ORACLE

SPARC SERVERS

An Oracle White Paper
May 2013

High-Performance Security for Oracle WebLogic Server Applications Using Oracle's SPARC T5 and SPARC M5 Servers

ORACLE

Introduction

This document details the high-performance security strategies for Oracle WebLogic Server applications and XML Web services using the on-chip/on-core cryptographic acceleration capabilities of Oracle's SPARC T5 and Oracle's SPARC M5 servers. This document presents the technical prerequisites, configuration, deployment, and verification guidelines for Oracle WebLogic Server, Java Runtime Environment, and the Oracle Solaris Cryptographic Framework features. These products support the cryptographic operations involved with encryption/decryption, digital signature, key management functions of SSL, and WS-Security application scenarios and using encrypted ZFS data sets (file systems and ZVOLs).

The hardware-assisted cryptographic acceleration strategies and applied techniques presented in this document have been tested and verified for use with Oracle WebLogic Server 12*c*-based application deployments on Oracle's SPARC T5 and Oracle's SPARC M5-32 servers.

This document is intended for security administrators and Oracle WebLogic Server administrators who are tasked to deploy and integrate the on-chip cryptographic capabilities of Oracle's SPARC T5 processor-based servers. Administrators should be familiar with the installation and configuration of Oracle's SPARC T-Series and SPARC M-Series servers, Oracle Solaris 11.1, including the use of ZFS encryption, as well as Oracle WebLogic Server 12*c* and applied techniques for enabling SSL and WS-Security for Oracle WebLogic Server applications.

Role and Relevance of Oracle's SPARC T-Series Processor Family

As security has taken unprecedented importance in all facets of the IT industry, organizations are proactively adopting cryptographic mechanisms to protect their businesses and information from unauthorized access and ensuring data confidentiality and integrity during transit and in storage. Cryptographic operations are heavily compute-intensive, burdening the host system with additional CPU cycles and network bandwidth resulting in significant degradation of overall throughput of the system and its hosted applications. For example, a host server capable of processing 1,000 transactions per second can perform only 10 transactions per second after deploying SSL to secure communications with the hosted application.

To speed up cryptographic performance, security experts often recommend and use cryptographic accelerator appliances to offload cryptographic operations and save CPU cycles, enhancing the system throughput and its hosted applications. While useful, adopting a specialized appliance for offloading cryptographic operations introduces a new set of costs, complexities, and issues in terms of procurement, installation, configuration, testing procedures, management, and support that significantly increases the power demands and costs of deployment projects.

Foreseeing the need for special-purpose hardware that can outpace workload demands, Oracle introduced the industry's first and fastest on-chip hardware cryptographic capabilities as part of Oracle's UltraSPARC T1 processor, which was launched during 2005. Oracle continued to augment the cryptography support into each new generation of Oracle's SPARC T-Series processors.

Oracle's SPARC T5—Integrated Cryptographic Acceleration

Oracle's new SPARC T5 processor is the fifth generation of the Oracle's SPARC T-Series family. It leverages the SPARC S3 core with a redesign of the non-core processor areas. The new Oracle SPARC T5 introduces a new floating-point pipeline and further increases network bandwidth, providing approximately 3X the single-threaded throughput gains compared to its predecessor. The Oracle SPARC T5 processor includes 16 computing cores with 8 threads per core (128 threads per socket), on-chip memory management, two 10 GbE interfaces, dual on-chip based PCIe Generation 3 root complexes, on-chip cryptographic acceleration and hardware-enabled virtualization capabilities. As a result, the Oracle SPARC T5 processor eliminates the need for expensive custom hardware and software development by integrating high-performance computing, security, and significant I/O onto a single chip.

The SPARC T5 features on-chip cryptographic accelerators made available through unprivileged ISA instructions. Each of the 16 SPARC S3 cores in the Oracle SPARC T5 contains a Stream Processing Unit (SPU) that accelerates cryptographic functions at the same clock speed as the rest of the computing core. The SPU on each is implemented within the core pipelines and is accessible by 29 new user-level instructions for performing cryptographic functions. During a cryptographic operation, the cryptographic function will leverage SPU and also use parts of Floating Point/Graphics Unit (FGU) and Integer Execution Unit (EXU) pipelines with Floating-point Register Files (FRF) and Integer Register Files

(IRF). The logical depiction of SPU in Oracle SPARC T5 processor is shown in Figure 1. As a result, the SPU is designed to achieve wire-speed encryption and decryption on the processor's 10 GbE ports.

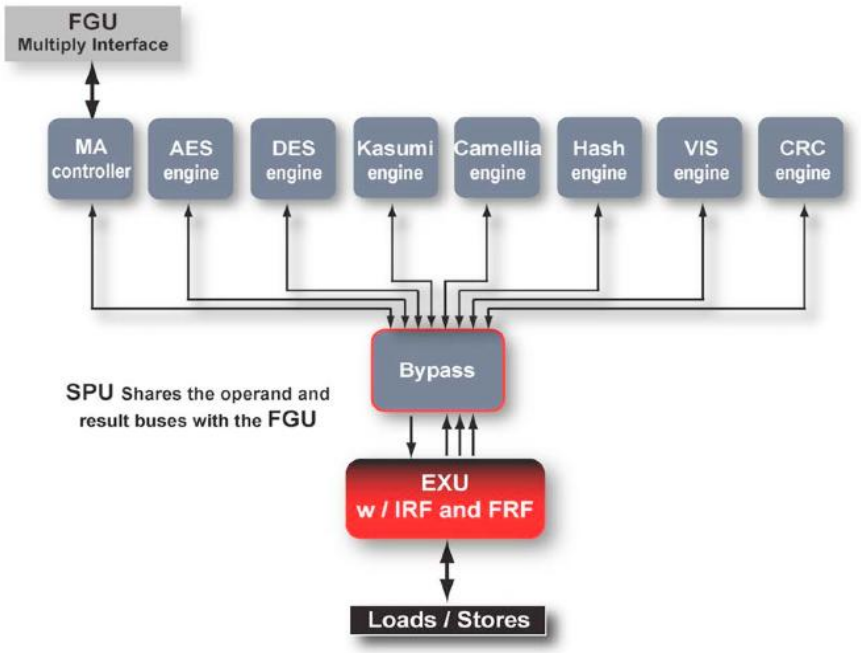

Figure 1: Oracle's SPARC T5 processor—logical depiction of stream processing unit (SPU).

The following table shows the cryptographic algorithms supported by the SPARC T5 processor

TABLE 1. SUPPORTED CRYPTOGRAPHIC ALGORITHMS

ON-CORE CRYPTOGRAPHY	SPARC T5
Accelerator drive	Userland (no drivers required)
Public key encryption	RSA, DSA, DH, ECC
Bulk encryption	AES, DES, 3DES, RC4, Kasumi, Camellia
Message digests	CRC32c, MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
APIs	PKCS#11 Standard Ucrypto APIs

When compared to alternative on-chip/on-core implementations of competition processors, the SPARC T5 offers a comprehensive set of algorithms supporting a long list of public-key encryption,

symmetric-key encryption, and message-digest algorithms. The SPARC T5 and SPARC M5-32 servers are equipped with SPARC T5 processors.

SPARC T5 Cryptographic Operational Model

With the SPARC T5 and SPARC T4 processors, the applications can directly access the on-core cryptographic functions, performing those functions in hardware—without requiring use of special configurations or drivers, kernel parameters, and administrative permissions.

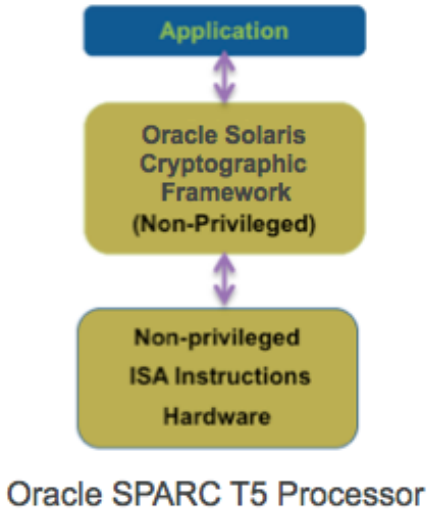

Figure 2: SPARC T5 cryptographic operational model.

The Role of Oracle Solaris Cryptographic Framework

In practice, the Oracle Solaris Cryptographic Framework acts as the core intermediary between the applications and the underlying hardware. The framework enables user-level applications to automatically leverage the hardware-assisted cryptographic acceleration functions. The Oracle Solaris Cryptographic Framework libraries provide a set of cryptographic services and application programming interfaces (APIs) whereby both kernel and user-level application consumers can transparently delegate the cryptographic operations to hardware—without adding any new code to the application.

Cryptographic Acceleration for Oracle WebLogic Server Security

The Oracle WebLogic Server applications can significantly gain security performance by offloading and delegating their cryptographic operations to the on-core cryptographic capabilities of the SPARC T5 processor.

Applied Security Mechanisms and Usage Scenarios

The Oracle WebLogic Server applications can offload select cryptographic operations for the following security scenarios.

Transport-Layer Security

- SSL/TLS acceleration offloads computationally intensive public-key encryption (e.g., RSA, ECC), bulk encryption (e.g., AES, 3DES), message digest (e.g., SHA1, SHA2), and random number generation operations.
- RMI over IIOP with SSL uses SSL/TLS to protect IIOP connections and RMI remote objects in transit.

Message-Layer Security

- Acceleration of cryptographic operations intended for supporting XML Web services security standards such as WS-Security and WS-SecurityPolicy. XML Web services security relies on public-key encryption, digital signature (e.g.. RSA, ECC), bulk encryption (e.g., AES, DES) and message digest (e.g., SHA-1, SHA-256) functions intended for supporting XML encryption, XML digital signature, and related cryptographic operations.

Secure Data at Rest

- Acceleration of cryptographic operations intended for supporting data stored in file system. This will be accomplished through the use of encryption integrated with the ZFS file system. The Oracle Solaris 11-based ZFS encryption automatically leverages SPARC T5 hardware-assisted cryptographic acceleration.

The SPARC T5 processor's on-core cryptographic acceleration capabilities can be accessed in a variety of ways by the Oracle WebLogic Server deployment, depending on the applied security scenarios and its requirements. The availability of Oracle Ucrypto provider features and SunPKCS#11 interfaces in the Java Cryptography Extension (JCE) framework enables the Oracle WebLogic Server-deployed Java EE applications and XML Web services to take advantage of hardware-assisted cryptographic acceleration of SSL and WS-Security-based cryptographic workloads.

WebLogic Security Acceleration Using Oracle Ucrypto Provider

By default, when deployed on SPARC T5 servers, the Oracle WebLogic Server relies on the Java Development Kit (JDK) and its Oracle Ucrypto provider environment for handling cryptographic operations. The Oracle Ucrypto provider in JCE contains a dedicated integration that leverages Oracle Solaris 11 Ucrypto APIs for offloading and delegating of cryptographic operations supported by Oracle's SPARC T5-based on-core cryptographic instructions. In addition, the Oracle WebLogic Server SSL must be configured to use the Java Secure Socket Extension (JSSE) provider as the default SSL provider. The JSSE provider uses the underlying JCE provider exclusively for all of its cryptographic operations and hence, WebLogic SSL configuration will be able to automatically take

advantage of hardware-assisted cryptographic acceleration capabilities through Oracle Ucrypto provider.

To leverage the Oracle Ucrypto provider, it is required to install and use JDK7 update 4 or later as the Java Runtime Environment on Oracle Solaris 11. After installation, make sure that the Oracle Ucrypto provider is identified as the default provider in the Java security properties file java.security located at $JAVA_HOME/jre/lib/security/ directory.

```
security.provider.1=com.oracle.security.ucrypto.UcryptoProvider
                        ${java.home}/lib/security/ucrypto-solaris.cfg
```

With the release of JDK7 update 4, Oracle introduced Oracle Ucrypto provider, which provides a specialized interface bypassing PKCS#11 and automatically leverages hardware-assisted cryptographic acceleration capabilities of Oracle's SPARC T4 (or newer) processors. In a typical JDK7 installation (JDK7u4 or later) on Oracle Solaris 11 (SPARC), the Java Runtime Environment is preconfigured to make use of the Oracle Ucrypto provider by default. This enables the Java and Oracle WebLogic Server-hosted applications and XML Web services to automatically delegate their cryptographic-intensive operations processed via Oracle Solaris Cryptographic Framework using Oracle's SPARC T5 on-core cryptographic instructions (Figure 3).

In addition to Oracle Ucrypto provider, the JDK provides a PKCS#11 provider implementation (SunPKCS11) that enables Java applications to access PKCS#11-based cryptographic provider implementations provided by the Oracle Solaris Cryptographic Framework.

Figure 3: Oracle WebLogic Server security using Oracle's SPARC T5 server.

Accelerating SSL Using Oracle Ucrypto Provider

The following steps explain how to configure Oracle WebLogic Server for SSL acceleration using the on-chip cryptographic acceleration capabilities of Oracle's SPARC T5 processor-based servers.

1. Configure Oracle WebLogic Server to listen for SSL. Before configuration, obtain the necessary private keys, server certificate including the public key, and trust CA certificates from a Certificate Authority (CA), and then store them into the Java keystore (identity and trust keystores) configured within the Oracle WebLogic Server environment. In case of development and testing, users may choose to use a self-signed certificate, private key, and trusted CA certificate created using Java key tool. Use the Oracle WebLogic Server Administration Console to configure the identity and trust keystores.
   Follow the SSL configuration guidelines specified in the *Oracle Fusion Middleware - Securing Oracle WebLogic Server 12c Release 1(12.1.1)* documentation.
2. Check that the Oracle WebLogic Server is listening and responds over the SSL port. This can be verified using the Oracle WebLogic Server console logs for the managed server.
3. Make sure the SSL configuration relies on the JSSE provider to enable automatic delegation of SSL-based cryptographic to automatically take advantage of Oracle Ucrypto provider. This can be accomplished by adding Java runtime option –Dweblogic.ssl.JSSEEnabled=true in the Java Runtime Environment settings of the WebLogic-managed server.
4. To enforce the hardware facilitated cryptographic capabilities, it is critical that the WebLogic SSL provider configuration must define SSL cipher suites that include cryptographic algorithms supported by the hardware. This also helps to disable weak SSL cipher suites. This can be accomplished by editing the Oracle WebLogic Server domain's config.xml:

```
<ssl>
       <enabled>true</enabled>
      <ciphersuite>TLS_RSA_WITH_AES_128_CBC_SHA</ciphersuite>
      <ciphersuite>TLS_RSA_WITH_AES_256_CBC_SHA</ciphersuite>
      . . .
</ssl>
```

5. Restart the managed server instance of Oracle WebLogic Server. This can accomplished using the Oracle WebLogic Server console.

Accelerating WS-Security Using Oracle Ucrypto Provider

WS-Security plays a critical role in providing message-level security of XML Web services by ensuring confidentiality, integrity, and access control of SOAP messages. The Oracle WebLogic Server relies on the JCE provider for supporting cryptographic operations involved with message-level security of XML Web services and the Oracle Ucrypto provider for facilitating cryptographic acceleration associated with encryption, signature, and message digest operations.

Oracle WebLogic Server strongly recommends the use of WS-SecurityPolicy standard, and it provides predefined WS-SecurityPolicy files to specify message-level security requirements. The WS-SecurityPolicy describes the message-level security requirements of the SOAP messages and how the requested operation should be digitally signed or encrypted using relevant cryptographic algorithm suites defined in the policy. The exposed Web service makes the associated security policy available via the WSDL.

The following steps explain how to configure the Oracle WebLogic Server for XML Web services security operations and its acceleration using the on-chip cryptographic acceleration capabilities of the SPARC T-Series. It is assumed that the Web service created is deployed as a JWS file that implements using Oracle WebLogic Server Web service and/or JAX-WS APIs.

Update the JWS file, including WebLogic-specific @Policy and @Policies JWS annotations to specify the predefined policy files that represent WS-Policy and WS-SecurityPolicy definitions for performing the required WS-Security mechanisms.

```
@WebService(name="Simple", targetNamespace="http://oracle.org")
@WLHttpTransport(contextPath="/wssp12/wss10",
serviceUri="UsernameTokenPlainX509SignAndEncrypt")
@Policy(uri="policy:Wssp1.2-2007-Wss1.0-UsernameToken-Plain-X509-
Basic256.xml")


public class UsernameTokenPlainX509SignAndEncrypt {

@WebMethod

@Policies({

@Policy(uri="policy:Wssp1.2-2007-SignBody.xml"),
@Policy(uri="policy:Wssp1.2-2007-EncryptBody.xml")})

public String echo(String s)

{

   return s;

}
```

The above WS-Policy identifies the WS-SecurityPolicy specifying the service authenticates the client using a username token, and both the request and response messages are signed and encrypted with X.509 certificates.

After including the policies, recompile and deploy the Web service. For Web services configuration and deployment steps, refer to the *Oracle Fusion Middleware documentation for Securing WebLogic Web Services for Oracle WebLogic Server 12c*.

It is also critical the client application that invokes a deployed Web service must be associated to a client-side security policy file. Typically, the security policy files are the same as those configured for the server-side Web service invoked, but because the server-side files are not exposed to the client Java runtime, the client application must load its own local copies. Users may also choose to use the weblogic.jws.jaxws.ClientPolicyFeature class in the client application to override the effective policy defined for a service.

Make sure the cryptographic mechanisms specified in the security policy file identify the WS-SecurityPolicy algorithm suite supported by the Oracle Ucrypto provider or the SunPKCS11 provider. Refer to the *Java PKCS#11 Reference Guide* for the supported list of cryptographic algorithms. If the specified algorithm suite is Basic256, it represents AES-256 algorithm for bulk encryption, Sha256 algorithm to represent SHA-256 based message digests and Rsa-oaep-mgf1p to represent RSA for key wrap. The Oracle Ucrypto provider supports both AES-256 and SHA256withRSA algorithms, and it leverages the use of Oracle's SPARC T5 hardware-assisted cryptographic acceleration.

Update the Java client application to make sure it loads the client-side policy files and rebuild/redeploy the client application. If the client is a Web application, restart the WebLogic managed server instances where it is deployed.

Securing Data at Rest Using ZFS Encryption

Oracle WebLogic Server applications are tested and verified to install and run on an encrypted file system provided by Oracle Solaris 11 ZFS encryption. By default, ZFS uses the Oracle Solaris 11 cryptographic services APIs, which automatically benefit from the hardware acceleration of the AES algorithm available on the SPARC T5 processors. The policy for encryption is set at the dataset level when datasets (file systems or ZVOLs) are created. Each ZFS on disk block (smallest size is 512 bytes, largest is 128 k) is encrypted using AES algorithm in either CCM or GCM mode. The wrapping keys need to be provided by the Oracle Solaris administrator who creates the file system, which can be changed at any time without taking the file system offline. The data encryption keys are randomly generated at dataset creation time. The easiest way to create the wrapping keys is to use the existing Oracle Solaris pktool command:

```
$ pktool genkey

       keystore=file keytype=aes keylen=128

                          outkey=/export/home/user/mykey
```

```
Using ZFS encryption support can be as easy as this:
```

```
# zfs create -o encryption=on -o

          keysource=raw,file:///export/home/user/mykey
```

```
                                                    myfilesystem/cryptofs
```

Alternatively, for ensuring secure storage and retrieval of wrapping keys, it is recommended to use the Oracle Solaris PKCS#11 softtoken as keystore for storing wrapping keys. Using Oracle Solaris PKCS#11 softtoken as keystore ensures that the wrapping key is encrypted in storage and the keystore is protected by a PIN. The steps involved with creating and storing the wrapping key in an Oracle Solaris PKCS#11 softtoken keystore and using the key to create an encrypted ZFS data set is as follows:

```
# pktool genkey
       keystore=pkcs11 keytype=aes keylen=128 label=mykey

Enter PIN for Sun Software PKCS#11 softtoken:

# zfs create -o encryption=on
      -o keysource=raw,pkcs11:object=mykey myfilesystem/cryptofs

Enter PKCS#11 token PIN for ' myfilesystem/cryptofs'
```

In the example above, an AES key is created in the default softtoken keystore for the user. This keystore requires authentication to create and use keys stored in it, so a user is prompted for the keystore PIN (it is really a passphrase, but PKCS#11 terminology uses the word PIN for legacy reasons). The syntax of the PKCS#11 URI that is used with the keysource property allows for specifying a path to the PIN file. Using this method ensures that the actual wrapping key is encrypted and protected in the PKCS#11 keystore.

For more details on ZFS encryption, refer to the *Oracle Solaris ZFS Administration Guide*.

Examining Hardware-Assisted Cryptographic Operations

After deployment of WebLogic SSL or WS-Security, it is important to examine and confirm whether the configured on-chip hardware accelerator is acting on those delegated operations or any other software providers performing them. To ensure the SPARC T5 hardware-assisted cryptographic acceleration is configured to use and working with the security scenarios, it is recommended to use the following Oracle Solaris DTrace feature script.

```
#!/usr/sbin/dtrace -s
```

```
pid$1:libsoftcrypto:yf*:entry,
pid$1:libmd:yf*:entry
{
  @[probefunc] = count();
}
tick-10sec
{
  printa(@);
  clear(@);
  trunc(@,0);
}
tick-100sec
{exit(0);}
```

Save the above script as cryptoverify.d file and run the Dtrace script including the "Weblogic server's Java process id" as command line argument.

```
# dtrace -s cryptoverify.d  <WeblogicServer Process ID>
```

For example, in an XML encryption scenario using AES-256 algorithm, a positive and growing value of aes jobs indicates that cryptographic acceleration is operational on the target AES bulk encryption payloads—refer to the following sample output.

```
# dtrace -s cryptoverify.d 5774
dtrace: script 'cryptoverify.d' matched 51 probes
CPU     ID                    FUNCTION:NAME
 65  83719                      :tick-10sec
  yf_aes256_ecb_decrypt                                    39922
  yf_aes256_load_keys_for_decrypt                          39922

 65  83719                      :tick-10sec
  yf_aes256_ecb_decrypt                                    44108
  yf_aes256_load_keys_for_decrypt                          44108

 65  83719                      :tick-10sec
  yf_aes256_ecb_decrypt                                    44534
  yf_aes256_load_keys_for_decrypt                          44534

..
```

Performance Characteristics

Hardware and Software Environments

The Oracle WebLogic Server 12*c* server environment and its multitier application security scenarios using SPARC T5 hardware-assisted cryptographic acceleration have been tested and verified to run on the following Oracle hardware and software environments (Table 2):

TABLE 2: HARDWARE AND SOFTWARE ENVIRONMENT TESTED

OPERATING SYSTEM	HARDWARE ENVIRONMENT	ORACLE WEBLOGIC SERVER ENVIRONMENT
Oracle Solaris 11.1 GA	SPARC T5-2 server	Oracle WebLogic 12*c* (12.1.1) (1 admin server, 4 managed servers)

SSL Performance

The following graph (Figure 4) represents the SSL operations performance characteristics of the following WebLogic SSL-based application security scenarios.

- WebLogic managed servers deployed with a Java EE / JAX-WS Web services application.
- WebLogic managed servers deployed with a Java EE / JAX-WS Web services application configured to use SSL with two-way SSL using JKS keystores (JDK configured using a third-party JCE provider with no cryptographic acceleration support).
- WebLogic managed servers deployed with a Java EE / JAX-WS Web services application and configured with two-way SSL using JKS keystores and JDK configured to use the Oracle Ucrypto provider for enabling Oracle SPARC T5 cryptographic acceleration.
- The WebLogic server environment, including applications and database, is installed and configured on encrypted ZFS data sets-based file system. WebLogic managed servers deployed with a Java EE / JAX-WS Web services application and configured with two-way SSL using JKS keystores and JDK configured to use the Oracle Ucrypto provider for enabling Oracle's SPARC T5 cryptographic acceleration.

Each of these scenarios uses a Java EE/JAX-WS Web services application using a 500 k XML payload deployed on Oracle WebLogic Server 12*c* managed server instances running on Oracle's SPARC T5-2 server. The tests are simulated to run with 300 concurrent users for 60 minutes. The SSL and WS-Security configuration uses the following cryptographic material:

- SSL certificate
  - RSA-2048 (Key algorithm)
  - SHA256withRSA (Signature algorithm)
- Oracle WebLogic Server-enforced SSL cipher suite
  - TLS_RSA_WITH_AES_256_CBC_SHA

- WS-Security
  - AES-256
- ZFS encryption
  - AES-128

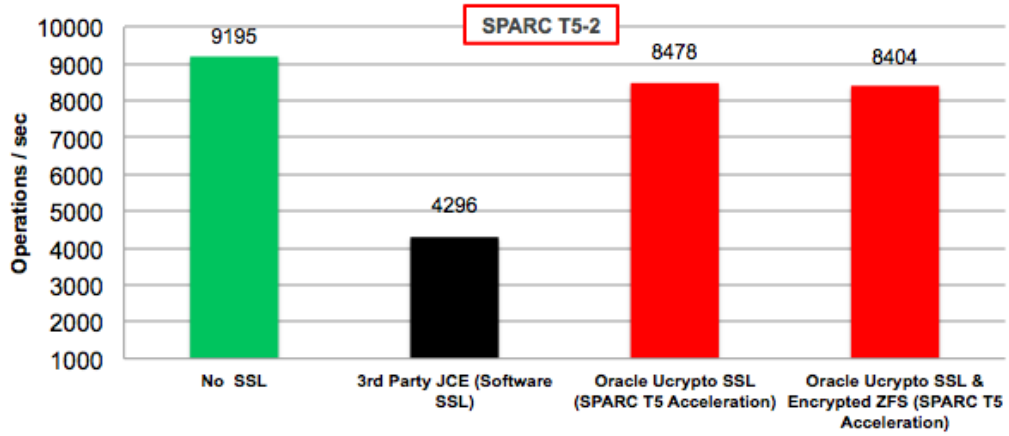

Figure 4: Performance characteristics—no SSL versus software SSL versus SPARC T5 hardware-accelerated SSL.

As a result, enabling hardware-assisted cryptographic acceleration with Oracle Ucrypto provider for WebLogic two-way SSL and WS-Security scenarios on encrypted ZFS file system solidly delivered between 200 percent to 300 percent overall application performance gain in comparison with Weblogic SSL and WS-Security running a third-party JCE provider with no cryptographic acceleration support. It is also observed that the hardware-accelerated WebLogic SSL (using SSL cipher with RSA-2048, AES-256, SHA256withRSA) on encrypted ZFS file system showed a negligible overhead of about ~7 percent while comparing to Oracle WebLogic Server running with no SSL.

System Performance

The following graph (Figure 5) represents the overall CPU utilization characteristics of using Oracle's SPARC T5 cryptographic acceleration and the effect of cryptographic overheads on the CPU while comparing with using no SSL.

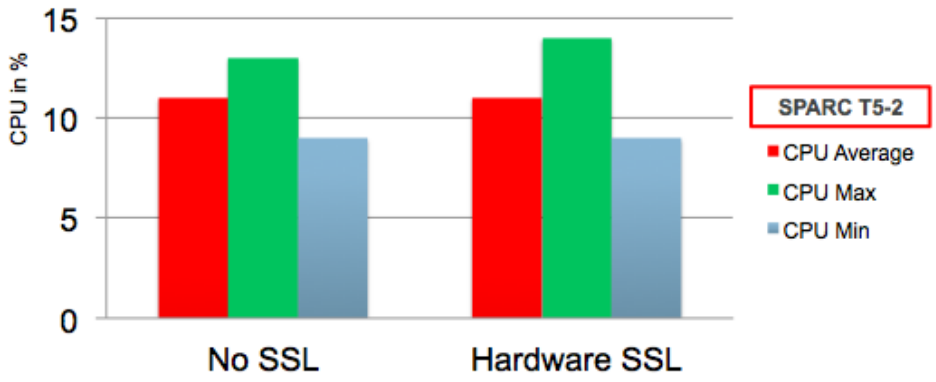

Figure 5: CPU utilization—no SSL versus hardware-assisted SSL.

The results show negligible and near-zero overheads in CPU utilization due to WebLogic SSL, WS-Security, and encrypted ZFS with SPARC T5 hardware-assisted cryptographic acceleration.

These performance characteristics yield tangible, immediate, and cost-efficient results in the form of faster and more secure transactions as well as better response times. Additionally, the results clarify the massive burden unaccelerated cryptographic workloads can have on a server.

Conclusion

This white paper presents Oracle's SPARC T5 processor's cryptographic acceleration features available with SPARC T5 and SPARC M5 servers and their support for securing Oracle WebLogic Server applications and XML Web services. This paper unveils the core mechanisms, configuration, deployment strategies, and the role and relevance of using the Java Cryptographic Extensions-based techniques, and ZFS encryption. SSL has become the defacto industry standard for delivering transport-layer security in Web applications and XML Web services. The SSL and WS-Security performance characteristics presented in this white paper clearly show the compelling security performance and consistent scalability benefits of adopting Oracle's SPARC T5-based hardware-assisted cryptographic acceleration mechanisms for securing Web applications and XML Web services scenarios.

Further References

Oracle's SPARC T5 and M5 servers

http://www.oracle.com/us/products/servers-storage/servers/sparc-enterprise/t-series/index.html

Oracle Fusion Middleware Securing Oracle WebLogic Server 12*c* Release 1 (12.1.1) documentation

http://docs.oracle.com/cd/E24329_01/web.1211/e24422/toc.htm

Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server 12*c* Release 1 (12.1.1) documentation

http://docs.oracle.com/cd/E24329_01/web.1211/e24488/overview.htm

Java PKCS#11 Reference Guide

http://download.oracle.com/javase/7/docs/technotes/guides/security/p11guide.html

Developer's Guide to Oracle Solaris 11 Security

http://docs.oracle.com/cd/E23824_01/html/819-2145/index.html

Java Secure Socket Extension (JSSE) Reference Guide

http://docs.oracle.com/javase/7/docs/technotes/guides/security/jsse/JSSERefGuide.html

High-Performance Security for Oracle WebLogic Applications Using Oracle's SPARC T5 and SPARC M5 Servers
May 2013
Authors: Ramesh Nagappan
Glenn Brunette

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together