Performante Verschlüsselung mit Oracle Hardware und Solaris

Stefan Hinker
Oracle Deutschland B.V. & Co KG
Ratingen

Schlüsselworte:
Verschlüsselung, Hardwarebeschleunigung, SPARC, Solaris, Anwendungen

Einleitung

Verschlüsselung von Daten im Netzwerk und auf Speichersystemen gehört zwingend in jedes Sicherheitskonzept. Verschlüsselung ist eine teure Operation, die viel CPU-Resourcen beansprucht. Unterstützende Hardware, entweder in Form von Zusatzkarten oder in Form erweiterter CPU-Funktionen beispielsweise in den UltraSPARC T CPUs oder der INTEL Xeon 5600 Serie schafft hier Abhilfe.

Dieser Vortrag gibt einen Überblick über verschiedene Varianten der Hardwarebeschleunigung im Produktportfolio von Oracle. An Beispielen zu Oracle Transparent Data Encryption und SSL-Beschleunigung wird gezeigt, wie das Solaris Cryptographic Framework diese transparent an verschiedene Anwendungen vermittelt.

Motivation

Verschlüsselung ist eine unverzichtbare Technik in der heutigen Datenverarbeitung. Die Kosten eines Datenlecks durch unberechtigten Zugriff können enorm sein. Im Jahr 2008 entstand deutschen Firmen im Durchschnitt Kosten in Höhe von 2,4 Millionen € pro Datenleck . Oft noch schlimmer sind die nur schwer zu beziffernden Schäden für Ansehen und Vertrauen. Daher ist es nicht verwunderlich, dass Verschlüsselung als Schutztechnik immer mehr Verbreitung findet und in vielen Fällen inzwischen auch gesetzlich vorgeschrieben wird.

Die dabei verwendeten Techniken wurden im Laufe der Zeit stetig verbessert. Eine Revolution erlebte die Verschlüsselungstechnik während des zweiten Weltkriegs, als Mathematiker der Alliierten mit Hilfe der ersten Computer den Code der „Enigma“ durchbrachen. Heute wie damals reicht es nicht mehr aus, nur den Algorithmus zu kennen, um einen Code zu brechen. Vielmehr muss der passende Schlüssel gefunden werden, ohne den auch die Kenntnis des Algorithmus wertlos ist. Entsprechend der stetigen Leistungssteigerungen in der Computertechnik wurden die Lösungsräume, in denen der Schlüssel letztlich durch Raten gefunden werden musste, immer größer. Hieraus ergab sich auch zwingend, dass das ver- und entschlüsseln von Daten auch bei Kenntnis des Schlüssels mit signifikantem Rechenaufwand verbunden ist und auch bleiben wird.

Wo Verschlüsseln?

Verschlüsselt wird heute an fast jeder Stelle im Datenfluss:

- Daten in Ruhe: Jede Festplatte und jedes Band verlassen das Rechenzentrum irgendwann. Sei es durch Reparatur, Erneuerung, Umzug oder auch Diebstahl. Sind die Daten auf der Festplatte verschlüsselt, sind sie auch dann noch sicher.

- Daten Unterwegs: Daten im Netzwerk sind häufig verschlüsselt, um ein Ausspähen zu verhindern.
- Daten bei der Verarbeitung: Nicht immer braucht man bei der Verarbeitung die Gesamtheit aller Daten. Die nicht benötigten Datenteile können verschlüsselt bleiben.

Gleichzeitig werden Daten auf allen Ebenen verschlüsselt: In der Anwendung, in der Middleware, von der Datenbank, dem Betriebssystem und eventuell der Hardware, bspw. einem Festplatten-Controller.

Verschlüsseln ist Aufwendig

Heutige Verschlüsselungsalgorithmen beruhen auf komplexen mathematischen Formeln, die aus einem Schlüssel und den Klartextdaten den Schlüsseltext berechnen, oder umgekehrt. Diese Berechnung ist aufwendig, belastet also die CPUs der Rechensysteme zusätzlich zu deren eigentlichen Aufgaben. Gerade auch dort, wo große Mengen an kleineren Datenpaketen ver- und entschlüsselt werden müssen entsteht durch diese Operationen auch eine zusätzliche Zeitverzögerung in der Verarbeitung, die nicht immer akzeptabel ist. Schon früh wurde deshalb nach Wegen gesucht, die Krypto-Operationen auf spezialisierte Hardware auszulagern, um damit einerseits eine Entlastung der CPU zu erreichen und gleichzeitig die einzelne Operation zu beschleunigen. Heute gibt es im wesentlichen Zusatzgeräte zur SSL-Terminierung, Erweiterungskarten mit Kryptofunktionen, verschlüsselnde Festplatten- und Bandlaufwerke sowie CPUs mit entsprechenden Funktionserweiterungen.

Weitere Funktionen

Neben dem Ver- und Entschlüsseln der eigentlichen Daten gibt es noch die Schlüssel selbst, denen in diesem Zusammenhang besondere Aufmerksamkeit gewidmet werden muss. Je nach Art der zu schützenden Daten muss der Schlüssel für wenige Sekunden, Minuten oder aber auch für mehrere Jahre vor unbefugtem Zugriff geschützt werden. Da Schlüssel meist nur wenige Bytes lang sind, ist dies wesentlich einfacher als bei der gesamten Datenmenge. Auch hier finden häufig wieder kryptographische Methoden Anwendung. Typischerweise werden mehrere Schlüssel in einem speziell gesicherten Bereich gespeichert, eventuell durch einen Master-Schlüssel nochmals verschlüsselt und so gesichert. Auch für die Speicherung und Verwaltung der Schlüssel gibt es heute geeignete Systeme.

Hardwareverschlüsselung bei Oracle

Auch Oracle Software verwendet Verschlüsselung an den verschiedensten Stellen, nicht zuletzt natürlich in der Datenbank, den Web- und Application Servern und in den Betriebssystemen Oracle Enterprise Linux und Oracle Solaris. Auf der Hardwareseite gibt es ebenfalls einige Produkte, auf die im weiteren näher eingegangen werden soll.

Oracle Sun Crypto Accelerator 6000

Diese PCIe Karte bietet einerseits Hardwareunterstützung für alle gängigen Algorithmen an und kann hier mit robusten Leistungszahlen aufwarten. Andererseits enthält die Karte einen nach FIPS 140-2 Level 3 zertifizierten Speicher, der als Tokenstore zur Speicherung von Schlüsseln und sonstigen Krypto-Objekten geeignet ist. Damit ist die Karte nicht nur ein Beschleuniger, sondern auch ein Hardware Security Modul (HSM). Mehrere dieser Karten können zur Lastverteilung und für bessere Verfügbarkeit in einem Rechner verwendet werden, aber auch in einem Verbund zusammen arbeiten. Die zum Betrieb notwendige Software ist unter Linux und unter Solaris (SPARC & x86) verfügbar.

Kryptobeschleunigung durch CPU-Erweiterungen

Die UltraSPARC T1, T2 und T2+ CPUs enthalten alle eine sogenannte „Modular Arithmetic Unit“, die die bei public-key Verfahren wie RSA oder DSA notwendige Modulo-Arithmetik stark beschleunigt.

Die CPUs der T2/T2+ Reihe enthalten zusätzlich eine „Stream Processing Unit“, die für die Beschleunigung der eigentlichen Verschlüsselungsalgorithmen wie bspw. AES oder DES zuständig ist. Mit diesen Einheiten sind diese CPUs in ihrer kryptographischen Leistung unerreicht.

Mit der Serie Xeon 5600, auch bekannt als Westmere, bringt Intel zum ersten mal ebenfalls eine CPU auf den Markt, die Funktionen zur Kryptobeschleunigung enthält. Allerdings ist diese auf Algorithmen für AES beschränkt und daher weniger universell einsetzbar als UltraSPARC T. Oracle bietet diese CPUs in Servern der X-Serie an.

Allen diesen Hardwarebeschleunigern gemeinsam ist die Software, mittels der sie Anwendungen zur Verfügung stehen: Das Solaris Cryptographic Framework.

Solaris Cryptographic Framework

Das Solaris Cryptographic Framework stellt sowohl dem Solaris Kernel als auch Benutzerprozessen kryptographische Funktionen zur Verfügung. Sogenannte „Provider“ können als Module eingebunden werden, um dem Framework einzelne Funktionen in Hardware oder Software zur Verfügung zu stellen. Der Kernel und die Benutzerprozesse verwenden die Schnittstellen des Frameworks, um diese Funktionen in Anspruch zu nehmen. Dabei hat der Kernel ein eigenes, geschlossenes API, während Benutzerprozesse typischerweise über die standardisierte PKCS#11 API auf das Framework zugreifen. Mittels des Kommandos „cryptoadm“ kann konfiguriert werden, welche Funktionen der einzelnen Provider jeweils verwendet werden.

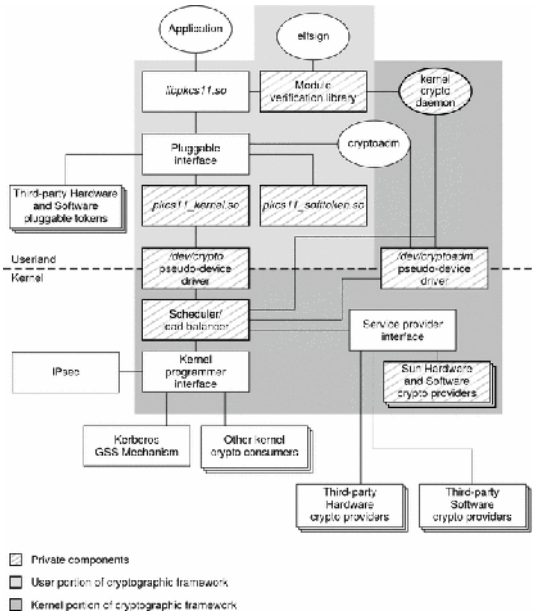

Abbildung 1: Das Solaris Cryptographic Framework

Das Framework überwacht durch den „kernel crypto daemon“ selbstständig die Integrität der verwendeten Module. Dieser Daemon stellt sicher, dass nur von Oracle signierte Softwaremodule in das Framework integriert werden. Module von Drittherstellern können mit einer entsprechenden Signatur durch Oracle natürlich ebenfalls verwendet werden.

Sowohl die Funktionen der SCA 6000 PCIe Karte als auch die der UltraSPARC T und Xeon 5600 CPUs werden als Hardware Crypto Provider in das Framework eingebunden. Ihre Verwendung ist dann durch alle Anwendungen möglich, die per PKCS#11 auf Kryptodienste zugreifen können. Im Falle der SCA 6000 gilt dies nicht nur für die Beschleunig von Verschlüsselung, sondern auch für die Funktion als HSM.

Leistungssteigerung durch Hardwarebeschleunigung

Die erreichbare Leistungssteigerung durch Hardwarebeschleunigung ist beachtlich. Die SPECweb2005 Weltrekorde der UltraSPARC T2 CPU (in einem Sockel für Sockel Vergleich) bestehen bis heute. Hardwarebeschleunigte Kryptographie spielt bei diesen Ergebnissen eine grosse Rolle, da ein signifikanter Teil des Benchmarks auf SSL beruht.

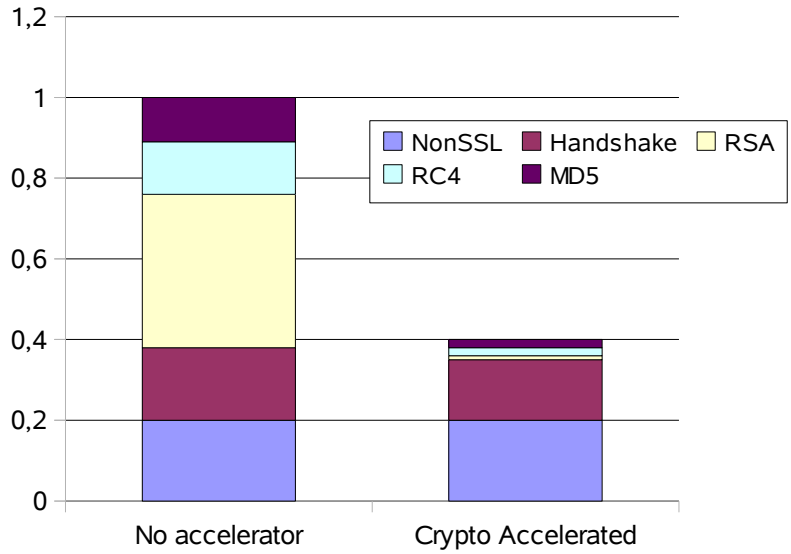

Abbildung 2: SPECweb2005: Gewinn durch Hardwarebeschleunigung

Aus dem Jahr 2007 stammt dieser Vergleich der reinen Krypto-Leistung dieser CPUs:

Cipher	2.2GHz dual-core Opteron	2.67GHz quad-core Clovertown	1.9GHz dual-core Power5	1.4GHz UltraSPARC T2
RSA1024	2.3K Ops/sec	4.8 K Ops/sec	0.5K Ops/sec	37.0K Ops/sec
AES-128	1.6 Gb/sec	4.2 Gb/s	1.3 Gb/sec	40.0 Gb/sec

Auch einen Vergleich mit PCIe Steckkarten muss diese CPU nicht fürchten:

Cipher	Sun SCA6000	Cavium Nitrox PX	1.4GHz UltraSPARC T2
RSA1024	13K Ops/sec	12K Ops/sec	37K Ops/sec
AES-128	1.0Gb/sec	2.5Gb/sec	40Gb/sec

Diese Werte wurden jeweils mit einem Oracle-Internen, systemnahen Test ermittelt. Sie sind daher nicht repräsentativ oder übertragbar.

Anwendungsbeispiele

Apache als Secure Webserver

Der Apache Webserver ist der wohl bekannteste Webserver mit hoher Verbreitung. Zum Betrieb als sicherer Webserver mit SSL verwendet Apache OpenSSL. OpenSSL seinerseits kann verschiedene „en-

gines“ als Cryptoprovider verwenden, u.A. PKCS#11. In der mit Solaris ausgelieferten Version von OpenSSL und Apache 2 sind die notwendigen Libraries zur Verwendung von PKCS#11 bereits enthalten. Um sie zu aktivieren und damit eine evtl. vorhandene Hardwarebeschleunigung zu verwenden ist es lediglich erforderlich, in der Konfigurationsdatei (typischerweise httpd.conf oder ssl.conf) eine Zeile mit „`SSLCryptoDevice pkcs11` “ aufzunehmen. Natürlich ist es auch weiterhin notwendig, die entsprechenden Zertifikate zu erzeugen, die entsprechende Konfiguration unterscheidet sich hier nicht weiter.

SSL per KSSL Proxy

Das Solaris Cryptographic Framework enthält u.A. einen Kernel SSL Proxy, mit dem fast beliebige TCP-Dienste um eine sichere Verbindung erweitert werden können. Dabei terminiert der KSSL-Proxy die SSL-Verbindung zum Client und leitet die Nachrichteninhalte im Klartext an den eigentlichen Server weiter. Der Vorteil des KSSL Proxy ist nicht nur, dass damit auch nicht-SSL fähige Dienste sicher imNetzwerk kommunizieren können, sondern auch, dass durch die Abwicklung der SSL-Terminierung im Kernel diese sehr effizient verarbeitet werden kann, was zu einem Geschwindigkeitsvorteil von ca. 20% gegenüber herkömmlichen SSL-Lösungen führt. Sind Hardwarebeschleuniger vorhanden, werden diese natürlich ebenfalls genutzt.

Um den KSSL Proxy bspw. für einen generischen Webserver zu konfigurieren, müssen lediglich die notwendigen Zertifikate vorhanden sein, z.B. in einer pk12-Datei. Der KSSL-Proxy wird dann mit dem Kommando „ksslcfg create -f pkcs12 -i schluesseldatei.p12 -x 8080 www.meinserver.de 443“ konfiguriert und gestartet. In diesem Beispiel würde der Webserver „www.meinserver.de“ auf Port 8080 arbeiten. Die Klienten dagegen können sich mit Port 443 per HTTPS verbinden. Ein Anwendungsbeispiel hierfür ist der Weblogic Application Server, der auf diese Weise Hardwarebeschleunigung nutzen kann.

SSH

Seit Solaris 10 5/09 unterstützt die mitgelieferte Version von SSH die PKCS#11 Engine von OpenSSL, und damit automatisch eventuell verfügbare Hardwarebeschleuniger. Dies ist insbesondere für „scp“ interessant, da hier die Übertragungsgeschwindigkeit deutlich erhöht werden kann. Da für die Übertragung Block-Chiffren wie bspw. AES oder DES verwendet werden, kommen hierfür UltraSPARC T2/T2+, Intel 5600 oder die SCA 6000 in Frage. Möchte man dies aus irgendwelchen Gründen nicht verwenden, kann man den Parameter „UseOpenSSLEngine“ aus der Konfigurationsdatei entfernen.

JAVA

Alle neueren Versionen der Java Laufzeitumgebung (JRE) verwenden automatisch das Solaris Cryptographic Framework, um Kryptofunktionen anzufordern. Damit steht diese Funktionalität jedem entsprechenden Java-Programm transparent und automatisch zur Verfügung. Konfiguriert wird dieser Zugriff in der Datei „$JAVA_HOME/jre/lib/security/java.security “, weitere Details, wie bspw. einzelne Algorithmen, können in der Datei „$JAVA_HOME/jre/lib/security/sunpkcs11-solaris.cfg“ eingestellt werden. Auf diese Weise profitiert z.B. der Oracle Weblogic Application Server von der Hardwarebeschleunigung. Dies ist im Blueprint „Taking Advantage of Wire-Speed Cryptography“ beschrieben.

Oracle Transparent Data Encryption und das Solaris Cryptographic Framework

Die Oracle Datenbank unterstützt seit der Version 10.2 die Verschlüsselung von einzelnen Spalten einer Tabelle. Seit der Version 11.1.7 bzw. 11.2 wird die Verschlüsselung von Tablespaces angeboten. In beiden Fällen wird der hierfür verwendete Schlüssel idR. im Oracle Wallet gespeichert. Sehen die Sicherheitsanforderungen zur Speicherung dieses Schlüssels ein HSM vor, kann u.A. die SCA 6000

zum Einsatz kommen. In diesem Fall verwendet Oracle die Solaris PKCS#11 Library, um den Schlüssel als Krypto-Token abzulegen. Der Zugriff auf den Schlüssel kann dann zusätzlich zu den normalen Passworten, die beim öffnen des in der SCA 6000 gespeicherten Wallets abgefragt werden, durch die Administratoren des Tokenstores der Karte kontrolliert werden. Dies ermöglicht eine weitere Sicherheitsstufe durch Trennung von DBA und Wallet-Admin, ggf. einschließlich der Verwendung des 4-Augen Prinzips.

Zu Test- oder Demo-Zwecken kann das gleiche Verfahren auch ohne eine SCA 6000 angewandt werden. Als Ersatz für den Tokenstore der Karte kommt der im Solaris Cryptographic Framework enthaltene Softtokenstore zum Einsatz. Dieser wird ebenfalls per PKCS#11 angesprochen, ist jedoch ähnlich wie das Oracle Wallet lediglich eine entsprechend geschützte Datei im Dateisystem des Oracle-Benutzers.

Kontaktadresse:

Stefan Hinker
Oracle Deutschland B.V. & Co KG
Brandenburger Str. 2, 40880 Ratingen

Telefon:	+49 6103 752 300
E-Mail	Stefan.Hinker@oracle.com
Internet:	www.oracle.com
	blogs.sun.com/cmt