographic operations
  - Symmetric cryptography
  - Asymmetric cryptography
  - Message digests and Message Authentication Codes
  - Digital certificates
- **Relevant System z10/z9 and z/OS cryptographic componentry**
  - Hardware components
  - Software components
- z/OS TCP/IP cryptography and how it uses z components
  - SSL/TLS
  - AT-TLS
  - IPSec and IKE
- Conclusion

z10, z9 Hardware Cryptographic components*

- **CP Assist for Cryptographic Function (CPACF)**
  - **Hardware assist for specific System z cryptographic instructions (DES, 3DES, AES encrypt/decrypt and SHA-1, SHA-2 hashing)**
  - **Available on general processors as well as zIIPs**
  - **Accessed directly through z series instruction set or through ICSF**
  - **Clear keys only (unencrypted key is kept in storage)**
  - **Available on z10, z9 and z890/z990**
- **Cryptographic adapters (e.g., CryptoExpress2)**
  - **Accelerators (CEX2A, for example)**
    - Performs RSA encrypt/decrypt and RSA signature operations
    - Accessed through ICSF
    - Clear keys only

  - **Coprocessors (CEX2C, for example)**
    - Focus on secure keys (no unencrypted keys in storage) and tamper detection / countermeasures
    - Provides RSA acceleration as well (slower than accelerators, though)
    - Accessed through ICSF

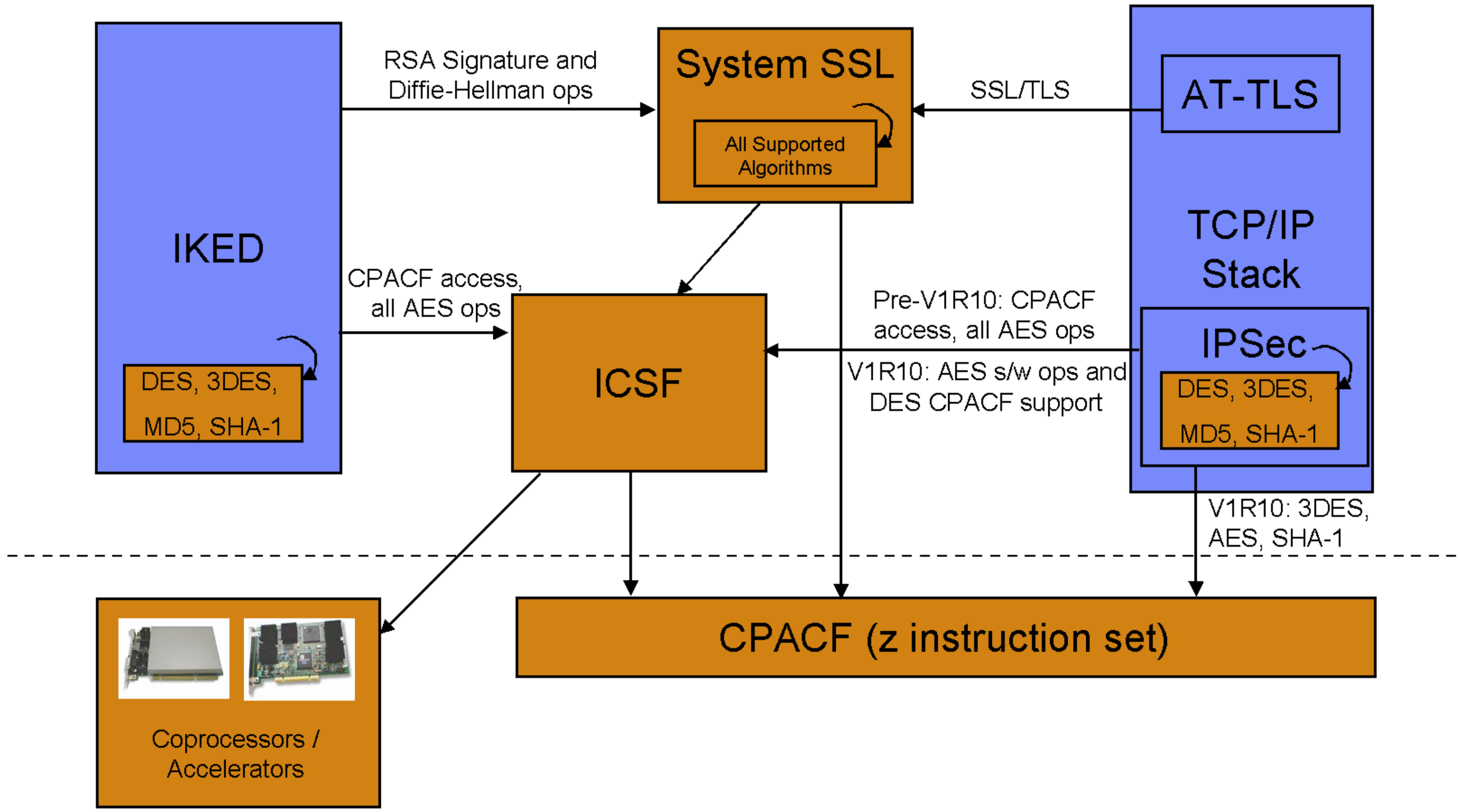
- **z9 or z10 Integrated Information Processor (zIIP)**
  - **Can be tasked to perform some crypto-intensive portions of IPsec processing**

* - capabilities are described relative to their usage by z/OS Communications Server and by System SSL only

z/OS Software Cryptographic components

- **z/OS Cryptographic Services**
  - Integrated Cryptographic Service Facility (ICSF)
    - z/OS component that provides secure, high-speed crypto services
    - A variety of cryptographic primitives
    - Application access to z/OS hardware crypto features
  - System SSL
    - z/OS component that provides SSL, TLS implementations
    - Also provides certificate-related APIs, including RSA signature generation and validation
    - Contains own software implementations of all crypto algorithms
    - Makes use of hardware crypto facilities to varying degrees
- **z/OS Communications Server**
  - TCP/IP stack implements
    - Application-Transparent TLS
    - IPsec
  - Internet Key Exchange daemon (IKED)
  - Both contain software implementations of most algorithms (not AES)
  - Both use hardware crypto facilities to varying degrees

z/OS TCP/IP Cryptographic Landscape



Agenda

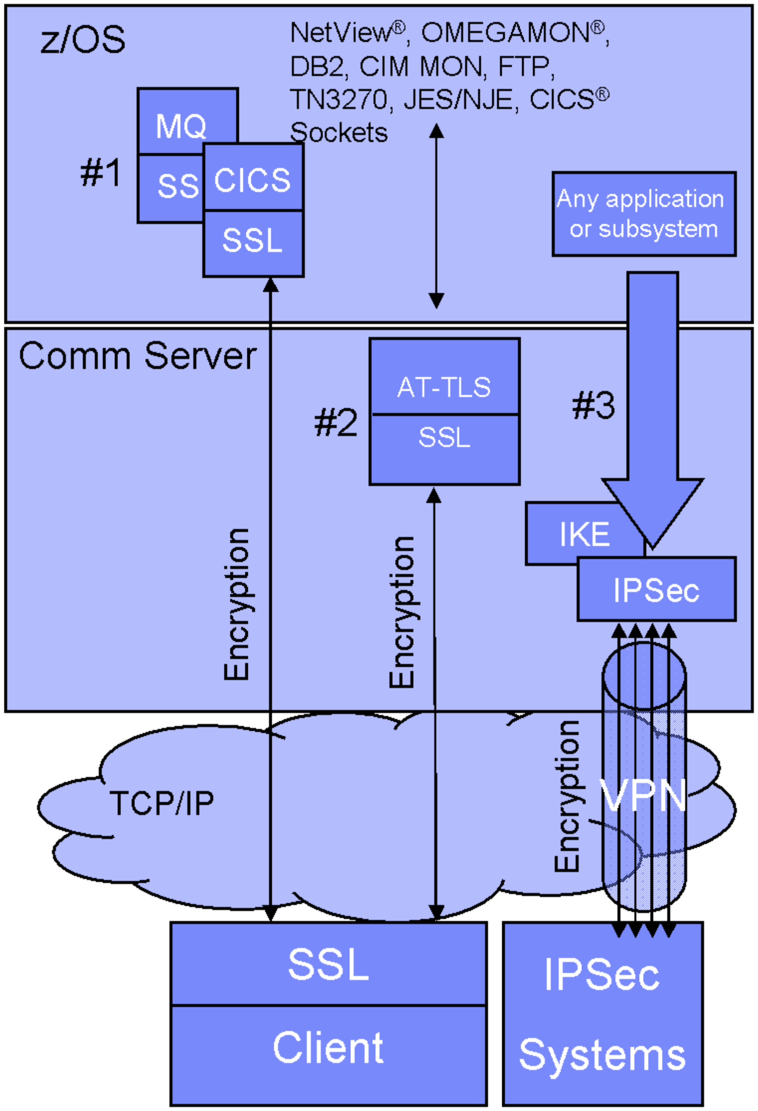
- **Review of basic cryptographic operations**
  - Symmetric cryptography
  - Asymmetric cryptography
  - Message digests and Message Authentication Codes
  - Digital certificates
- **Relevant System z and z/OS cryptographic componentry**
  - Hardware components
  - Software components
- **z/OS TCP/IP cryptography and how it uses z components**
  - SSL/TLS
  - AT-TLS
  - IPSec and IKE
- **Conclusion**

z/OS TCP/IP secure networking protocols

- **z/OS TCP/IP cryptographically protects network data in 3 ways:**
  #1 Secure Sockets Layer (SSL) and Transport Layer Security (TLS) through System SSL
  - **Application is explicitly coded to use these**
  - **Per-session protection**
  - **TCP only**

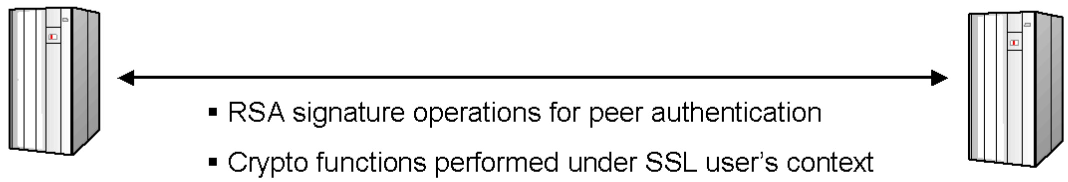
  #2 Application Transparent TLS (AT-TLS)
  - **TLS applied in transport layer (TCP) as defined by policy**
  - **Typically applied transparently to application**
  - **TCP/IP stack is user of System SSL services**

  #3 Virtual Private Networks using IP Security (IPSec) and Internet Key Exchange (IKE)
  - **"Platform to platform" encryption**
  - **IPSec implemented at the IP layer as defined by policy**
  - **Wide variety (any to all) of traffic is protected**
  - **Completely transparent to application**
  - **IKE allows IPSec tunnels to be established dynamically**
- **When do you use one form versus another?**
  - **Depends on client, application, topology, performance requirements, etc.**
  - **Beyond scope of this presentation**

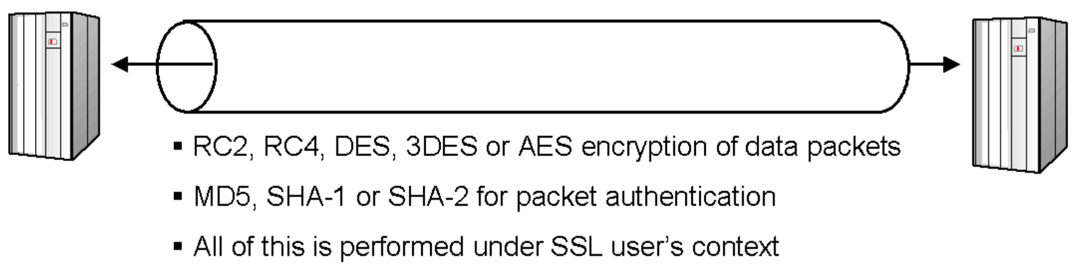


Establishing SSL/TLS sessions

1 SSL handshake identifies and authenticates SSL client and server and negotiates cipher suite to be used for data protection

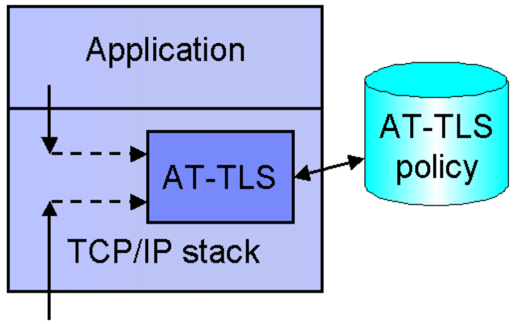


2 Data flows through protected session using symmetric encryption and message authentication negotiated during handshake

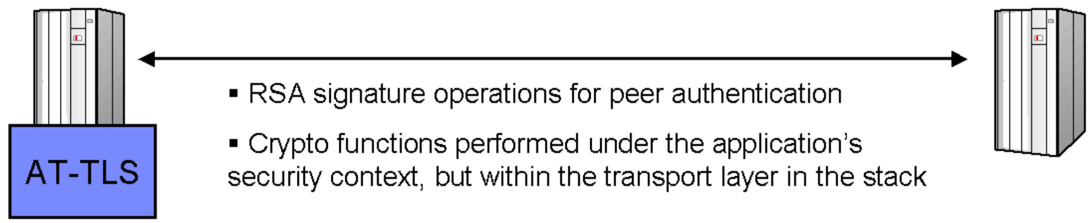


AT-TLS sessions

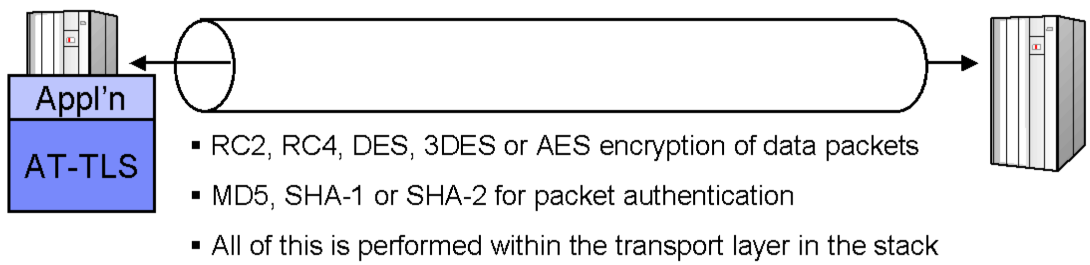
1 A z/OS application issues a connect() or accept() on a socket to establish a new outbound or inbound connection, respectively. Within the transport layer of the stack, AT-TLS policy is consulted to decide if TLS protection is configured for this traffic. If so, the stack's AT-TLS support establishes the TLS connection…



2 AT-TLS directs the SSL handshake. All identities, cipher suites, etc. are defined in AT-TLS policy. Note that sessions established by AT-TLS on z/OS interoperate seamlessly with "regular" TLS applications on remote nodes.



3 AT-TLS takes outbound cleartext and sends it over the TLS-protected session. Likewise, it receives encrypted data off the session and presents it to the application as cleartext.. Many applications never know the TLS session exists, although some may want/need to (AT-TLS aware, AT-TLS controlling)

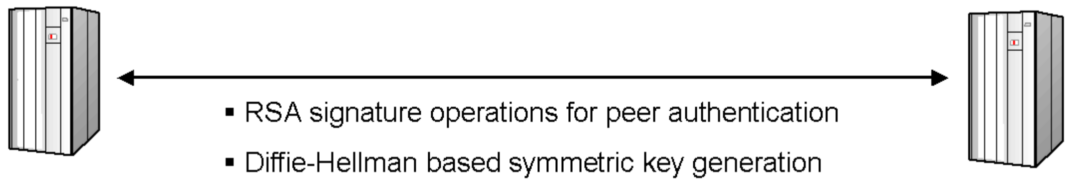


SSL/TLS (and AT-TLS) hardware crypto usage

Algorithm	CPACF only	CPACF + Coprocessor/Accelerator
RSA signature generation	In software	In coprocessor mode only. Otherwise in software (accelerator does not support this operation).
RSA signature verification	In software	In coprocessor/accelerator.
PKA encrypt/decrypt for handshake	In software	In coprocessor/accelerator
SHA-1 digest generation	CPACF	
SHA-224 digest generation	CPACF	
SHA-256 digest generation	CPACF	
SHA-384 digest generation	In software on z9, CPACF in z10 EC	
SHA-512 digest generation	In software on z9, CPACF in z10 EC	
DES encrypt/decrypt	CPACF	
3DES encrypt/decrypt	CPACF	
AES-128 encrypt/decrypt	CPACF	
AES-256 encrypt/decrypt	In software on z9, CPACF in z10 EC	

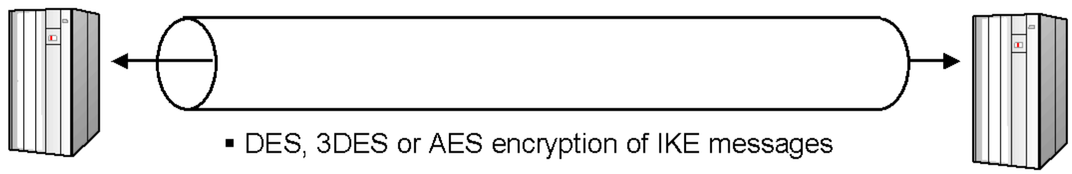
Creating IPSec Security Associations (SAs)

1 IKE peers negotiate an IKE (“phase 1”) tunnel (one bidirectional SA) over an unprotected UDP socket.



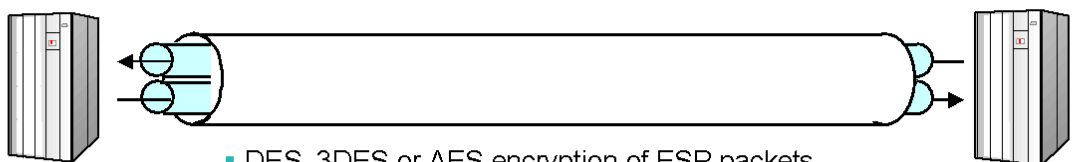
- RSA signature operations for peer authentication
- Diffie-Hellman based symmetric key generation
- IKE daemon invokes crypto operations

2 IKE peers negotiate IPSec (“phase 2”) tunnel (two unidirectional SAs) under protection of the IKE tunnel



- DES, 3DES or AES encryption of IKE messages
- MD5 or SHA1 hashing for IKE message authentication
- IKE daemon invokes crypto operations

3 Data flows through IPSec tunnel using Authentication Header (AH) and/or Encapsulating Security Payload (ESP) protocol



- DES, 3DES or AES encryption of ESP packets
- MD5 or SHA1 hashing for AH or ESP packets
- TCP/IP stack invokes crypto operations

IKE hardware crypto usage (IKED)

- **Diffie-Hellman based symmetric key generation**
  - Generated keys are used to encrypt traffic that flows over Phase 1 SA
- **RSA signature generate, signature verify for peer authentication**
  - Due to z/OS IKED single-threaded design, multiple Coprocessors or Accelerators will not provide any significant advantage for IKE operations
- **DES, 3DES, AES encryption of IKE payloads**
  - AES requires ICSF (unsupported if ICSF is not available)
- **SHA-1, MD5 HMACs for message authentication**

Algorithm	CPACF only	CPACF + Coprocessor/Accelerator
Diffie-Hellman operations	In software via System SSL	In software via System SSL
RSA signature generation (clear key only)	In software via System SSL	In Coprocessor (not accelerator) if available, otherwise in software
RSA signature verification	In software via System SSL	In Coprocessor/Accelerator
DES	In software	
3DES	In software	
AES-128	In CPACF via ICSF, otherwise not supported	
SHA-1	In software	
MD5	In software	

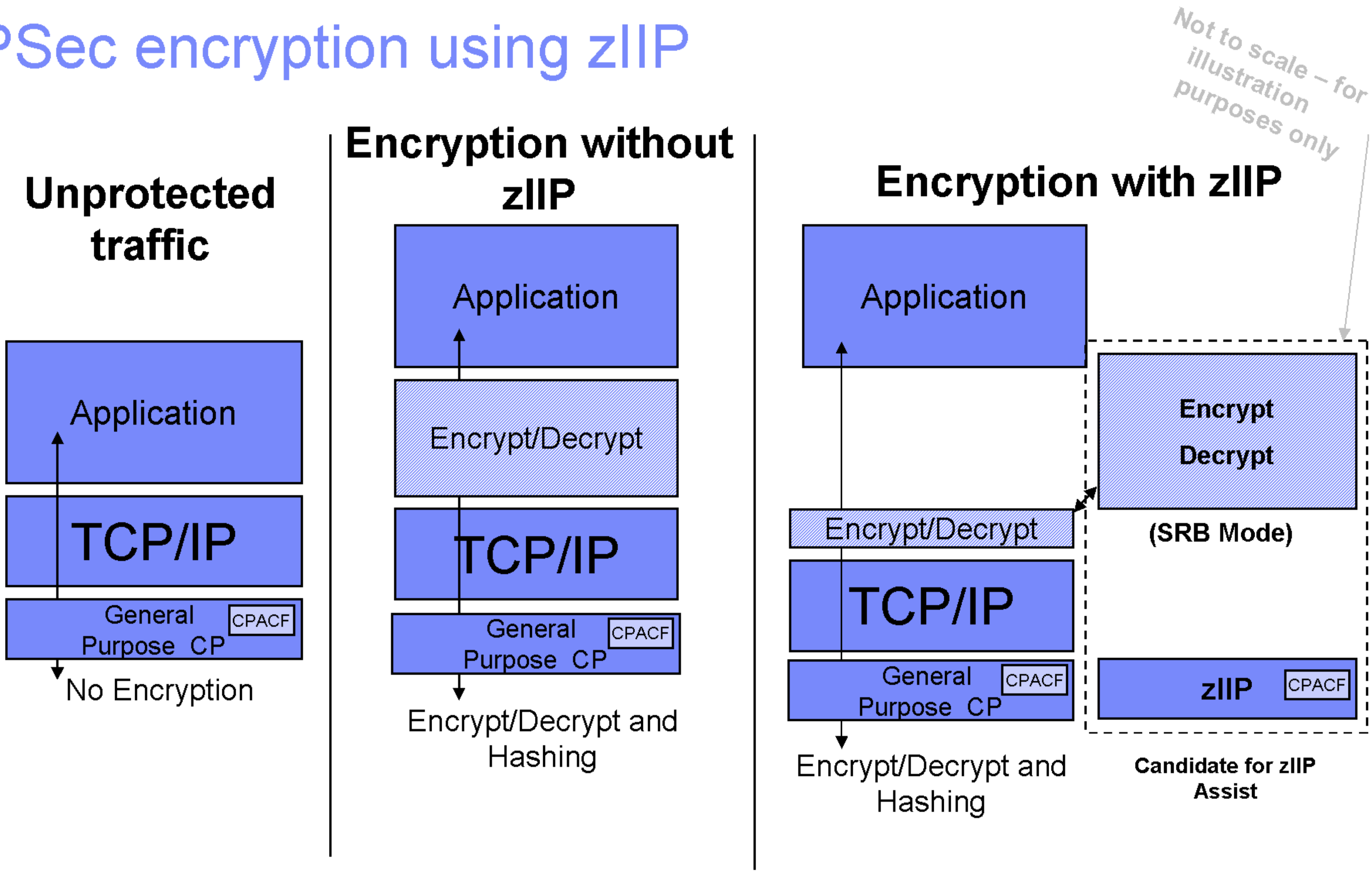
IPSec (AH, ESP) hardware crypto usage (Stack)

- **DES, 3DES, AES encryption of IKE payloads**
  - AES requires ICSF (unsupported if ICSF is not available)
- **SHA-1, MD5 HMACs for message authentication**

Algorithm	CPACF (stack doesn't use coproc'r or accel'r)
DES	In CPACF (via ICSF)
3DES	In CPACF
AES-128	In CPACF
SHA-1	In CPACF (via ICSF)
MD5	In software

- **Starting with V1R8 (APAR PK40178), all SRB-based processing in stack, *including these crypto operations*, can be offloaded to zIIP to reduce cost of IPSec protection.**

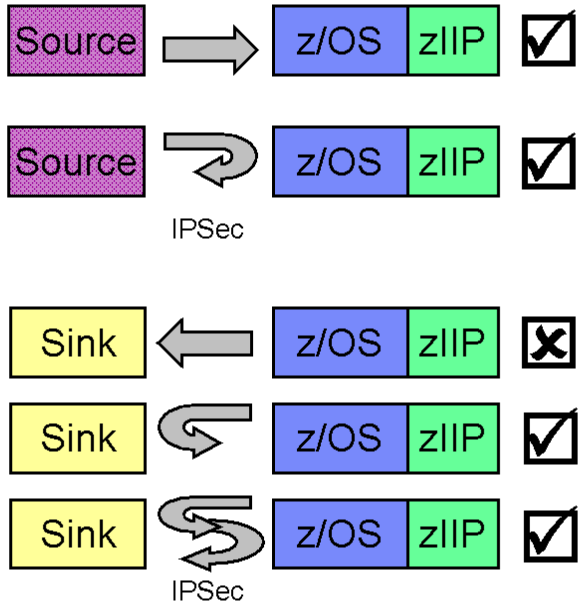
IPSec encryption using zIIP



- CPACF is exploited in the same manner on both the general CPs as well as the zIIPs
- Function is enabled via a new TCP/IP configuration keyword when zIIP hardware in place and pre-req software

What IPSec workload is eligible for zIIP?

- **The zIIP assisted IPSec function is designed to move most of the IPSec processing from the general purpose processors to the zIIPs**
- **z/OS CS TCP/IP recognizes IPSec packets and routes a portion of them to an independent enclave SRB – this workload is eligible for the zIIP**
  - Inbound operation (not initiated by z/OS)
    - All inbound IPSec processing is dispatched to enclave SRBs and is eligible for zIIP
    - All subsequent outbound IPSec responses from z/OS are dispatched to enclave SRB. This means that all encryption/decryption of message integrity and IPSec header processing is sent to zIIP
  - Outbound operation (initiated by z/OS)
    - Operation which starts on a TCB is not zIIP eligible
    - BUT… any inbound response or acknowledgement is SRB-based and therefore zIIP eligible
    - AND… all subsequent outbound IPSec responses from z/OS are also zIIP eligible



- **See session 3942 for much more on this**

Agenda

- **Review of basic cryptographic operations**
  - Symmetric cryptography
  - Asymmetric cryptography
  - Message digests and Message Authentication Codes
  - Digital certificates
- **Relevant System z and z/OS cryptographic componentry**
  - Hardware components
  - Software components
- **z/OS TCP/IP cryptography and how it uses z components**
  - SSL/TLS
  - AT-TLS
  - IPSec and IKE
- **Conclusion**

Conclusion

- System z and z/OS offer a rich set of cryptographic features
- z/OS TCP/IP support provides a rich set of secure networking protocols
- The combination of the two provides a powerful set of capabilities for securing your TCP/IP network traffic
- The combinations are numerous (and sometimes confusing ☺)
- The z platform continues to focus on improving secure TCP/IP networking performance

For more information

URL	Content
http://www.ibm.com/systems/z	IBM System z
http://www.ibm.com/systems/z/hardware/networking/index.html	IBM System z Networking
http://www.ibm.com/software/network/commserver/zos	IBM z/OS Communications Server
http://www.ibm.com/software/network/commserver/library	IBM Communications Server Library - white papers, product documentation, etc.
http://www.redbooks.ibm.com	IBM Redbooks
http://www.ibm.com/software/network/commserver/support	IBM Communications Server Technical Support
http://www.ibm.com/support/techdocs	Technical Support Documentation (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Requests For Comment (RFCs)
http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp	IBM Education Assistant

Questions?

. . .

Thank You!