# Wire-speed Cryptography for Securing Oracle SOA & Java EE Applications on Solaris

**(Emphasis on using Sun Chip Multi-threading (CMT) systems)**

Chad Prucha, Solutions Engineer
Ramesh Nagappan, Security Architect

ORACLE OPEN WORLD

# Agenda

- SOA Security : Challenges and Motivators
  - Prejudicial Barriers
  - Relevance of Cryptography in SOA
- Sun CMT and its On-chip Crypto Accelerator
  - Comparing On-chip vs. Off-chip Crypto accelerators
  - Sun CMT Crypto accelerator – How it works ?
  - Role of Solaris Cryptographic Framework (SCF)
- Enabling Crypto Acceleration for Oracle SOA
  - SOA Security: Applied Crypto Acceleration
- Realizing Wire-speed Security Performance
  - Performance studies on SSL and WS-Security scenarios
- Adopting Sun CMT Systems for Oracle SOA
  - Security, Virtualization and 10GbE networking
  - Achieving compliance goals – PCI DSS, HIPPA
  - Introduction to Sun CMT Servers family
- Call To Action
- Q & A

# Challenges and Motivators

# Security Requires a Delicate Balance

# SOA Security : Challenges and Motivators

## Fortifying SOA with Bolstering Compliance and Mitigating Risks

- Security is one of today's most critical business challenges.
  - > Greater business impacts due to increasing threats and application exploits.
  - > Increasing need for stronger access control and data security.
- Regulatory statutes enforce organizations act proactively secure information throughout its business life cycle.
  - > PCI DSS, HIPAA, FISMA, EU Data Protection and many..
  - > Mandates to enforce data confidentiality and compliance – Negligence claims leads to penalties and jail sentences !
- Predictable Scalability and Performance is critical to catering mission-critical application deployments
  - > Optimize utilization for QoS demands – ex. High availability, Reliability
  - > Deliver end-to-end security – Network, Communication, Application, Data
- Improve ROI while reducing Cost and Complexity
  - > Simplify management while lowering system acquisition and operating costs

# SOA Security : Prejudicial Barriers

- Growing IT costs and complexity to identify and defend against cyber threats.
  - > Security overheads leads to performance degradation of mission-critical applications.
    - *Cryptographic operations, Non-deterministic payloads burdens CPU and Network bandwidth.*
  - > Need for high-performance security solutions that protects application at network speed
    - *Increasing costs due to need for specialized appliances.*
- Mounting Regulatory pressures to manage and mitigate risks.
  - > Mandates organizations to ensure compliance with effective security controls.
    - *End-to-end data protection*
    - *Stronger access control*
    - *Tamper-proof audit controls.*
  - > Need to meet Compliance goals, SLAs and avoiding penalties.

# Role and Relevance of Cryptography

## SOA Security: Using Crypto for Transport/Message/Application-level Security

- Cryptographic operations plays a vital role in SOA security and trustworthy Java EE applications.
  - Confidentiality
  - Data integrity
  - Non-repudiation
  - Access Control.
- SSL/TLS has been the ***de facto*** standard for securing application-to-application communication and data in transit.
  - Use Public-key algorithms : RSA, DSA, ECC
- Securing XML Web services mandates the use of public-key encryption and digital signature services
  - To deliver XML message-level confidentiality, integrity and non-repudiation
  - Use standards such as WS-Security (XML Encryption, XML Signature), SAML 2.0, XACML, WS-Policy, WS-SecurityPolicy, WS-Trust and Liberty Alliance standards

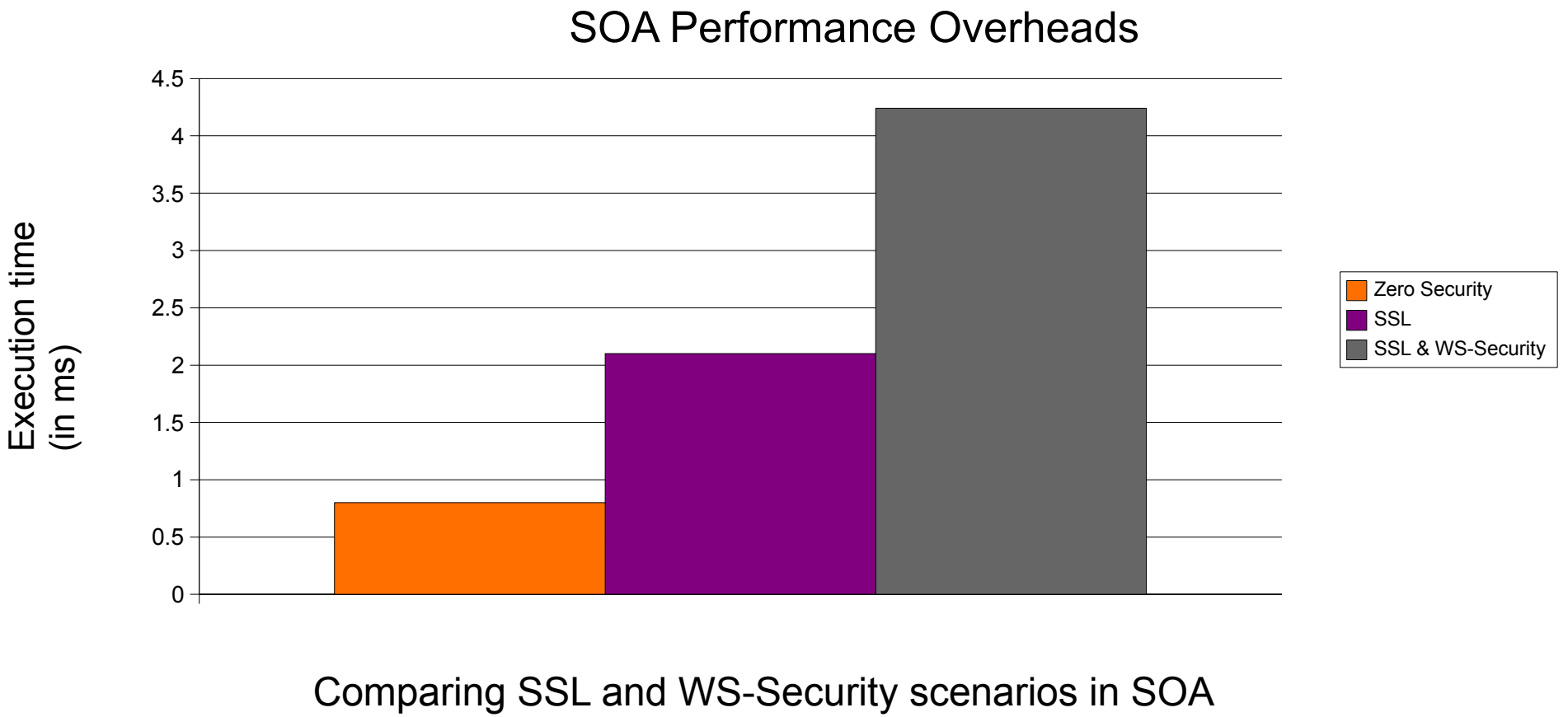
# Adopting Cryptography – Pain points

Know the stumbling issues with adopting to Cryptography

- Cryptographic functions tends to be consuming more CPU and Network bandwidth.
  - Crypto functions are usually compute-intensive operations, which taxes high CPU and Network bandwidth utilization.
- Compelling need to perform acceleration of Cryptographic operations.
  - To avoid performance degradation and meet mission-critical application requirements and SLAs.
  - Use of dedicated cryptographic appliances help eliminate performance overheads.
- Increasing costs and complexity with supporting Cryptographic operations
  - On-going acquisition and management costs
  - Integration with user applications and support virtualized deployments.

# SOA Security : Performance Overheads

## Understanding SOA performance overheads with SSL and WS-Security

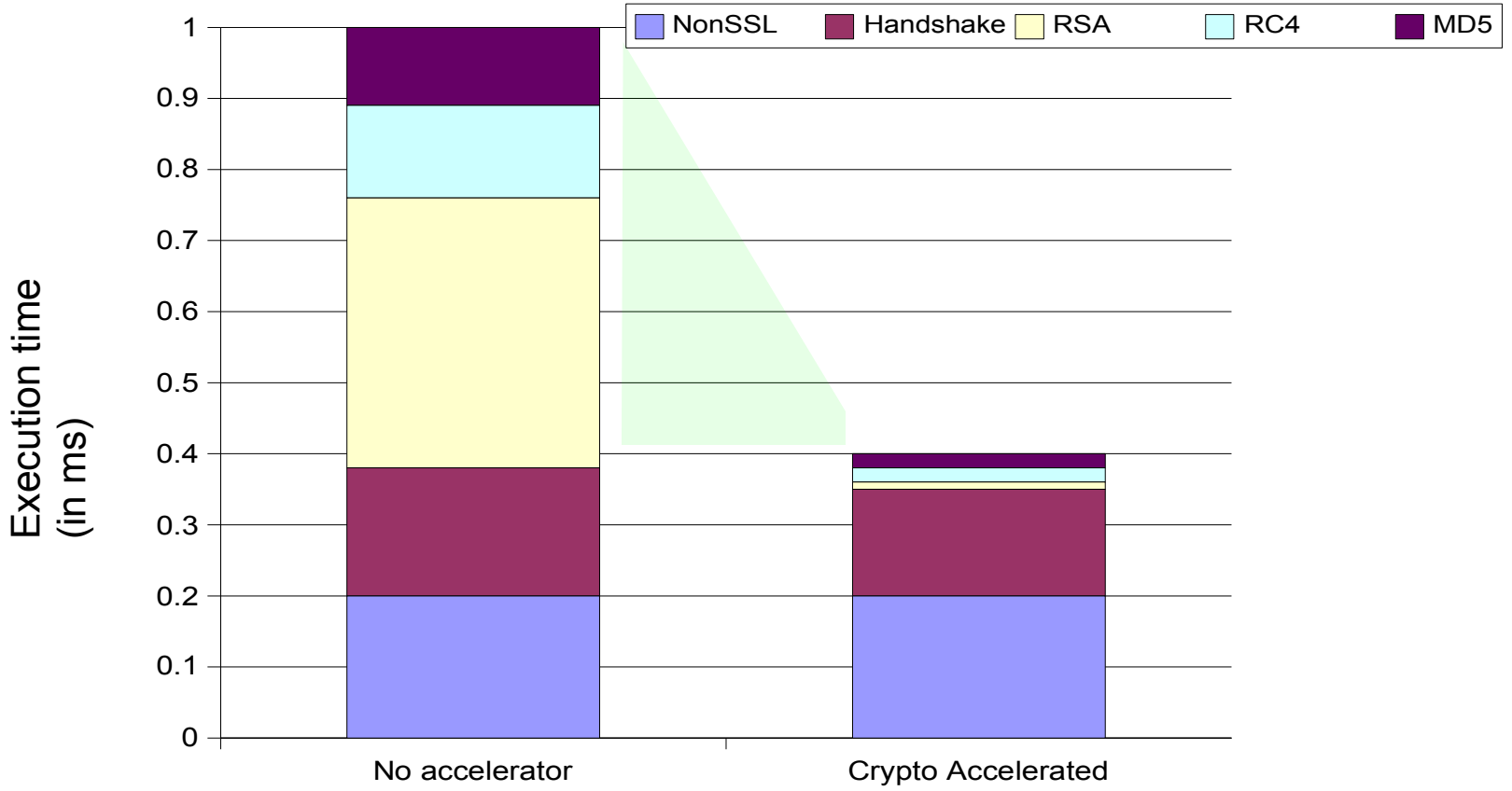



Comparing SSL and WS-Security scenarios in SOA

- Significant performance overhead occurs after introduction of SSL and WS-Security.

# Effect of Crypto Acceleration in SOA

## Understanding the overheads and relevance of crypto acceleration



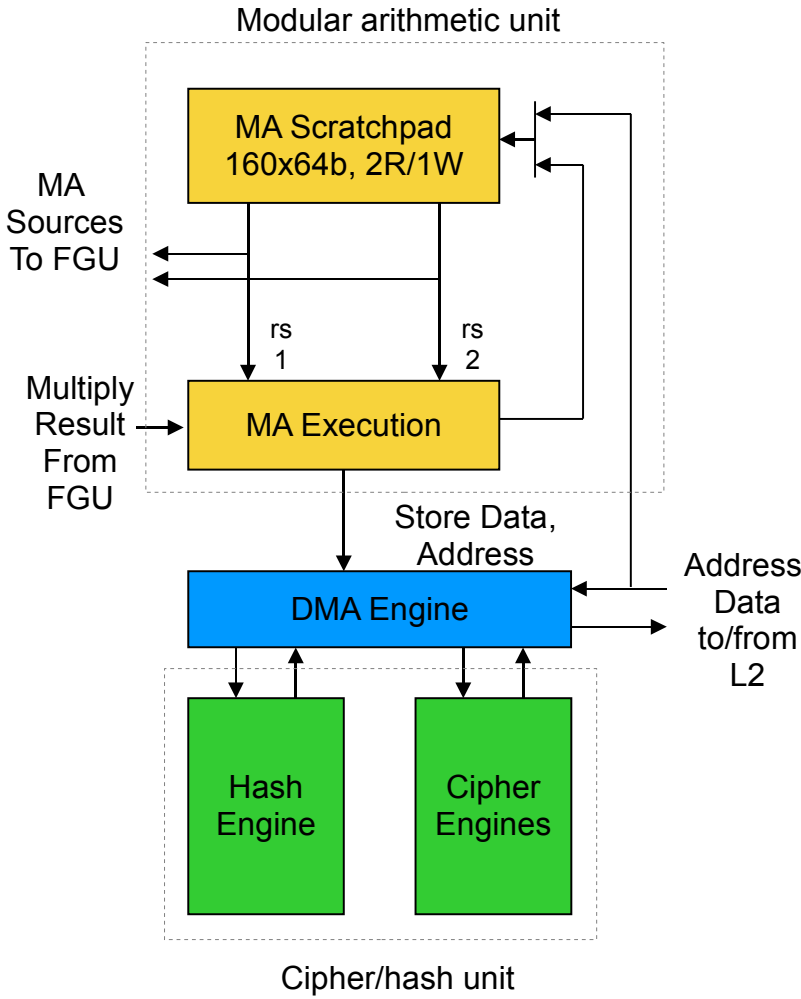

Comparing SSL scenarios w. Cryptographic Acceleration in SOA

- Performance gains can be achieved ONLY by using hardware-based cryptographic acceleration.

# Delivering Sun CMT Based On-Chip Cryptographic Acceleration

# Sun Chip Multithreading Technology (CMT)

- Multi-core & Multi-threaded processor
  - 8 Cores/chip & 8 Threads/Core
    - *Available as part of UltraSPARC T1/T2 based Sun Servers*
- Industry's first "System on Chip" processor technology
  - Integrates computing, networking and security on a single chip.
- Built-in Crypto Accelerator per Core.
  - 8 crypto accelerators per chip
  - Composed of two independent units
    - *Modular Arithmetic Unit (MAU) and Cipher/Hash Unit*
  - Runs in parallel at core CPU speed and offloads target cryptographic operations from CPU.
    - *Performs public-key encryption, bulk encryption, hashing and random functions with CPU bus speed*




**Sun CMT Servers deliver Wire-speed Crypto Acceleration**

# CMT Crypto Accelerators and its Ciphers

## Understanding Sun CMT processors and supporting Ciphers


- UltraSPARC T1 Processor
  - > First generation CMT processor that introduced built-in Cryptographic accelerator
  - > Capable of accelerating public-key encryption operations.
    - – *RSA, DSA, Diffie-Hellman*
- UltraSPARC T2 Processor
  - > Second-generation CMT processor
  - > Crypto accelerators are enhanced to support more cryptographic operations.
    - – *Bulk encryption (RC4, DES, 3DES, AES)*
    - – *Message digests (MD5, SHA-1, SHA-256)*
    - – *Additional public-key encryptions (ECC)*
- Both T1 and T2 provide Light-weight accelerator drivers for Solaris.
  - > NCP, N2CP and N2RNG drivers available on Solaris
  - > Stateless communication just Fire and Forget – Consumer application is informed when operation is complete.

# Sun CMT On-Chip Vs Off-chip Crypto Accelerators

## Comparison : Sun Onchip Crypto with Competition Off-chip Accelerators

### Sun On-Chip Accelerator

- **Zero-cost Security**
  - No additional investment
  - No installation and tuning
  - Minimal configuration
- **Runs in parallel with CPU speed**
  - Offloads target crypto overheads efficiently
  - Object and session size does'nt matter – effective on all
- **Non-Intrusive & Ready-to-use with applications**
  - PKCS11 and Solaris Crypto
  - Kernel SSL support
  - Virtualization support

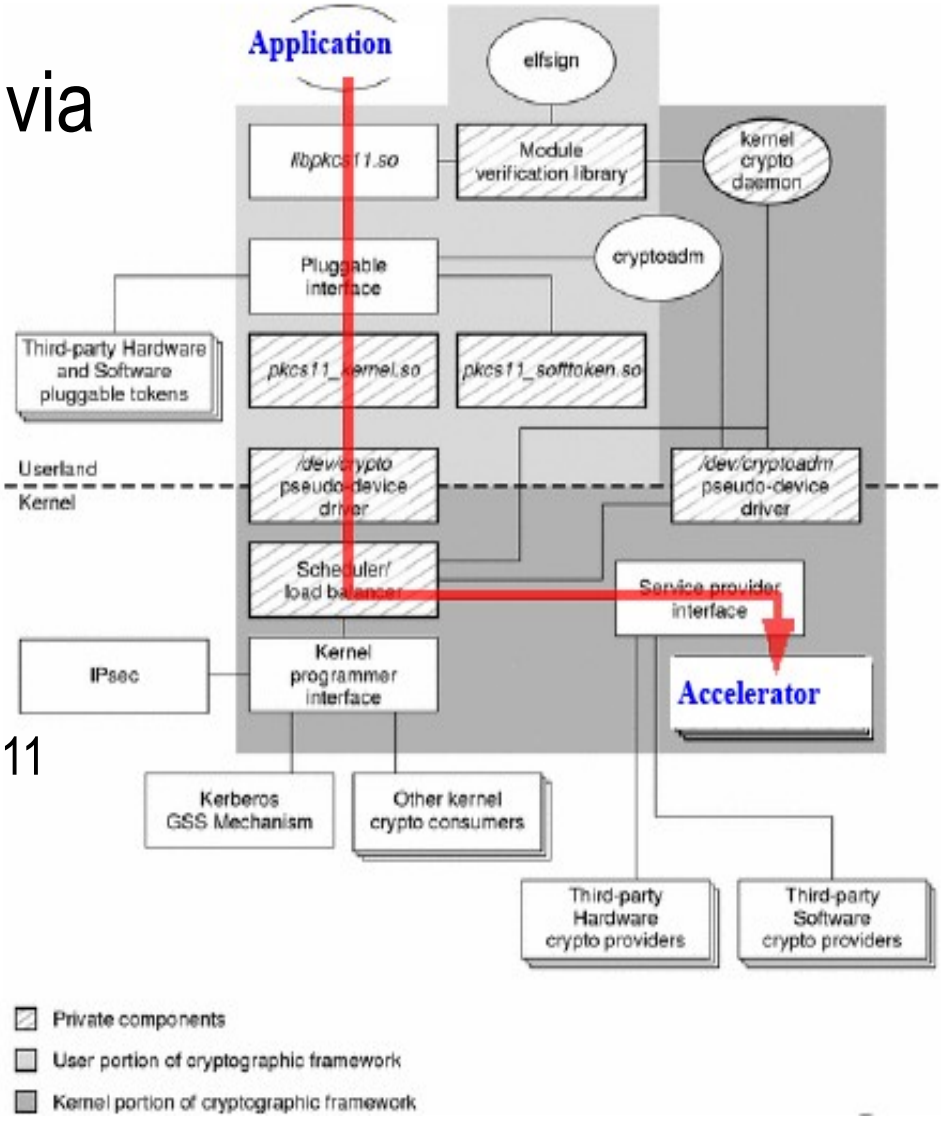
### Off-Chip Accelerator

- **Additional Costs incurred**
  - Cost per accelerator
  - Installation and Maintenance required
  - Extensive configuration and testing required
- **Runs as add-on PCI-E device/appliance**
  - Not effective on smaller object offloads
  - Limited to No. of SSL sessions or memory size
- **Custom integration required**
  - Needs driver configuration and device mapping
  - No out-of-box virtualization

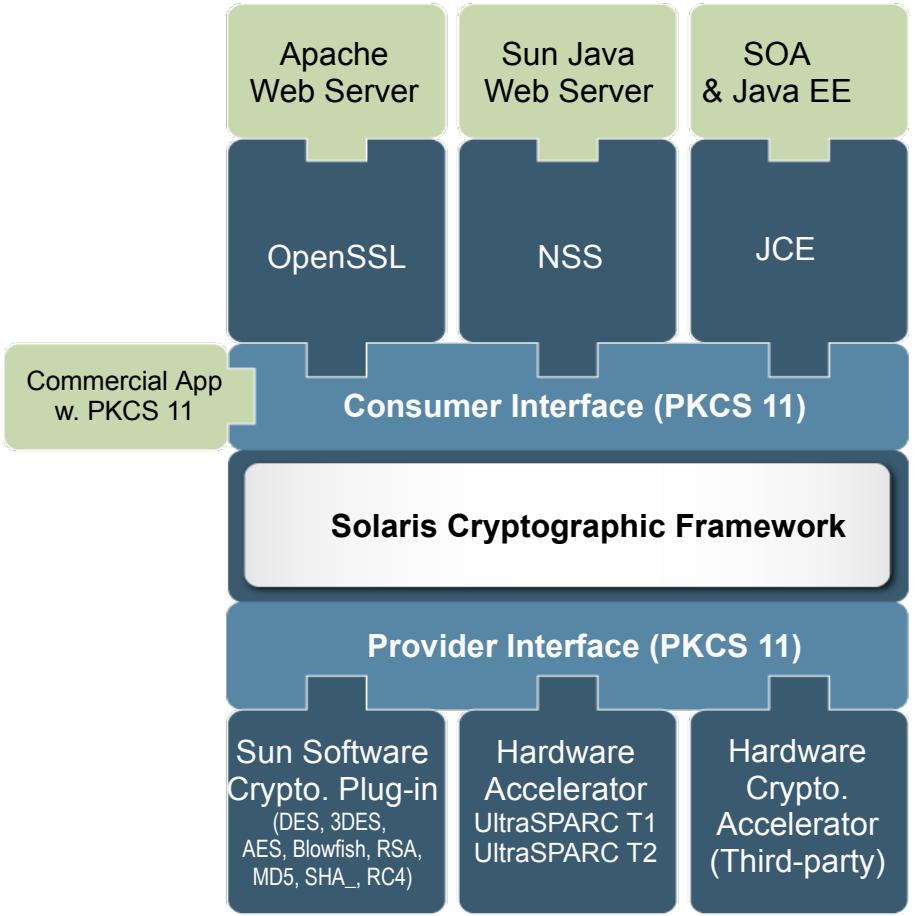
# CMT Crypto Acceleration: How it works ?

## Operational model of Sun CMT based Cryptographic Acceleration

- Access to CMT cryptographic acceleration provider is controlled via Solaris Cryptographic Framework (SCF).
  - > Applications can access accelerator via PKCS11 standard interfaces
    - *Most applications can use Solaris SunPKCS11 provider.*
    - *SOA and Java EE applications can access via JCE (Java SunPKCS11 provider)*
    - *OpenSSL interfaces also supported*
  - > All requests from user application traverses from userland applications to accelerator via SCF PKCS11 libraries
- Solaris kernel modules can communicate directly with accelerator using SCF.
  - > ex. KSSL and IPSec drivers support



# Solaris Cryptographic Framework (SCF)




- Common framework for providing cryptographic services for Solaris applications and users
  - PKCS11 interfaces for consumers and providers
  - Allows performing, consuming and integrating cryptographic operations and providers.
    - *Kernel or userland providers*
    - *Hardware or software based (JCE, NSS, OpenSSL, Files and PKCS11)*
  - Implements major Ciphers and algorithms
    - *AES, Blowfish, RC4, DES, 3DES, RSA*
    - *MD5, SHA-1, SHA-256, SHA-384, SHA-512*
    - *DES MAC, MD5 HMAC, SHA-1 HMAC, SHA-256 HMAC, SHA-384 HMAC, SHA-512 HMAC*
  - Key Management
  - Optimized for SPARC, Intel and AMD processors

# Solaris Kernel SSL (KSSL)

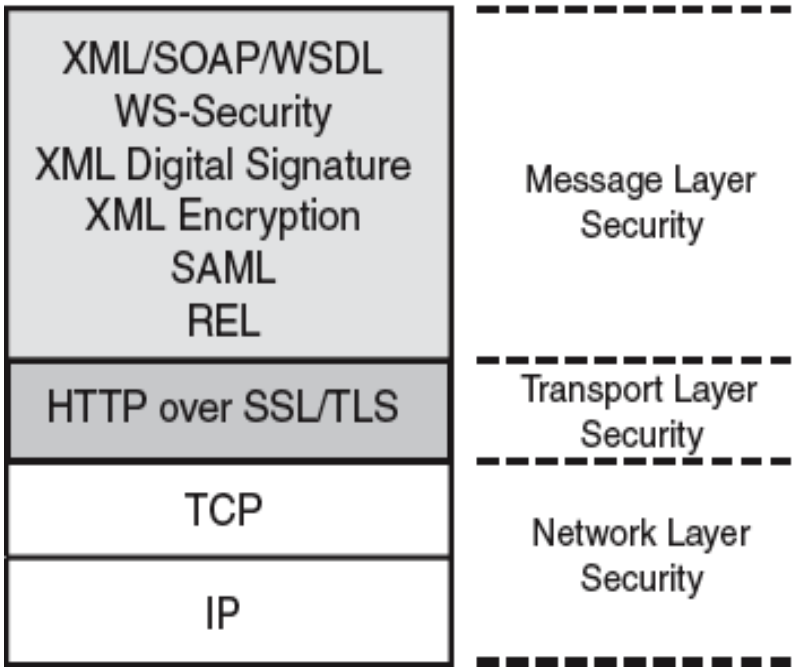
- Solaris KSSL
  - Facilitates an SSL Proxy service for applications and performs SSL operations right in the Solaris kernel.
  - Integral part of Solaris Cryptographic Framework (SCF) and makes use of its SSL/TLS cipher suites.
  - Supports using hardware-based cryptographic accelerators and HSMs (via PKCS11) for Private key storage.
    - *Can use non-extractable RSA private keys stored in HSM*
  - Non-intrusive SSL configuration independent of applications.
    - *Managed via Solaris Service Management Facility (SMF)*
    - *ksslcfg to create and configure KSSL SMF service*
    - *FMRI is svc:/network/ssl/proxy*
  - Can act as a SSL proxy for both SSL and Non-SSL capable applications.
  - Delivers 20% - 35% faster SSL performance in comparison with traditional applications managed SSL
    - *Kernel consumers tends to have less overhead when using hardware accelerators*

# Sun CMT Cryptographic Acceleration for Oracle SOA/XML Web Services and Java EE Applications

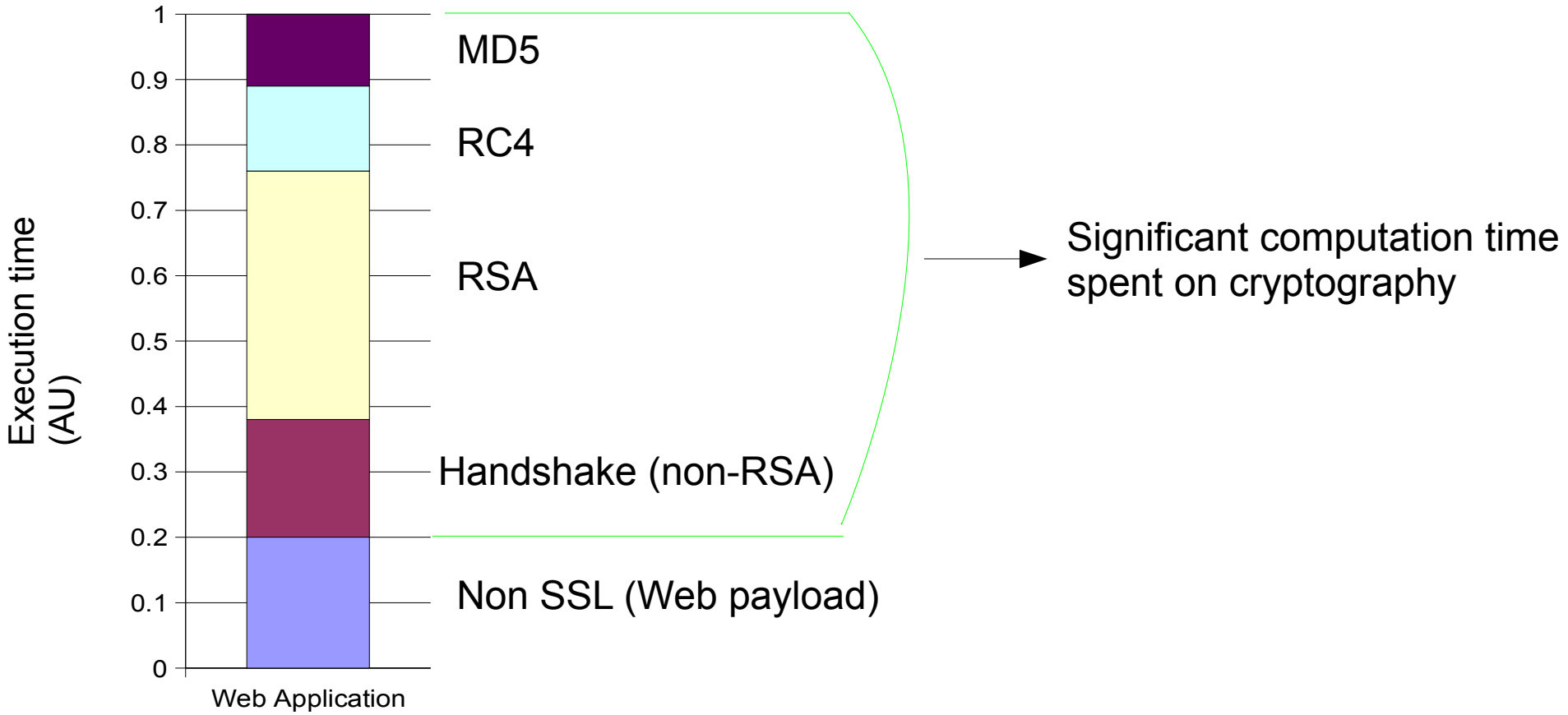
# Accelerating SOA Security: Ground Up

## Applied SOA Security Usecases with Sun CMT Crypto Acceleration

- Message-layer Security
  - WS-Security (XML Encryption and XML Signature)
    - *Use WS-Policy/WS-SecurityPolicy and enable JCE/SunPKCS11 provider configuration for offloading to CMT acceleration*
- Transport-layer Security
  - SSL/TLS
    - *Option 1: Use KSSL as SSL Proxy*
    - *Option 2: Use Application Server managed SSL and then enable JCE/SunPKCS11 provider configuration for offloading to CMT acceleration*
- Network-layer Security
  - IPSec enabled
    - *Follow Sun CMT driver configuration guide for IPSec*



# Anatomy of an SSL Scenario in SOA



# Option 1: Solaris KSSL as SSL Proxy

## Non-invasive way for enabling SSL with Sun CMT Crypto Acceleration

1. Obtain your SSL certificate from your CA
   - Make sure the certificate artifacts (including CA certs) are available in a single file or a PKCS11 store.
   - Certificates may need to be in PKCS#12 or PEM formats.
2. Configure the KSSL proxy and its redirect HTTP/Cleartext port
3. Verify KSSL using Solaris SMF
4. Make sure your application/web server listens to the KSSL redirect port
5. Test for SSL interaction with your target Web server

# Option 1: Solaris KSSL as SSL Proxy - Quick Configuration

1. Obtain your SSL certificate
   - For example using OpenSSL:
     - openssl req -x509 -nodes -days 365 -subj "/C=US/ST=Massachusetts/L=Burlington/CN=myhostname"
       -newkey rsa:1024 -keyout /etc/pki/mySSLKey.pem -out /etc/pki/mySSLServerCert.pem
     - KSSL requires all certificate artifacts in a single file (in case of file based keystore, concatenate them to a single file), otherwise import your certificates to a PKCS#11 keystore.
2. Configure the KSSL proxy and its redirect HTTP/Cleartext port
   - ksslcfg create -f pem -i /etc/pki/mySSLCerts.pem -x 7001 -p /etc/pki/passwordfile myhostname 443
   - 7001 is the cleartext port (Your Weblogic application server listens)
3. Verify KSSL using Solaris SMF
   - svcs -a | grep "kssl"
4. Make sure your application/web server listens to the KSSL redirect port
   - Test drive https://myhostname.com:443/

# Option 2: SSL Acceleration for Weblogic

## Configuring Weblogic SSL and offload to Sun CMT Crypto Acceleration

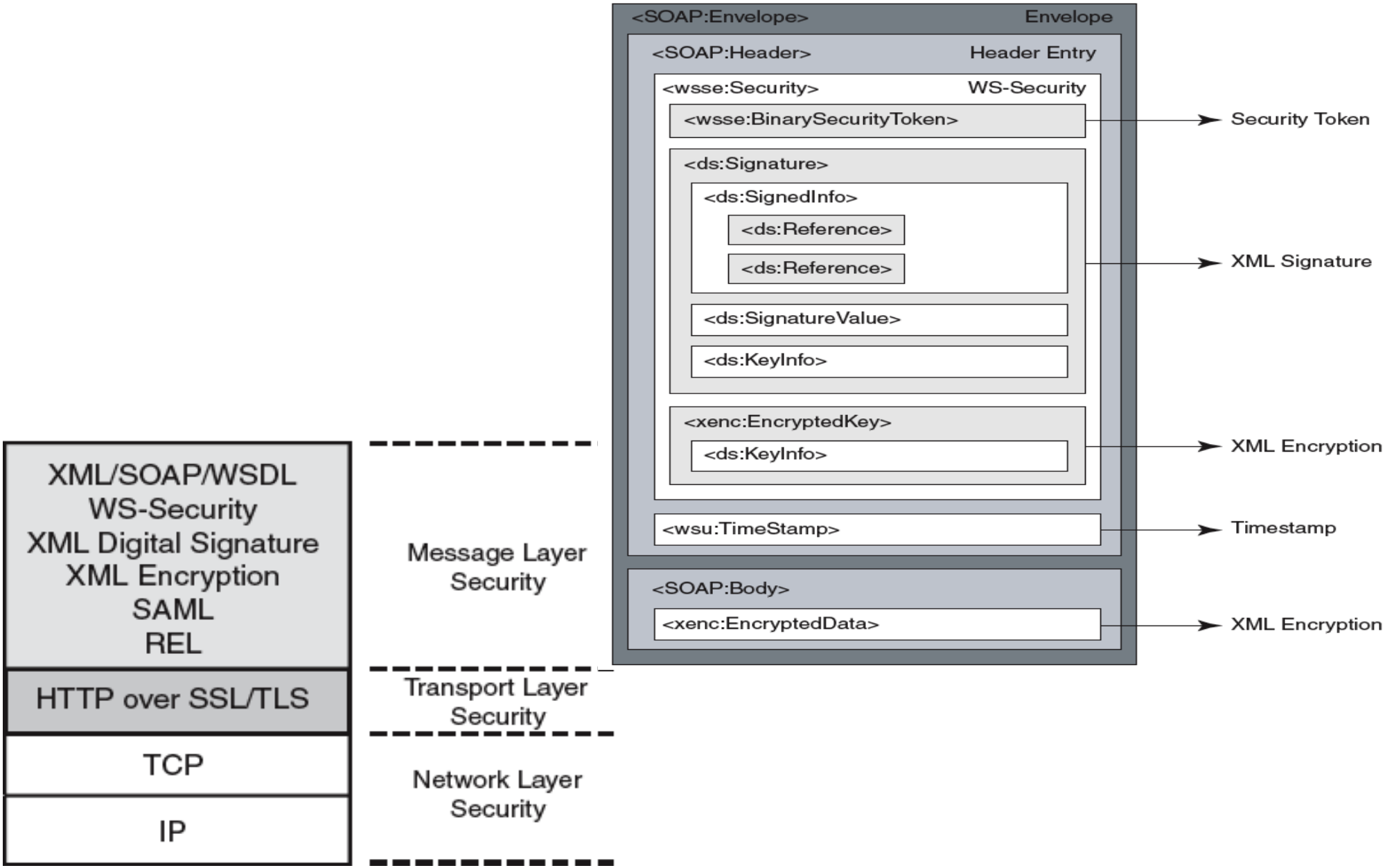
1. Setup SSL listener for your Weblogic Server instance
   - Follow your Admin guide instructions for configuring SSL
   - Install the SSL certificates
2. Enable cryptographic acceleration for Weblogic SSL by editing JRE's SunPKCS11 provider configuration.
   - SunPKCS#11 provider is a generic provider to utilize any PKCS11 provider implementation.
   - The sunpkcs11 configuration file contains the attributes for accessing the hardware accelerator.
     - *Located at <weblogic-java-home>/jre/lib/security/sunpkcs11-solaris.cfg*
   - Mechanisms/attributes supported by the underlying hardware accelerator can be enabled or disabled at SunPKCS11 configuration file.
     - *Include the RSA mechanisms in disableMechanisms list of SunPKCS11 softoken.*
     - *Helps to force those RSA mechanisms performed by NCP (Sun CMT accelerator)*
3. Restart the Weblogic server instance.

# Example: SunPKCS11 Provider configuration

## Disabling Soft-token and enabling RSA mechanisms to use HW accelerator

```
name = Solaris
description = SunPKCS11 accessing Solaris Cryptographic Framework
library = /usr/lib/$ISA/libpkcs11.so
handleStartupErrors = ignoreAll
attributes = compatibility
disabledMechanisms = {
  CKM_MD2
  CKM_MD5
  CKM_SHA_1
  CKM_SHA256
  CKM_SHA384
  CKM_SHA512
  CKM_DSA_KEY_PAIR_GEN
  CKM_SHA1_RSA_PKCS
  CKM_MD5_RSA_PKCS
  CKM_DSA_SHA1
  CKM_TLS_KEY_AND_MAC_DERIVE
  CKM_RSA_PKCS_KEY_PAIR_GEN
  CKM_SSL3_PRE_MASTER_KEY_GEN
  CKM_SSL3_MASTER_KEY_DERIVE
  CKM_SSL3_KEY_AND_MAC_DERIVE
  CKM_SSL3_MASTER_KEY_DERIVE_DH
  CKM_SSL3_MD5_MAC,CKM_SSL3_SHA1_MAC
}
```

# Anatomy of WS-Security Scenario



# Enforcing WS-Security in Oracle SOA

- Oracle Fusion Middleware builds on Oracle Weblogic 10.3.x for implementing WS-Security 1.1
  - X.509 certificates to sign and encrypt a SOAP message
  - SOAP message targets (SOAP Body, Headers, Elements) are signed and encrypted.
  - Authentication token support – username/password, SAML, X.509
- Allows representing WS-Security scenarios using pre-defined WS-Policy and WS-SecurityPolicy based assertions.
  - Based on OASIS WS-SecurityPolicy 1.2 and WS-Policy 1.2 specifications
  - Applications use Java annotations to configure security policies
  - Attach a relevant WS-Policy to define a WS-Security scenario
    - ex. ***@Policy(uri=policy:Wssp1.2-2007-Wss1.0-UsernameToken-Plain-X509-Basic256)***
    - Refers to the WS-Policy including WS-SecurityPolicy
      - Service to authenticate the client with a username token
      - Both request and response messages are encrypted + signed with X509 certificates.
      - Basic256 identifies the cipher algorithm suite to use.
    - Alternatively, you may use JAX-WS (Metro) for attaching WS-Policy ( via Netbeans IDE).

# Accelerating WS-Security : Configuration

1. Identify the algorithm suite used in the WS-Policy
   - For example: Basic128Sha256Rsa15 refers to
     - Encryption algorithm: AES 128
     - Digest algorithm: SHA256
     - Symmetric Key Wrap: KwAes128
     - Asymmetric Key Wrap: KwRSA15
     - Signature Key Derivation: Psha1L128
2. Install keys and certificates in Java keystore or your HSM.
3. Disable mechanisms in the Java SunPKCS11 provider configuration file, to force those operations performed by NCP and N2CP (Sun CMT accelerators)
   - *Edit the SunPKCS11 provider configuration file*
     - *Located at <weblogic-java-home>/jre/lib/security/sunpkcs11-solaris.cfg*
   - *Force the RSA and AES mechanisms to use NCP and N2CP by including them in disableMechanisms list of softtoken.*

# Solaris Crypto Admin commands

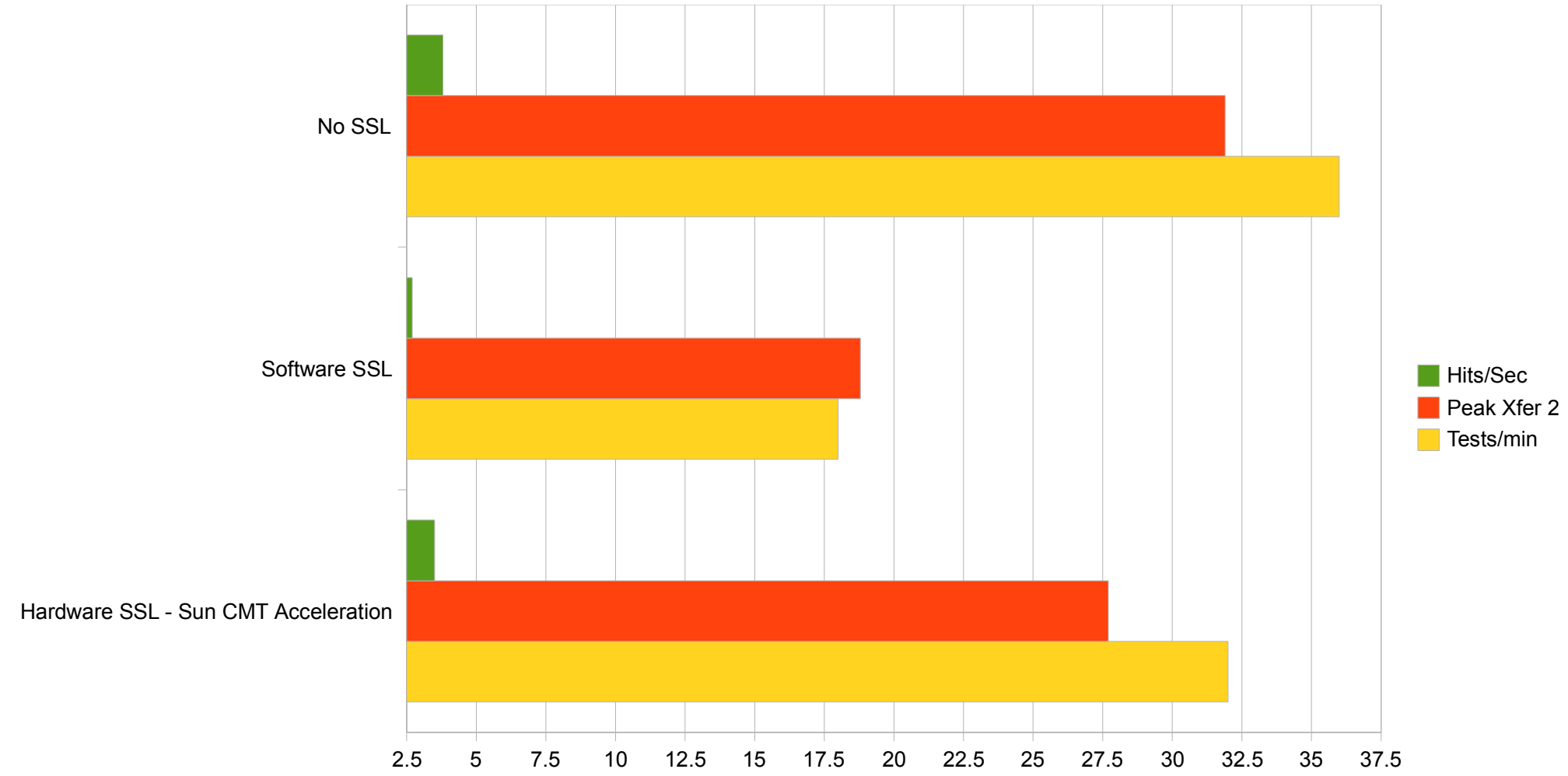
## Provider administration and Kernel Statistics

- Crypto Provider Administration
  - To display the list of providers installed
    - *cryptoadm list -p*
  - To display the list of cryptographic mechanisms supported by the provider
    - *cryptoadm list -m*
  - To install the softtoken provider implementation
    - cryptoadm install provider=/usr/lib/security/\$ISA/pkcs11_softtoken.so
  - To disable the selected mechanisms from the softtoken provider
    - cryptoadm disable provider=/usr/lib/security/\$ISA/pkcs11_softtoken.so mechanism=<.....>
  - To enable the selected mechanisms for the softtoken provider
    - *cryptoadm enable provider=/usr/lib/security/\$ISA/pkcs11_softtoken.so mechanism=<.....>*
- Kernel Statistics
  - To report the kernel statistics of NCP module
    - *kstat -n ncp0*
  - *To report the kernel statistics of N2CP module*
    - *kstat -n n2cp0*

# Wire-speed SOA Security using Sun CMT

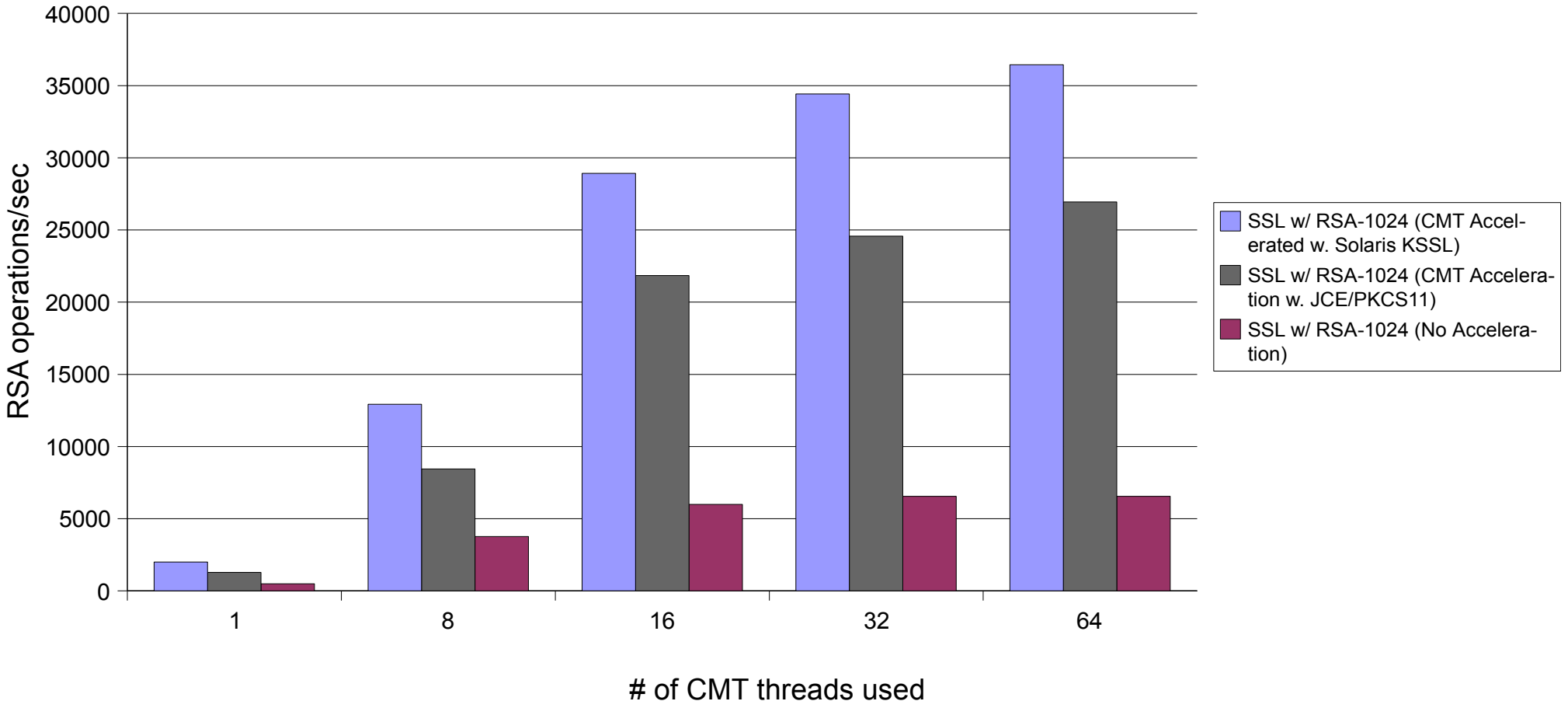
# Performance Studies

# Sun CMT based SSL Acceleration for Oracle Weblogic : Quick Look



# Weblogic SSL Performance on Sun CMT

## Predictable SSL performance with/with-out Sun CMT Crypto Acceleration




Using Sun CMT for Weblogic SSL: Comparative Study

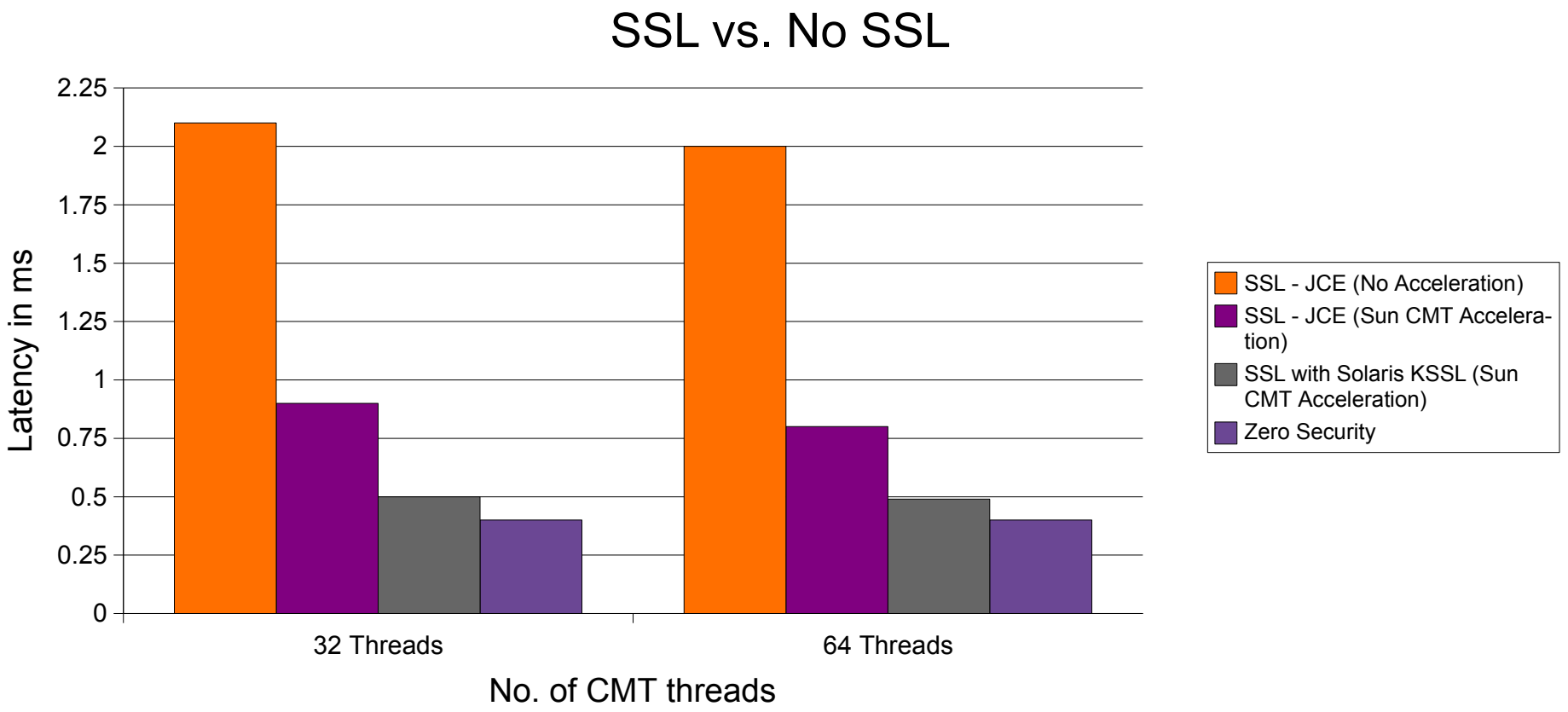
Solaris KSSL vs. Sun JCE vs. No Acceleration

on

Sun SPARC Enterprise T5440

# Effect of Weblogic SSL vs. No SSL on Sun CMT

## Throughput performance on Sun CMT



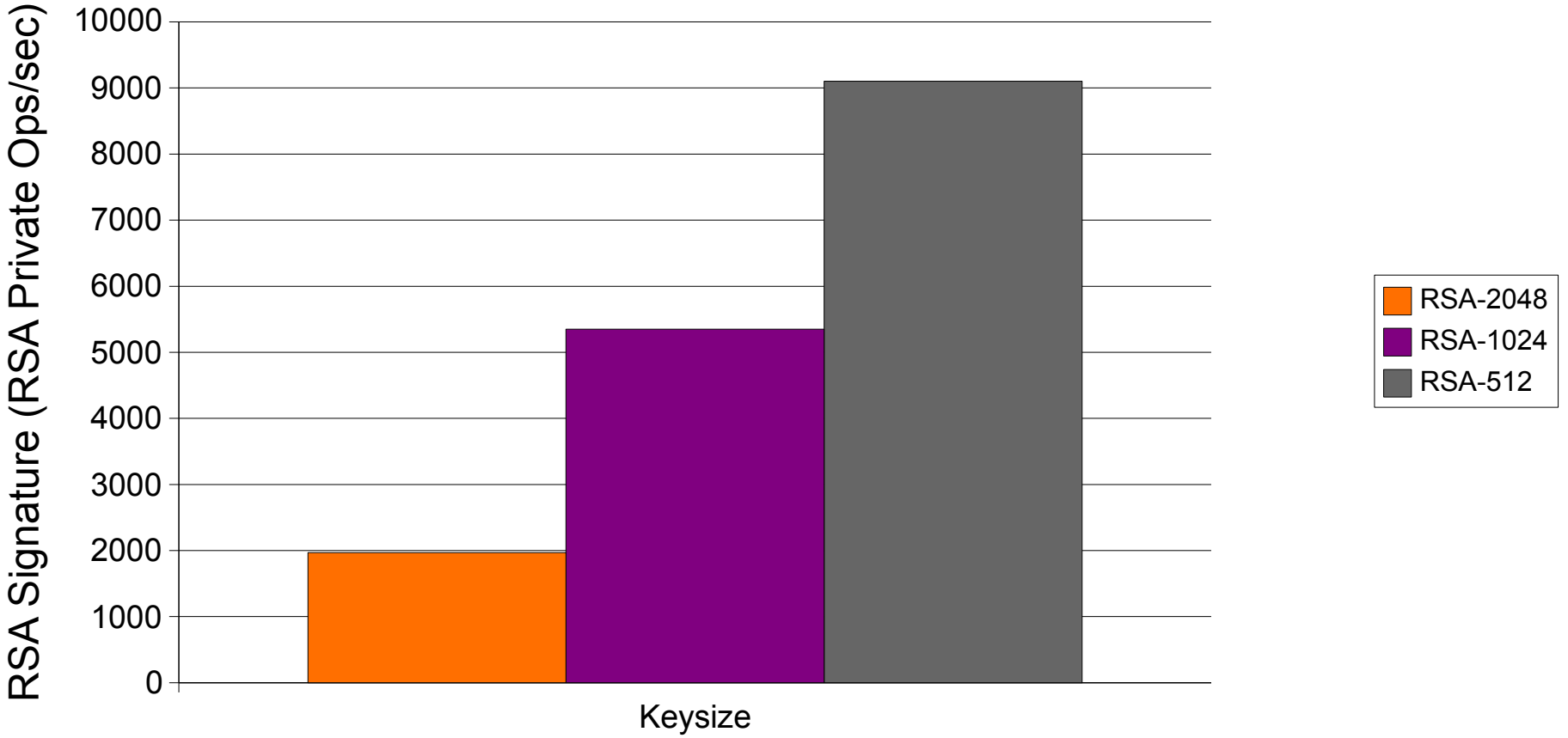

Using Sun CMT for Weblogic SSL: Comparative Study

Solaris KSSL vs. Sun JCE vs. No Acceleration

on

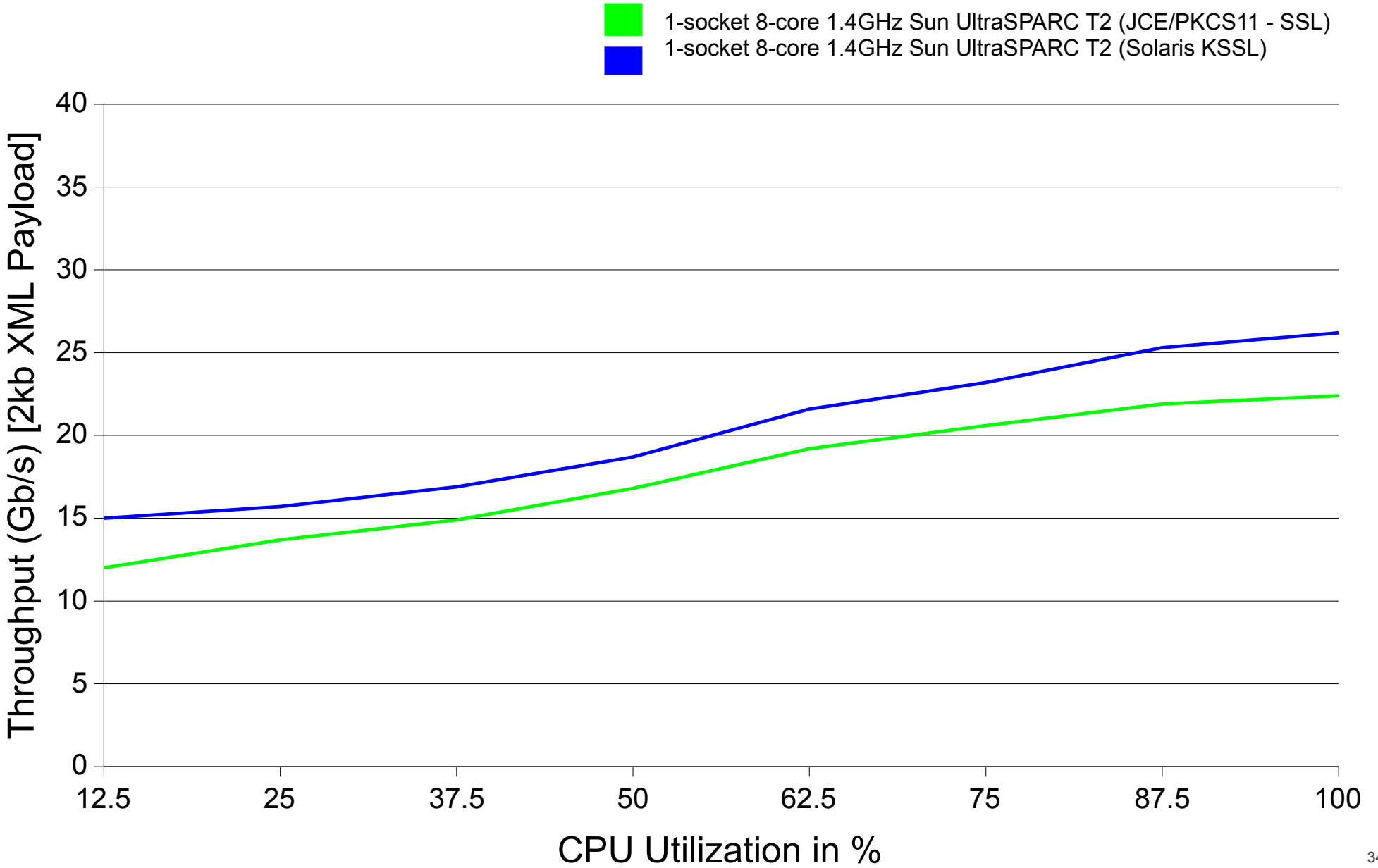
Sun SPARC Enterprise T5440

# SOA: WS-Security Performance (XML Signature)




WS-Security (XML Signature) Performance
(Using Basic128Sha256Rsa15 Algorithm Suite in WS-Policy)
on
Sun SPARC Enterprise T5440

# SOA Security: SSL and WS-Security Combined



# UltraSPARC T2 Processor Performance

## Peak Cryptographic Performance

### Bulk Cipher

| Algorithm | Gb/s/chip |
|---|---|
| RC4 | 83 |
| DES | 83 |
| 3DES | 27 |
| AES-128 | 44 |
| AES-192 | 36 |
| AES-256 | 31 |

### Secure Hash

| Algorithm | Gb/s/chip |
|---|---|
| MD5 | 41 |
| SHA-1 | 32 |
| SHA-256 | 41 |

### Public key

| Algorithm | Ops/sec/chip |
|---|---|
| RSA-1024 | 37K |
| RSA-2048 | 6K |
| ECCp-160 | 52K |
| ECCb-163 | 92K |

- Accelerators support most common ciphers, hashes and modes of operation


Achieving
Compliance Objectives
PCI-DSS
&
HIPPA
Scenarios

# Addressing PCI-DSS Checklists

## Adopting Sun CMT for achieving PCI-DSS goals

- Sun CMT can contribute to core PCI-DSS requirements
  - > Requirements 1 through 9
- PCI-DSS Section 2.3
  - > Encrypt all administrative access interfaces
    - – *Use SSH, VPN, SSL/TLS based administrator interactions*
- PCI-DSS Section 4.1
  - > Use Strong cryptography and security protocols such as SSL/TLS or IPSec to safeguard sensitive cardholder data during transit over public networks.
    - – *Use SSL/TLS and IPSec for securing transmission over public networks*

# Addressing HIPPA Compliance

## Adopting Sun CMT for achieving HIPPA

- Sun CMT can contribute to HIPPA data confidentiality requirements
  - > CFR 63, No 155, 43255
- Guard against unauthorized access that is transmitted over a communication network
  - > Data confidentiality, Integrity controls
  - > Message authentication
    - *Encryption and Digital signature mechanisms for LANs or ...*
    - *Private-wire exception*

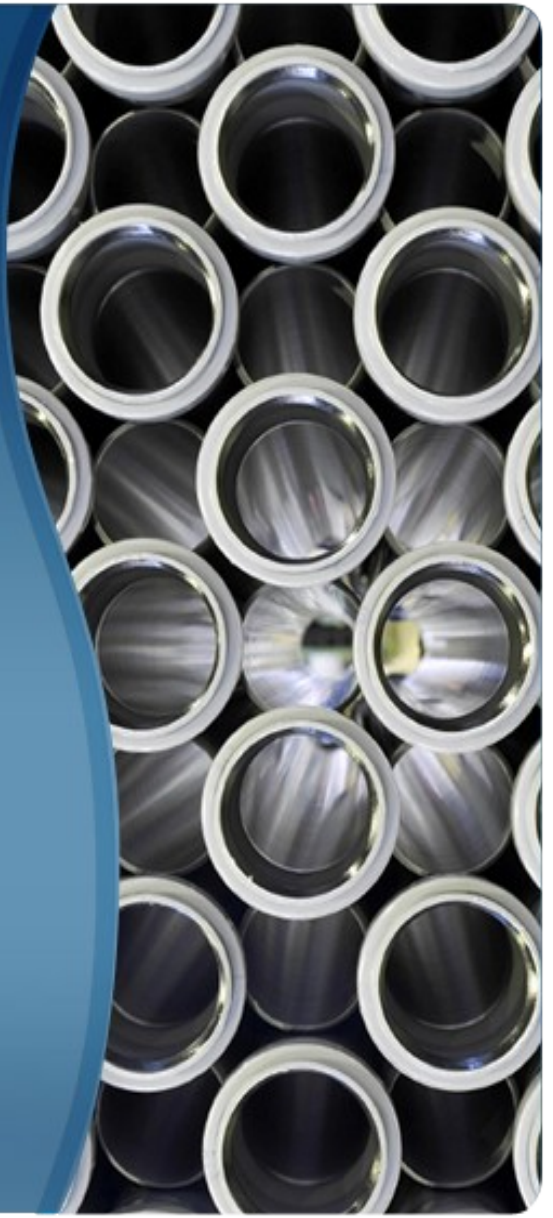

Adopting
Sun CMT Servers

# Built-in, On-Chip Wire-speed Security


- Save Money
  - > Don't pay extra for a separate cryptographic processor, and keep your PCI-Express slots free for other uses
- Highest Security with minimal performance impact
  - > Supports ten most common ciphers and secure hashing functions, including NSA approved algorithms
  - > Enable SSL, WS-Security and IPSec
    - > Securing Web applications, servers, networks, filesystems
- Faster Performance
  - > Outperforms competing accelerators by more than 10x
  - > Avoid the performance penalty previously associated with secure operation

# Revolutionary Multi-Threaded Networking

- Integrated 10 gigabit ethernet (10GbE) on the UltraSPARC T2 processor
- 10GbE on the motherboard of T2 Plus servers
- Save Money
  - Add low-cost XAUI interface cards instead of costly 10GbE NICs
- Faster Performance
  - Delivers up to 4x the performance of current network interface cards
  - Total bandwidth nearly 40 Gb/sec.
  - On-chip network interface reduces bottlenecks, enables faster network access




Introducing
Sun CMT Family

# CMT Product Line

## Dramatically Changing Your Business Application ROI

Sun Blade™ T6340 and T6320

Sun SPARC Enterprise T5220

Sun SPARC Enterprise T5120

Sun SPARC Enterprise T5240

Sun SPARC Enterprise T5140

Sun SPARC Enterprise T5440

# Summary & Call To Action

# Call To Action

- Visit the Sun booth in Moscone South #1101
  - See the Sun Storage and Server portfolio in person
  - View Sun Oracle solutions that bring Extreme Innovation to the Enterprise
  - Talk to Sun experts and leave with answers
  - Get briefed on next-gen Sun Storage and Servers under NDA
- After the show...
  - Feel free to contact us for more information
  - Visit sun.com for our Blueprint on this topic

# Thank You

**Chad Prucha**
**Chad.Prucha@sun.com**
**http://blogs.sun.com/soyuz**

**Ramesh Nagappan**
**Ramesh.Nagappan@sun.com**
**http://www.coresecuritypatterns.com/blogs**