Maximizing Security with LinuxONE

Lydia Parziale
Yongkook Kim
Rushir Patel
Narjisse Zaki
Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (October 2019)

This edition applies to LinuxONE, 2019.

This document was created or updated on October 16, 2019.
### Contents

<table>
<thead>
<tr>
<th>Notices</th>
<th>.v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trademarks</td>
<td>vi</td>
</tr>
<tr>
<td>Preface</td>
<td>vii</td>
</tr>
<tr>
<td>Authors</td>
<td>vii</td>
</tr>
<tr>
<td>Now you can become a published author, too!</td>
<td>vii</td>
</tr>
<tr>
<td>Comments welcome</td>
<td>vii</td>
</tr>
<tr>
<td>Stay connected to IBM Redbooks</td>
<td></td>
</tr>
</tbody>
</table>

**Chapter 1. Introduction**

1.1 Introduction to LinuxONE .......................... 1
1.2 Enterprise Security Challenges ...................... 2
   1.2.1 Data protection and privacy .................... 2
   1.2.2 Secure hybrid cloud integration ................. 3
   1.2.3 Cyber resiliency and availability ............... 4
   1.2.4 Industry and regulatory compliance .............. 4
1.3 IBM LinuxONE servers .................................. 5
   1.3.1 IBM LinuxONE III ................................ 6
   1.3.2 IBM LinuxONE Emperor II .......................... 7
   1.3.3 IBM LinuxONE Rockhopper II ....................... 9

**Chapter 2. Core Security Technologies on LinuxONE**

2.1 Secure cryptographic hardware ......................... 13
   2.1.1 Central Processor Assist for Cryptographic Functions 15
   2.1.2 IBM Crypto Express adapter ....................... 16
2.2 Virtualization technology ............................. 21
   2.2.1 PR/SM and LPARs ................................ 22
   2.2.2 KVM ........................................... 22
   2.2.3 z/VM .......................................... 23

**Chapter 3. Exploits of Security on LinuxONE**

3.1 IBM Secure Service Container ......................... 25
3.2 IBM Secure Boot for Linux .............................. 26
3.3 IBM Cloud Hyper Protect Services ....................... 26
   3.3.1 IBM Cloud Hyper Protect Crypto Services .......... 27
   3.3.2 IBM Cloud Hyper Protect DBaaS ................... 27
3.4 Cryptographic Key Management for LinuxONE .......... 27
   3.4.1 Operational Key Lifecycle Management ............ 28
   3.4.2 Master Key Lifecycle Management .................. 29

**Chapter 4. Use cases**

4.1 Containers and data encryption use case ............... 33
   4.1.1 Context and challenges .......................... 34
   4.1.2 Solution ....................................... 34
   4.1.3 Implementation .................................. 35
   4.1.4 Summary ....................................... 40
4.2 Database and volume encryption use case ............... 41
   4.2.1 Context and challenges .......................... 41
   4.2.2 Solution ....................................... 42
Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.
Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

<table>
<thead>
<tr>
<th>Db2®</th>
<th>IBM Cloud™</th>
<th>Redbooks®</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS8000®</td>
<td>IBM Cloud Pak™</td>
<td>Redbooks (logo)®</td>
</tr>
<tr>
<td>FICON®</td>
<td>IBM Z®</td>
<td>z/VM®</td>
</tr>
<tr>
<td>IBM®</td>
<td>RACF®</td>
<td></td>
</tr>
</tbody>
</table>

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.
Preface

LinuxONE is a hardware system that is designed to support and exploit the Linux operating system based on the value of its unique underlying architecture. LinuxONE can be used within a private and multi-cloud environment to support a range of workloads and service a variety of needs.

On LinuxONE, security is built into both the hardware and software.

This IBM® Redpaper publication gives a broad understanding of how to leverage the various security features that exploit and complement the LinuxONE hardware security features, such as these:

- Hardware accelerated encryption of data, which is delivered with near-zero overhead by the on-chip Central Processor Assist for Cryptographic Function (CPACF) and a dedicated Crypto Express adapter.
- Virtualization and industry-leading isolation capabilities with PR/SM, EAL 5+ LPARs, DPM, KVM, and IBM z/VM®.
- The IBM Secure Service Container technology, which provides workload isolation, restricted administrator access, and tamper protection against internal threats, including from systems administrators.
- Other technologies that exploit LinuxONE security capabilities and deeper dives into practical use cases for these technologies.

This IBM Redpaper publication was written for IT executives, architects, specialists, security administrators, and others who consider security for LinuxONE.

Authors

This paper was produced by a team of specialists from around the world working at the IBM Redbooks Center, Poughkeepsie.

Lydia Parziale is a Project Leader for the IBM Redbooks® team in Poughkeepsie, New York, with domestic and international experience in technology management including software development, project leadership, and strategic planning. Her areas of expertise include business development and database management technologies. Lydia is a certified PMP and an IBM Certified IT Specialist with an MBA in Technology Management and has been employed by IBM for over 25 years in various technology areas.

Yongkook Kim is an engineer/architect with over 20 years of industry experience. Yongkook started his career as a design engineer for IBM Z® Crypto HW at IBM Poughkeepsie, New York in 2001, and designed AES engine in ASICs for Crypto Express 2. He then joined IBM sales team for Wall Street to assist financial sector clients with IBM Z technology. He has been a technical advisor for IBM Z clients in NY Metro area since 2007 and participated in multiple benchmarks with international teams as well. Yongkook joined Vicom Infinity, a premiere IBM Business Partner in 2014 as a Solutions Architect. He actively participated in PoCs and solution designs with various industry clients and enjoys adopting new technologies such as blockchain, voice assistants, and IoT into enterprise solutions like LinuxONE. Yongkook holds Masters degree from NYU Polytech School of Engineering.
Rushir Patel is an IBM Offering Manager for LinuxONE based in Raleigh, North Carolina, USA. He has 8+ years of experience in technology product management, agile software development, and UX research and design. His areas of expertise include cybersecurity, enterprise data management, and cloud computing technology. He has an MBA from the University of North Carolina Kenan-Flagler Business School and a BSc Electrical Engineering from North Carolina State University.

Narjisse Zaki is an IT Architect in IBM ATS. She provides pre-sales technical support for Linux consolidation projects on Linux on Z and LinuxONE across EMEA countries. She is based in Montpellier, France. Narjisse is also the engagement leader for LinuxONE pre-sales activities in MEA at the IBM Systems Center, Montpellier. In past years, Narjisse supported Oracle consolidation projects within the European IBM Oracle Center (IOC). Before she joined the IOC, Narjisse was involved in Agile Project Management leading projects as a Scrum Master. She has been an international speaker in several conferences and events around the world. She holds a M. Engineering in Computer Science and Management. One of her favorite quotes is, “We travel, some of us forever, to seek other places, other lives, other souls.”

Thanks to the following people for their contributions to this project:

Robert Haimowitz
IBM Redbooks, Poughkeepsie Center

Edi Lopes Alves
IBM Brazil

Guillaume Hoareau, Sylvain Carta
IBM France

Dr. Reinhard Bündgen, Stefan Raspl, Pradeep Parameshwaran, Tony Gargya
IBM Germany

Alixander Laffredo-Dietrich, Jin VanStee, Rebecca Gott, Brett Webb, Anthony Sofia, Todd Arnold
IBM USA

Tom Amodio
President, Vicom Infinity, Inc.

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience with leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies last from 2 to 6 weeks, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html
Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

► Use the online Contact us review Redbooks form found at:
  ibm.com/redbooks

► Send your comments in an email to:
  redbooks@us.ibm.com

► Mail your comments to:
  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

► Find us on Facebook:
  http://www.facebook.com/IBMRedbooks

► Follow us on Twitter:
  http://twitter.com/ibmredbooks

► Look for us on LinkedIn:
  http://www.linkedin.com/groups?home=&gid=2130806

► Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

► Stay current on recent Redbooks publications with RSS Feeds:
  http://www.redbooks.ibm.com/rss.html
Introduction

At IBM, we believe your data is yours – and yours alone\(^1\). The insights and advantages that come from your data are yours to use in the pursuit of your business objectives. IBM is dedicated to this mission, and our LinuxONE platform has been designed around this core statement.

The world is experiencing a time of exponential growth in the sheer volume of data, fueled by the digital transformation of systems, services, and interconnected devices that all require strong data serving capabilities. Businesses must be able to manage, store, and most importantly, protect this information while they use it to gain competitive advantage.

IBM LinuxONE is an all-Linux enterprise platform for open innovation that combines the best of Linux and open technology with the best of enterprise computing in ONE system.

In this Redbooks document, we explore the features and capabilities of the LinuxONE platform that extend enterprise-grade security to the open source world.

In this chapter, we provide a high-level overview of the following topics:

- “Introduction to LinuxONE” on page 2
- “Enterprise Security Challenges” on page 2
- “IBM LinuxONE servers” on page 5

1.1 Introduction to LinuxONE

Linux adoption has grown dramatically over recent years, expanding from initial use by startups for web servers, into its use today for a vast range of enterprise computing workloads. It has grown up alongside the open source community, which is a unique resource that leverages a network of passionate and dedicated developers who are willing to contribute to various projects.

IBM LinuxONE is an all-Linux enterprise platform for open innovation that combines the best of Linux and open technology with the best of enterprise computing in ONE system. It has been designed to support customers who want an efficient and cost-effective solution for protecting their data and hosting various enterprise-grade Linux workloads. The LinuxONE platform is defined by security, speed, scalability, reliability, and of course, openness.

LinuxONE’s hardened Linux-based software stack can run most open source software packages, such as databases and data management (in other words, MariaDB, PostgreSQL, MongoDB, and Apache Spark), virtualization platforms, and containers (IBM z/VM, KVM, and Docker), automation and orchestration software (OpenStack, Puppet, Node.js, Juju, and Chef), and compute-intensive workloads, such as blockchain.

1.2 Enterprise Security Challenges

The IBM LinuxONE platform provides unique capabilities to help with overcoming security challenges and differentiating your business offerings. These challenges can range from maintaining regulatory compliance so that you aren’t subject to fines and penalties, to making sure that your critical systems aren’t compromised or taken over by malicious entities.

When you are considering an infrastructure platform, you must understand the security features that are inherent on the platform both in the cloud and on-premises. IBM LinuxONE has been engineering from the ground up to protect your business from all manner of cyberthreats. By providing a highly securable, massively scalable, data serving platform, LinuxONE can help any business that wants to thrive in a data-centric economy.

1.2.1 Data protection and privacy

As shown in a 2019 study\(^2\), there is a 29.6% chance that any given organization will have a data breach within the next two years, with each breach costing an average of $3.92 M. A single data breach can cost your organization millions in lost customers, reduced brand equity, lost revenue, and regulatory fines.

Data protection is the practice of ensuring that data doesn’t fall into the wrong hands, and if it does, ensuring that the data is unreadable or unusable. Encryption is the most effective way to safeguard your sensitive data from theft or misuse by unauthorized parties and has been shown to reduce breach costs by an average of $360,000. Yet of the nearly 15 billion records that have been breached since 2013, only 4% were encrypted.

This is because until recently, encrypting data was time consuming, expensive, and it severely degraded system performance. Businesses that chose to encrypt did it selectively, leaving the rest of their data exposed to threats.

\(^2\) https://databreachcalculator.mybluemix.net/?cm_mc_uid=4152661384021564493627568&cm_mc_sid_50200000=21830241564493627516

There are four levels in which data encryption can be deployed. Figure 1-1 shows that, as you go higher in the pyramid, you gain more security control of your data at the cost of a more complex and intrusive encryption implementation. Conversely, as you go lower, complexity and costs are reduced but with a less granular approach to encryption. Deploying data encryption on one layer or another is a tradeoff that depends on the context and regulatory environment of each client.

**Figure 1-1  Various levels of data protection**

With the latest generations of LinuxONE hardware, IBM has embedded encryption logic and processing onto each processor chip in the system. This allows you to encrypt massive amounts of data with little impact to your system performance.

The on-chip encryption capability is further enhanced by the IBM Crypto Express adapter, which enables industry-leading key protection technology. This combination of integrated cryptographic hardware makes the LinuxONE platform an efficient and cost-effective solution for securely hosting various enterprise-grade Linux workloads.

For more detail on these cryptographic hardware capabilities, see “Secure cryptographic hardware” on page 14.

### 1.2.2 Secure hybrid cloud integration

As enterprise applications are modernized or built for cloud-native computing, it is important to ensure that these workloads and their hosting environments are secure. It’s critical for end users and clients to have confidence that their data is protected yet available from anywhere and on any device. The ability to be fast and flexible in delivering new services, with insight and security, is the key to differentiating a business.

IBM Secure Service Container (SSC) technology is exclusive to LinuxONE and provides an easy-to-deploy secure hosting appliance for container-based applications that run in hybrid cloud environments. SSC is a secure computing environment for microservices-based applications that can be deployed without any application code changes, which makes it an easily consumable solution for cloud-native development. It provides a number of unmatched
security benefits such as automatic pervasive encryption of data in-flight and at-rest, protection from privileged administrators, and tamper protection during installation and start time to protect against malware.

For more information, see “IBM Secure Service Container” on page 26.

This service can be used as part of a private cloud deployment or through public cloud services that are hosted on IBM Cloud™. In other words, a client can implement an on-premises LinuxONE machine to build a private cloud, or they can gain access to the secure services by provisioning a LinuxONE instance on the public cloud.

Running cloud services on LinuxONE allows organizations to take advantage of both IBM and open source software ecosystems while they use container orchestration tools such as Kubernetes in a secure cloud environment. By using automated container management tools in conjunction with enterprise security capabilities, clients can securely build and host their own flexible hybrid and private cloud deployments without compromise.

For more information, see “IBM Cloud Hyper Protect Services” on page 26.

1.2.3 Cyber resiliency and availability

In today’s digital economy, being continuously open for business is a competitive advantage. Your customers expect to transact business with you 24 x 7 with no excuses for interruptions or outages. To this end, cyber resiliency is an often overlooked method for providing business value through security. You could lose access to your core systems due to a ransom ware attack or having unplanned downtime as a result of a distributed denial of service (DDoS) attack. These types of outage can cause very costly business disruptions and productivity losses.

To be competitive, enterprises must provide trusted services with high uptime to their clients, while consistently delivering new value and features. This demand from clients requires a computing platform that accommodates your developers’ creative genius and a highly secure infrastructure that provides instantaneous data delivery at any time, whether you have thousands or millions of simultaneous users.

IBM LinuxONE machines have the industry’s highest reliability levels for over a decade, with up to 99.999% or greater availability. In fact, the experts who track downtime say that the underlying hardware infrastructure is “in a class of its own.”

The LinuxONE platform also features technology to address the next evolution in cyberattacks. The potential of quantum computing is quickly advancing, and will soon explode. This shift will force the entire industry to evolve as quantum computing could break currently secure cryptographic algorithms. The new IBM Crypto Express7S introduces support for quantum-safe signing algorithms to ensure that data can be secured today and well into the age of pragmatic quantum computing. The current generation of quantum-safe cryptographic algorithms were developed internally by IBM, to help prevent the eventual quantum computing attacks of the future.

1.2.4 Industry and regulatory compliance

Many businesses operate on the assumption that adhering to regulatory compliance standards is enough to mitigate business risk and protect a company’s data. While this might have been true in the past, it’s no longer an option to implement a static compliance policy and be done with it.
Cyberthreats in the modern era are constantly evolving and move too quickly for an organization to sit back with passive data protection policies. Protecting only enough data to achieve compliance should be viewed as the bare minimum, not a best practice.

Corporate risk management and compliance is an ongoing cost and effort that only increases over time. The scope of these industry and government regulations is constantly in flux. And it is expanding with additional compliance requirements, not to mention the introduction of completely new mandates such as GDPR (General Data Protection Regulation).

At a high level, global security regulations have varying requirements. But a common thread is that most of them have explicit requirements around encryption of data and access to that data. LinuxONE addresses both of these needs as a core feature of the platform.

Figure 1-2  Data compliance on LinuxONE vs x86

<table>
<thead>
<tr>
<th>IBM LinuxONE capabilities</th>
<th>x86 capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Encrypt everything quickly and economically with hardware acceleration</td>
<td>• Slow/costly encryption due to lack of hardware acceleration</td>
</tr>
<tr>
<td>• Protect against side-channel attacks and insider threats with tamper resistant encrypted keys and Secure Service Containers</td>
<td>• Vulnerable attacks due to clear keys</td>
</tr>
<tr>
<td>• Isolate LPARs at the architectural level with EAL5+ design and HSM crypto isolation</td>
<td>• Weak isolation</td>
</tr>
<tr>
<td>• Protect keys in memory with tamper resistant design (FIPS 140-2 Level 4 HSM) and encryption</td>
<td>• Selectively encrypt sensitive data at best due to slow/costly encryption</td>
</tr>
<tr>
<td>• Hardware-accelerated SSL/TLS encryption</td>
<td>• Little protection against insider threat</td>
</tr>
<tr>
<td>• Limit access to data by encrypting everything</td>
<td>• Large audit scope</td>
</tr>
<tr>
<td>• Remove entire groups of users from audit scope</td>
<td></td>
</tr>
<tr>
<td>• Limit access to sensitive data with Secure Service Containers (even system admins don’t have access!)</td>
<td></td>
</tr>
</tbody>
</table>

LinuxONE provides secure hosting environments where an organization that runs on-premises or cloud-based services can ensure that their end-user’s data is protected at all times. Even IT administrators with physical access to hardware are unable to access data, including sensitive and personally identifiable information.

1.3 IBM LinuxONE servers

The first two LinuxONE products were named Emperor and Rockhopper. Emperor II and Rockhopper II, the second iteration of LinuxONE, were launched in 2017 and early 2018. The new iteration of the system is the IBM LinuxONE III and will be generally available in late 2019.

- IBM LinuxONE III
  The newest member of the LinuxONE family, the IBM LinuxONE III delivers a radically new form factor, featuring a 19" frame that flexibly scales from 1 to 4 frames. It is designed around a new 12-core, 5.2Ghz processor and is configurable with up to 190 processor cores, up to 40 TB of RAM, and 8 TB of RAIM (Redundant Array of Independent Memory) per central processing drawer.
1. IBM LinuxONE Emperor II

This machine features up to 170 processor cores, running at 5.2 GHz, up to 32 TB of RAM, and 640 dedicated I/O processors, all housed in a dual-frame server. It supports tens of thousands of sessions and millions of containers. It can run 8,000 virtual servers and over 30 billion RESTful web interactions per day. This server is dedicated for Enterprise environments.

2. IBM LinuxONE Rockhopper II

The same technology as Emperor II, but at a lower price. It is housed in an industry-standard, 19-inch rack. Rockhopper II is available with up to 8 TB of memory and 30 processor cores, running at 4.5 GHz. It supports hundreds of production and development virtual machines (VMs) in a single footprint.

For more information, see: https://www.ibm.com/it-infrastructure/linuxone

1.3.1 IBM LinuxONE III

The IBM LinuxONE III has radically changed the footprint for LinuxONE servers. It is built with a 19" industry-standard frame that flexibly scales from 1 to 4 frames depending on the configuration that is required. This new form factor maintains approximately the same maximum floor space as prior generations and allows most clients to reduce their floor space significantly. The doors are designed for acoustics and optimized for air flow. The IBM LinuxONE III offers both air-cooled (internal radiator) or water-cooled systems (WCS).

At the heart of the LinuxONE III is the new processor chip, made with 12 cores and leveraging the density and efficiency of 14 nm silicon-on-insulator technology, running at 5.2Ghz, and delivering increased performance and capacity across a wide range of workloads.

There are up to 190 client configurable cores (known as IFLs, or Integrated Facility for Linux). The IBM LinuxONE III comes in a single model (Model LT1) with processor capacity that is represented by feature codes. There are five processor capacity feature codes that you can order for the IBM LinuxONE III: Max34, Max71, Mac108, Max145, and Max190. The numbering signifies that, for example, a Max 34 can configure up to 34 IFLs (cores), Max71 for up to 71 IFLs and so on.

The system offers 8 TB of Redundant Array of Independent Memory (RAIM) per central processing complex (CPC) drawer and up to 40 TB per system. RAIM is intended to provide redundancy for primary memory, sockets, and memory channels for additional reliability and availability. IBM Virtual Flash Memory (VFM) is now located in the RAIM and provides high levels of availability and performance. IBM Adapter for NVMe supports the Non-Volatile Memory express (NVMe) communications protocol, which was built specifically for solid-state drives (SSDs). This feature brings integrated storage to LinuxONE by allowing a client-procured SSD up to 64 TB to be directly connected to the I/O subsystem through an IBM PCIe adapter card. It provides the low latency and high I/O throughput that can help with real-time analytics, memory-intensive and fast storage workloads such as streaming, paging/sorting, and traditional applications such as relational databases.

IBM LinuxONE III also integrates new hardware compression capabilities, delivering greater compression throughput than previous generation systems. This on-chip compression co-processor uses industry standard compression algorithms and can reduce data storage requirements and costs. This compression can also increase data transfer rates to boost throughput above comparable x86 CPUs — all without adversely impacting response times.

For more information about the IBM LinuxONE III, see: https://www.ibm.com/it-infrastructure/linuxone
**LinuxONE III data sheet**

The data sheet for LinuxONE III models is shown in Table 1-1.

### Table 1-1  LinuxONE III at a glance

<table>
<thead>
<tr>
<th>IBM LinuxONE III features</th>
<th>LinuxONE III Models</th>
<th>Cores: Min - Max</th>
<th>Memory: Min - Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cryptography</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crypto-Express7S</td>
<td>LT1</td>
<td>2 - 215</td>
<td>512 GB - 40 TB</td>
</tr>
<tr>
<td>Crypto-Express6S</td>
<td></td>
<td>Minimum 2 features; maximum 30 features</td>
<td></td>
</tr>
<tr>
<td>Crypto-Express5S</td>
<td></td>
<td>Minimum 2 features; maximum 16 features</td>
<td></td>
</tr>
<tr>
<td><strong>Disk Connectivity</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM FICON® Express32S/FICON Express16S+/FICONExpress1 6S/FICON Express8</td>
<td>Maximum: 384 channels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FC Express32S</td>
<td></td>
<td>Maximum: 192 features</td>
<td></td>
</tr>
<tr>
<td>IBM Adapter for NVMe1.1</td>
<td></td>
<td>Maximum: 16 features</td>
<td></td>
</tr>
<tr>
<td><strong>Connectivity</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HiperSockets</td>
<td></td>
<td>Up to 32 high-speed virtual Local Area Networks</td>
<td></td>
</tr>
<tr>
<td>Shared Memory Communications - Direct Memory Access (SMC-D)</td>
<td>Up to 32 ISM virtual CHIPDs</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Supported Linux distributors</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat</td>
<td>Red Hat Enterprise Linux (RHEL) 7.7 and 8.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE</td>
<td>SUSE Linux Enterprise Server (SLES) 12 SP4, and 15SP1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canonical</td>
<td>Ubuntu 18.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Supported hypervisors</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM z/VM</td>
<td>z/VM V7.1, z/VM 6.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KVM</td>
<td>KVM hypervisor, which is offered with the following Linux distributions: RHEL 7.6 or higher, SLES-12 SP2 or higher, Ubuntu 18.04 or higher</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM partitioning technology</td>
<td></td>
<td>Up to 85 LPARs for secure workload isolation</td>
<td></td>
</tr>
</tbody>
</table>

#### 1.3.2 IBM LinuxONE Emperor II

Emperor II is available with up to 170 configurable cores that use a 5.2 Ghz processor for unmatched performance and massive scaling.

The vertical scale allows Emperor II to scale up to 2 million Docker containers in a single system. It can serve up to 30 billion web data requests a day and can host databases that are 20 times larger, without the added cost and latency of fragmenting data across server farms.
An additional 640 processors are dedicated to I/O processing to increase I/O speeds and assure data integrity.

With 32 TB of real memory, Emperor II can open opportunities, such as in-memory data marts, large buffer pools for data access, and in-memory analytics. Advances in the machine instruction set of the processor help to accelerate analytic workloads by using the Vector Packed Decimal Facility, which allows packed decimal operations to be performed in registers rather than memory.

Java improvements, such as pause-less garbage collection, enables vertical scaling. The use of crypto-acceleration delivers more improvements in throughput per core, which provides a boost to Java processes that use cryptographic functions.

For more information about the IBM LinuxONE Emperor II, see: https://www.ibm.com/us-en/marketplace/linuxone-emperor-ii

**LinuxONE Emperor II data sheet**
The data sheet for Emperor II models is shown in Table 1-2.

<table>
<thead>
<tr>
<th>IBM LinuxONE Emperor II features</th>
<th>Cores: Min - Max</th>
<th>Memory: Min - Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emperor II Models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LM1</td>
<td>1 - 33</td>
<td>256 GB - 8 TB</td>
</tr>
<tr>
<td>LM2</td>
<td>1 - 69</td>
<td>256 GB - 16 TB</td>
</tr>
<tr>
<td>LM3</td>
<td>1 - 105</td>
<td>256 GB - 24 TB</td>
</tr>
<tr>
<td>LM4</td>
<td>1 - 141</td>
<td>256 GB - 32 TB</td>
</tr>
<tr>
<td>LM5</td>
<td>1 - 170</td>
<td>256 GB - 32 TB</td>
</tr>
<tr>
<td><strong>Cryptography</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crypto-Express6S</td>
<td>Minimum 2 features; maximum 16 features</td>
<td></td>
</tr>
<tr>
<td>Crypto-Express5S</td>
<td>Minimum 2 features; maximum 16 features</td>
<td></td>
</tr>
<tr>
<td><strong>Disk Connectivity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FICON Express16S+/FICON Express16S/FICON Express8</td>
<td>Maximum: 320 ports</td>
<td></td>
</tr>
<tr>
<td><strong>NIC - Connectivity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 GbE RoCE Express2</td>
<td>Maximum 8; minimum recommended: 2</td>
<td></td>
</tr>
<tr>
<td>OSA - Express6S</td>
<td>Maximum: 96 ports</td>
<td></td>
</tr>
<tr>
<td>OSA - Express5S</td>
<td>Maximum: 96 ports</td>
<td></td>
</tr>
<tr>
<td><strong>High-speed “Virtual” LANS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HiperSockets</td>
<td>Up to 32 connections</td>
<td></td>
</tr>
<tr>
<td><strong>Supported Linux distributors</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat</td>
<td>Red Hat Enterprise Linux (RHEL) 6 and 7</td>
<td></td>
</tr>
<tr>
<td>SUSE</td>
<td>SUSE Linux Enterprise Server (SLES) 11 SP4, SLES 12 SP2, and SLES 15</td>
<td></td>
</tr>
</tbody>
</table>
### 1.3.3 IBM LinuxONE Rockhopper II

Rockhopper II delivers secure capabilities in a 19-inch frame with a lower cost of entry that can coexist with other platforms in any cloud data center. It is built on the strong foundation of the LinuxONE Emperor II platform.

Rockhopper II is housed in an industry-standard, 19-inch IBM-supplied rack. The design includes power distribution unit (PDU)-based power and redundant power, cooling, and power cords. These features allow you to install Rockhopper II within any data center with a server that is rated at ASHRAE A3. Up to 16U of available frame space can be used in the new 19-inch rack design.


### LinuxONE Rockhopper II data sheet

Table 1-3 provides some basic information about Rockhopper II models.

<table>
<thead>
<tr>
<th>IBM LinuxONE Rockhopper II features</th>
<th>Rockhopper II models</th>
<th>Cores: Min - Max</th>
<th>Memory: Min - Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canonical</td>
<td>LR1 Max4</td>
<td>1 - 4</td>
<td>64 GB - 2 TB</td>
</tr>
<tr>
<td>Canonical</td>
<td>LR2 Max12</td>
<td>1 - 12</td>
<td>64 GB - 4 TB</td>
</tr>
</tbody>
</table>

| Canonical                            | Ubuntu 16.04 LTS and Ubuntu 18.04 LTS |

<table>
<thead>
<tr>
<th>Supported hypervisors</th>
<th>Canonical Ubuntu 16.04 LTS and Ubuntu 18.04 LTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM z/VM</td>
<td>z/VM 6.4 (until the EOS) and z/VM 7.1 or higher</td>
</tr>
</tbody>
</table>

KVM hypervisor, which is offered with the following Linux distributions: SLES-12 SP2 or higher, Ubuntu 16.04 or higher, and RHEL 7.5 or higher

<table>
<thead>
<tr>
<th>IBM partitioning technology</th>
<th>Up to 85 LPARs for secure workload isolation</th>
</tr>
</thead>
</table>

**Table 1-3**: IBM LinuxONE Rockhopper II at a glance - need to update with GA2 features
<table>
<thead>
<tr>
<th>IBM LinuxONE Rockhopper II features</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR3 Max24</td>
</tr>
<tr>
<td>LR4 Max30</td>
</tr>
</tbody>
</table>

**Cryptography**

Crypto-Express6S/Crypto Express5S  
Minimum 2 features; maximum 16 features

**Disk connectivity**

FICON Express16S+/FICON Express16S / FICON Express8S  
Maximum features (two ports per feature)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max4</td>
<td>16</td>
</tr>
<tr>
<td>Max12</td>
<td>32</td>
</tr>
<tr>
<td>Max24, Max30</td>
<td>64</td>
</tr>
</tbody>
</table>

**NIC - Connectivity**

10 GbE RoCE Express2, 10 GbE RoCE Express  
4 Maximum features (two ports per feature); minimum recommended is 2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OSA-Express6S / OSA-Express5S / OSA-Express4S / 1000-BaseT</td>
<td>Maximum features (two ports per feature)</td>
</tr>
<tr>
<td>Max4</td>
<td>16</td>
</tr>
<tr>
<td>Max12</td>
<td>32</td>
</tr>
<tr>
<td>Max24, Max30</td>
<td>48</td>
</tr>
</tbody>
</table>

**High Speed “Virtual” LANs**

HiperSockets  
Up to 32 high-speed “virtual” local area networks

**Supported Linux distributors**

Red Hat  
Red Hat Enterprise Linux (RHEL) 6 and 7

SUSE  
SUSE Linux Enterprise Server (SLES) 11 SP4, SLES 12 SP2, and SLES 15

Canonical  
Ubuntu 16.04 LTS and Ubuntu 18.04 LTS

**Supported Hypervisors**

IBM z/VM  
z/VM 6.4 (until the EOS) and z/VM 7.1 or higher

KVM  
KVM hypervisor, which is offered with the following Linux distributions: SLES-12 SP2 or higher, Ubuntu 16.04 or higher, and RHEL 7.5 or higher

IBM partitioning technology  
Up to 40 LPARs for secure workload isolation

**Typical Physical Weight**

Minimum configuration weight of new build 735 kg (1621 lb)  
Maximum configuration weight of new build 795 kg (1753 lb)

<p>| Weight without side covers | Without overhead cabling 735 kg (1621 lb) | With overhead cabling adds approx. 12 lbs (5 kg) 1633 lbs (740 kg) |</p>
<table>
<thead>
<tr>
<th>IBM LinuxONE Rockhopper II features</th>
<th>Without overhead cabling</th>
<th>With overhead cabling adds approximately &lt;12 lbs (5 kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight with side covers adds approx. 42.7 lbs (19.4 kg)</td>
<td>754 kg (1663 lb)</td>
<td>1675 lbs (760 kg)</td>
</tr>
<tr>
<td><strong>Note:</strong> Optional seismic resistance hardware adds approximately 35 kg (78 lb)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Dimensions (D x W x H) without side covers</td>
<td>Without overhead cabling: 107 x 60 x 201.5 cm (42.1 x 23.6 x 79.3 in)</td>
<td>With overhead cabling increases height 107 x 60 x 212.3 cm (4.3 inches 42.1 x 23.6 x 83.6 in)</td>
</tr>
<tr>
<td>Product Dimensions (D x W x H) with side covers</td>
<td>Without overhead cabling: 120.4 x 62.4 x 202 cm (47.4 x 24.6 x 79.5 in)</td>
<td>With overhead cabling increases height 120.4 x 62.4 x 212.8 cm (47.4 x 24.6 x 83.8 in)</td>
</tr>
<tr>
<td>Airflow (Capacity of Exhaust)</td>
<td>2000 cubic meters per hour (1200 CFM)</td>
<td></td>
</tr>
</tbody>
</table>
Core Security Technologies on LinuxONE

A secure digital business starts at the hardware level. LinuxONE allows you to establish foundational system integrity across physical and virtual infrastructure to address rapidly evolving security threats to your enterprise.

In this chapter, we cover the core technologies that are embedded in the LinuxONE hardware for ensuring a pervasive level of data protection.

- Section 2.1, “Secure cryptographic hardware” on page 14
- Section 2.2, “Virtualization technology” on page 21
2.1 Secure cryptographic hardware

Traditionally, encryption of all your data would require a large amount of time and computation overhead. This is due to the limitations of software-based encryption, which shares computing resources with the rest of your system, and can slow down other shared applications as a result.

While it can be cheap to get started with software-based encryption, it quickly becomes prohibitively expensive as you scale your business. Especially with more secure forms of encryption (more complex algorithms, longer bit values) it becomes very resource intensive. So, you typically must pay more for additional processing power for the same level of performance.

The LinuxONE solution to this problem is through dedicated hardware that is tuned for encryption and can encrypt 100% of data that is at-rest and in-flight, by default, with minimal compute overhead. This level of protection is achieved through hardware accelerated encryption capabilities and does not require any code or application changes.

Types of cryptographic keys
An important starting point for understanding the cryptographic hardware on LinuxONE is the types of encryption keys that are used. LinuxONE cryptography uses the following types of encryption keys that are listed in Table 2-1.1

<table>
<thead>
<tr>
<th>Key type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>A data-encrypting key that is used to encrypt and decrypt data.</td>
</tr>
<tr>
<td>Key-encrypting</td>
<td>A key that encrypts or wraps other keys.</td>
</tr>
<tr>
<td>Effective</td>
<td>A type of data-encrypting key; also called a data key that is wrapped by a key-encrypting key (KEK).</td>
</tr>
<tr>
<td>Master</td>
<td>A special KEK that is in a tamper-responding, Crypto Express adapter only and sits at the top of a KEK hierarchy. Loading and managing the master key can be done by using the Trusted Key Entry (TKE) workstation.</td>
</tr>
<tr>
<td>CPACF wrapping</td>
<td>A special key-encrypting key that is generated at LPAR activation and is in the Hardware System Area, which is inaccessible to applications and the operating systems. It is used to create protected keys.</td>
</tr>
<tr>
<td>Secure</td>
<td>Key values are encrypted under a Master Key and no key ever appears in unencrypted form outside of the Crypto Express HSM. Crypto operations are performed only within the Crypto Express HSM</td>
</tr>
<tr>
<td>Clear or plain</td>
<td>A data-encrypting key that is not encrypted by any other key. The key material is in plain text.</td>
</tr>
<tr>
<td>Protected</td>
<td>Key values are encrypted under a CPACF wrapping key. Crypto operations are performed by using only CPACF. When the key is not in use, it is protected by the Crypto Express HSM. In the case of Linux in a native LPAR, the wrapping key is specific to the LPAR. However, for guests of z/VM or KVM, the wrapping key is specific to the guest.</td>
</tr>
<tr>
<td>Operational</td>
<td>A key that is not a master key or KEK, such as a data-encrypting key (which can be clear, secure, or protected).</td>
</tr>
</tbody>
</table>

The tradeoffs between clear key, protected key, and secure key encryption implementations are described in Figure 2-1.

Cryptographic key management
Cryptographic key management is a complex task that must be managed according to strict policies. You must account for various legal, regulatory, and compliance requirements. Your key management system should allow authorized persons a method for key identification, exchange, separation, update, backup, and management.

For more information about key management, see “Cryptographic Key Management for LinuxONE” on page 27

2.1.1 Central Processor Assist for Cryptographic Functions

The hardware accelerated encryption capabilities of LinuxONE are enabled by Central Processor Assist for Cryptographic Functions technology (CPACF, for short). The CPACF on-chip encryption co-processor is on every compute chip next to the main processor and can encrypt up to 13 GB of data per second per core. This has been shown to give performance improvements of up to 6x and is best suited for symmetric, high-speed bulk encryption.

CPACF supports the AES, DES, TDES, SHA-1, SHA-2, SHA-3, and Elliptic Curve Cryptography (ECC) algorithms. ECC support on CPACF for LinuxONE III is something new that helps accelerate applications such as IBM Blockchain Platform. In these cases, you don’t have to change the application code but can still take advantage of better performance and acceleration throughput of ECC functions such as EdDSA (Ed448, Ed25519), ECDSA (P-256, P-384, P-521), and ECDH.

CPACF is a no-charge feature of LinuxONE. If you have the hardware, you can enable the technology and start to get the benefits right away.
True Random Number Generator

An additional feature of LinuxONE cryptographic hardware is the ability to generate irreproducible, unique data with the on-chip true random number generator (TRNG). This capability is the basis for generating high-quality cryptographic keys. TRNG is an improvement of Deterministic RNG because the numbers that are generated are more random.

Protected key encryption

The ability to use protected key encryption with CPACF is a differentiating feature on LinuxONE. Many encryption services use plaintext “clear keys” that are stored unsecured in main system memory. These clear keys are visible and vulnerable during the encryption and decryption process. Clear keys can be stolen from memory or system dumps, meaning that your encrypted data is now at risk.

Protected key technology uses CPACF for encrypting data at high speeds without exposing keys to main system memory. The data-encrypting (protected) keys are encrypted or “wrapped” by a special key that is stored in a secure environment. In that environment, it is inaccessible to applications, hypervisor, and operating system (known as the HSA, or Hardware System Area). Among the many use cases, this technology enables fast and highly secure encryption and decryption of complete disks (volumes) or selected partitions.

With the introduction of LinuxONE III, CPACF supports the creation of protected key signatures.

2.1.2 IBM Crypto Express adapter

LinuxONE can also include IBM Crypto Express adapters, a Hardware Security Module (HSM), and a cryptographic co-processor that supports high-speed, asymmetric encryption. This co-processor also supports a symmetric cryptographic function to assist key encryptions, encrypt data, compute message authentication codes, protect financial PINs, secure EMV card transactions, and many other things. This specialized hardware performs AES, DES, TDES, RSA, ECC, SHA-1, SHA-2, SHA-3, and other cryptographic operations.²

An HSM is designed to withstand both physical and logical attacks and has special hardware to perform cryptographic operations and protect keys. The HSM is accessed from a host computer that uses a set of generic or specialized API functions. The IBM Crypto Express HSM can support two primary secure key cryptographic modes, APIs, and one clear key accelerator mode, and you can reload your HSM firmware at any time to switch from one to the other.

- The first mode is the IBM Common Cryptographic Architecture (CCA). CCA provides a set of general-purpose cryptographic functions, and its primary strength is support of finance industry payments applications. This mode is also often called as cryptographic co-processor mode, and many IBM software products support CCA modes to enhance security.
- The second mode is IBM Enterprise PKCS #11 (EP11), which supports the industry standard PKCS #11 API. EP11 is specifically designed for customers who seek support for open standards and enhanced security. It offers a wide variety of general purpose, secure-key-only cryptography functions.
- The last mode is Accelerator mode, which makes Crypto Express hardware become an asymmetric cryptographic function accelerator. When tested with a Crypto Express 6S running on LinuxONE as an accelerator mode with 8 concurrent processes, it could

process about 9 K SSL/TLS handshakes per second with 2048-bit RSA key.\(^3\) This mode might be very useful for enterprise web services.

Figure 2-2 shows new Crypto Express Adapter options for LinuxONEIII with enhanced performance and throughput, and support for new cryptographic algorithms.

For more information on these cryptographic devices, see the following web page:  
https://www.ibm.com/security/cryptocards/hsms

**HSM Certifications**

The IBM Crypto Express adapter is a Hardware Security Module that is certified for PCI-HSM and FIPS 140-2 Level 4, the highest level available in industry and one that very few devices are able to attain. This means the cryptographic co-processors are protected within a tamper-resistant and tamper-responsive environment that erases encryption keys if it senses an attack, eliminating the risk of exposing cryptographic keys during a breach. The following section describes how the Crypto Express adapter HSM can provide the maximum protection for your encryption keys.

**FIPS certification levels**

The Federal Information Processing Standard (FIPS) Publication 140-2, (FIPS PUB 140-2), is a U.S. government computer security standard that is used to approve cryptographic modules. It defines what areas of security requirement to meet the standard in terms of design and behaviors of the module. This includes specification, ports and interfaces, roles and services, and authentication; finite state model; physical security; operational environment; cryptographic key management; electromagnetic interference/electromagnetic compatibility (EMI/EMC); self-tests; design assurance; and mitigation of other attacks.\(^4\)

---

\(^4\) [https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf)
You can read more about the FIPS 140-2 standards in the sources that are mentioned above. However, Figure 2-3 on page 18 shows a summary of what each level of FIPS 140-2 is certified to protect against. You can see that the level 4 device meets the highest requirements, and IBM Crypto Express adapter for LinuxONE is designed to meet those level 4 requirements.

Level 1 - No physical security features required

Level 2 - Tamper-evident physical security features

Level 3 - Tamper-responding features designed to notify of unauthorized access

Level 4 - Complete tamper-responding envelope of protection that immediately deletes all plaintext keys upon detection of unauthorized access.

<table>
<thead>
<tr>
<th>FIPS 140-2 Security Levels</th>
<th>Security Level 1</th>
<th>Security Level 2</th>
<th>Security Level 3</th>
<th>Security Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>At least one cryptographic algorithm or security function implemented</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Tamper evidences</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>An attacker leaves visible traces. The attack may have been successful.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tamper detection and response</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Attempts at removal or penetration of the strong enclosure will have a high probability of causing serious damage to the module (i.e. the module will not function).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enhanced protection of secret and private keys</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Key entry and output only encrypted or in split-knowledge procedure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identity-based authentication</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>The operator be individually identified.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tamper resistance</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Including active and immediate zerorization of plain text secret keys in case of attacks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supported on Crypto Express adapter with LinuxONE!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environmental Failure Protection</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Protection against attacks using extreme voltage or temperature changes from outside.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supported on Crypto Express adapter with LinuxONE!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Secure key encryption**

The HSM feature of the IBM Crypto Express adapter enables secure encryption with the capability to protect cryptographic keys that use a special key called Master Key. (We use the terms HSM and Crypto Express adapter interchangeably in this document.) The master key is used to encrypt the keys used by your applications, which are stored outside of the Crypto Express HSM. Those keys are fully protected, because the Master Key is protected by the security features of the HSM. The Master Key is stored in Crypto Express adapter hardware. Loading and managing the Master Key into the Crypto Express adapter can be done in software, but it is highly recommended to perform it with the Trusted Key Entry (TKE)
workstation. We cover more details about TKE and key management in Chapter 3, “Exploiters of Security on LinuxONE” on page 25 and Chapter 4, “Use cases” on page 33.

Secure key encryption uses a wrapped key for encrypting data that never appears in clear text outside of a secure environment. The Crypto Express adapter ensures that these keys are never exposed in the clear. Any unauthorized attempts to access the Crypto Express adapter enclosure result in the deletion of the stored keys.

Figure 2-4 on page 19 shows a high-level diagram and workflow of how master keys and secure keys are used in LinuxONE. There are four different types of Master Keys for Crypto Express Adapter: AES, DES, RSA, and ECC. For each Master Key type, the adapter can set up to 85 domains, which is designed to support up to 85 LPARs in LinuxONE II and III system.

Creating a secure key - example
At the time of writing this book, the current Crypto Express for LinuxONE supports AES encryption for the user-defined secure key. The hardware is designed to support RSA, DES, AES, and ECC to be the secure keys, so keep an eye on the support page for new updates.5

The following diagram in Figure 2-5 on page 20 shows an example of use case of secure key, a high-level process that is used to create a secure key for a LUKS2 format volume to encrypt a disk volume by using secure key and protected key. We also show another example in Chapter 4.

---

This process includes the following steps:

1. A secure key is created by using a `zkey` command. The `zkey` utility generates the secure key with the help of the `pkey` utility and an assigned Crypto Express adapter (with master key). The secure key is also stored in the key repository.

2. The use of the `zkey cryptsetup` command generates output strings that are copied and pasted to the `cryptsetup` command to create the encrypted volume with the appropriate secure key.

3. The `cryptsetup` utility formats the physical volume and writes the encrypted secure key.

A quick example of this process is shown in Figure 4 on page 33. See the following Redbooks document for more information:

2.2 Virtualization technology

Modern data serving systems must be able to scale up and scale out in size, performance, and features with the drop of a hat. Virtualization — essentially creation of a virtual version of a computing system with a subset of total resources — is a key technology that enables any hardware system to achieve this level of scaling performance.

Thankfully, virtualization is one of the core strengths of the LinuxONE platform. The LinuxONE platform is extremely virtualized, with the goal of maximizing utilization of computing resources, while it lowers the overall cost and resource requirements for running critical workloads for enterprises.

The embedded architecture and hardware on LinuxONE is thoughtfully designed around the ability to partition resources (compute, memory, storage, network) to be used independently in distinct virtualized environments.

**Virtual Machine (VM)**

A virtual machine is a virtual environment that provides the core functionality of a single physical computer. It typically runs its own operating system and uses a fraction of the host server’s total compute, memory, storage, and network resources. VMs are sometimes referred to as guests or images. As the name says, LinuxONE can host many distributions of Linux as guests, including Red Hat, SuSE, and Ubuntu.

**Hypervisor**

The hypervisor is a core part of the virtualization technology stack. Hypervisors are designed to enable simultaneous execution of multiple operating systems and allocating the correct amount of virtual resources. The hypervisor is necessary to run and manage other virtual machines guests. LinuxONE supports three key virtualization technologies: z/VM, KVM, and PR/SM. Each virtualization technology has its own strengths and benefits to the enterprise applications. We take a closer look at these hypervisors in the next section.
2.2.1 PR/SM and LPARs

The IBM LinuxONE system has a unique capability to implement a hypervisor at the hardware/firmware level. The hardware hypervisor is IBM PR/SM (Processor Resource/Systems Manager, informally called PRISM). PR/SM is implemented in firmware as a part of the base system that fully virtualizes the system resources and runs without any additional software.

PR/SM is a Type-1 hypervisor that runs directly on bare metal. With it, you create multiple isolated partitioned environments on the same physical server. These isolated environments are known as logical partitions or LPARs.

**EAL 5+ isolation and cryptographic key protection**

LinuxONE systems have EAL 5+ isolation and cryptographic key protection\(^6\). EAL5+ is a regulatory certification for logical partitions (LPARs) that verifies the separation of partitions to improve security. Therefore, you can run many virtual servers concurrently, and use LinuxONE’s ability to isolate and protect each virtual server as though they were running on physically separated servers.

LinuxONE’s PR/SM-based LPARs are the only technology commercially available that can provide this highly certified level of isolation between workloads.

Isolation and cryptographic key protection is achieved by using a dedicated cryptographic coprocessor. The CP Assist for Cryptographic Function (CPACF) delivers cryptographic and hashing capabilities in support of clear-key operations. The Crypto Express adapter is used to create the fortified data perimeter by using the IBM LinuxONE protected key in which the keys that are used in the encryption process are not visible to the applications and operating system.

Each LPAR on a LinuxONE system has its own uniquely generated and assigned cryptographic keys that are held in a secure hardware area. This provides a level of cryptographic isolation between secure environments that is specifically called for under many regulatory compliance frameworks (PCI-DSS, for example).

2.2.2 KVM

KVM (Kernel-based Virtual Machine) is the most popular open source Linux hypervisor and a key technology for the LinuxONE platform. It is a Type-2 hypervisor that provides simple, cost-effective virtualization technology for Linux workloads. It allows sharing of CPU, memory, and I/O resources and can coexist with other types of virtualization technologies simultaneously running on LinuxONE. KVM on LinuxONE frees operators to adopt and switch from various hardware platforms, and more familiar interfaces. More use cases and examples can be found here: [http://kvmonz.blogspot.com/](http://kvmonz.blogspot.com/)

One of the advantages for KVM virtualization is the familiar standard Linux user interfaces for open source developers, offering a low barrier to adoption and easy integration with hybrid environments.

KVM on LinuxONE is supported through the following Linux distribution partners:
- Red Hat Enterprise Linux (RHEL)
- SuSE Linux Enterprise Server (SLES)
- Canonical Ubuntu

---

\(^6\) [https://www.commoncriteriaportal.org/files/epfiles/0900a_pdf.pdf](https://www.commoncriteriaportal.org/files/epfiles/0900a_pdf.pdf)
2.2.3 z/VM

z/VM is IBM’s internally developed Type-2 hypervisor that manages the sharing of a LinuxONE system’s physical resources between virtual guests. The z/VM hypervisor typically runs on an LPAR and manages Linux virtual machines. It also can manage other types of operating systems, including z/VM itself, on top of the hypervisor. z/VM is a proven and established virtualization platform with industry-leading capabilities for very efficient vertical and horizontal scaling that has been proven over many decades.7

Additional security capabilities on z/VM

z/VM supports encrypted paging, in support of the philosophy of encrypting all data in flight and at rest (available with z/VM V7.1 and z/VM V6.4). Ciphering occurs as data moves between active memory and a paging volume.

IBM RACF® for z/VM provides security systems that include access control and auditing functionality as the backbone for Linux security.

7 https://www.ibm.com/it-infrastructure/z/zvm
Exploiters of Security on LinuxONE

Building on the secure foundation of the LinuxONE platform, IBM is continuously developing new technology further up the stack that can exploit the system’s industry-leading security capabilities.

Organizations can take advantage of these unique offerings to differentiate their services from competitors with the power and speed that users demand, the security-rich environment that businesses and regulators require, and efficiencies that lower operational expenditures.

In this chapter, we cover several secure technologies that run on LinuxONE:

- Section 3.1, “IBM Secure Service Container” on page 26
- Section 3.2, “IBM Secure Boot for Linux” on page 26
- Section 3.3, “IBM Cloud Hyper Protect Services” on page 26
- Section 3.4, “Cryptographic Key Management for LinuxONE” on page 27
3.1 IBM Secure Service Container

The IBM Secure Service Container (SSC) is a solution that hosts container-based applications for hybrid and private cloud workloads and is exclusive to IBM LinuxONE. This secure computing environment can be deployed without any application code changes for the following advanced security capabilities and benefits:

- Tamper protection during installation and start time to protect against malware attacks
- Restricted administrator access to help prevent the misuse of privileged user credentials for cloud and on-premises environments
- Automatic pervasive encryption of data that is in flight and at rest

IBM Secure Service Container uses LinuxONE’s EAL5+ certification for vertical isolation of workloads and achieves horizontal isolation that separates the running application from the underlying host environment. Secure Service Container is designed to offer the highest security level available for protected key management (FIPS 140-2 level 4).

IBM Secure Service Container technology builds on the workload isolation of the firmware that is based LPARs and is unique to IBM LinuxONE. It was used in the IBM Cloud on LinuxONE to provide the advanced security of IBM Blockchain Platform and is now extended for generic container-based applications through IBM LinuxONE.

For more information, see this web page:

3.2 IBM Secure Boot for Linux

IBM Secure Boot for Linux brings boot integrity to the LinuxONE platform -- a complete chain of trust from trusted source to boot loader. Secure Boot is part of the Unified Extensible Firmware Interface (UEFI) -- a central interface between the firmware, operating system, and individual components of a computer.

This capability protects your system from root level attacks and viruses that target vulnerabilities during the boot process. The system checks images at boot time for a vendor-signed cryptographic key to verify that the image is from an official provider, and that the image hasn't been tampered with or replaced by malicious third parties.

This feature can be enabled through a simple interface option on the Hardware Management Console. The system firmware first confirms that the system boot loader is signed with a verified cryptographic key. The system then confirms that the key has been authorized by a database that is contained in the firmware and only recognized keys allow the system to boot.

3.3 IBM Cloud Hyper Protect Services

Built on IBM LinuxONE technology, IBM Cloud Hyper Protect Services provide built-in data-at-rest and data-in-flight protection to help developers easily build applications with highly sensitive data.

These cloud services are infused with enterprise-grade data protection and are made possible by bringing IBM LinuxONE into IBM’s global public cloud data centers.
Now, developers and clients can build, deploy, and host applications with an industry-leading data protection that encrypts information that is at rest and in flight. This technology is designed to help protect against threats inside and outside of an organization.

The IBM Cloud Hyper Protect family provides the following services and intends to expand to include others that are crucial for providing protected cloud capabilities:

- IBM Cloud Hyper Protect Crypto-Services
- IBM Cloud Hyper Protect DBaaS

### 3.3.1 IBM Cloud Hyper Protect Crypto Services

IBM Cloud Hyper Protect Crypto Services is a key management and cloud hardware security module (HSM). It is designed to enable you to take control of your cloud data encryption keys and cloud hardware security models, and is the only service in the industry that is built on FIPS 140-2 Level 4-certified hardware.

Built on IBM LinuxONE technology, the service helps ensure that only you have access to your keys. A single-tenant key-management service with key vaulting provided by dedicated customer-controlled HSMs helps you to create encryption keys with ease.

Alternatively, you can bring your own encryption keys to manage. The managed cloud HSM supports industry standards, such as PKCS #11, so that your applications can integrate cryptographic operations like digital signing and validation. Keep your own keys for cloud data encryption protected in a dedicated cloud hardware security module (HSM). Maintain control of the key hierarchy, including the HSM master key.

For more information, see the following web page:
https://www.ibm.com/cloud/hyper-protect-crypto

### 3.3.2 IBM Cloud Hyper Protect DBaaS

As business leaders look to leverage the cloud, enterprises in highly regulated industries are naturally concerned about protecting confidential and sensitive customer data. This leading-edge solution offers a highly secure database environment for enterprise workloads with sensitive data. With IBM Cloud Hyper Protect DBaaS, you can provision, manage, maintain, and monitor multiple database types like MongoDB and PostgreSQL through standardized APIs.

Hyper Protect DBaaS is built on LinuxONE technology, which provides built-in data encryption along with excellent vertical scalability and performance. It helps to protect against threats of data breaches and data manipulation by privileged users and provides a high level of data confidentiality for data owners.

For more information, see the following web page:
https://www.ibm.com/cloud/hyper-protect-dbaas

### 3.4 Cryptographic Key Management for LinuxONE

As shown in Chapter 2, different types of keys are used for different encryption algorithms. The “Secrecy of Data” is maintained by the encryption keys, not by the encryption algorithms. Although in the early days of cryptography, the encryption algorithm itself was regarded as the protector of the information, this has been proven to not always be true.
The method of storing keys is important. It is also important how often you change the keys before they get too old. It is often referred to as the 'life-cycle' of a key or a pair of keys that get generated (born), used (lives) and the destroyed/changed (dies). When keys are created and stored it's also very important to decide who can create and distribute the key, and decide on owners for a key that is split into different segments. The National Institute of Standards and Technology (NIST) defines this role as a key custodian, and every enterprise should assign that role to a party responsible for key lifecycle management.

In this section, we look at this key lifecycle management in two parts:

- “Operational Key Lifecycle Management” on page 28 - Operational Encryption Key Management with IBM Security Key Lifecycle Manager (ISKLM)
- “Master Key Lifecycle Management” on page 29 - Master Key Management for Crypto Express Adapter with Trusted Key Entry (TKE)

### 3.4.1 Operational Key Lifecycle Management

Key lifecycle management is a critical aspect in any encryption strategy. Cryptographic keys feature a lifecycle that includes tasks, such as key creation, key activation, key deactivation, key archival, and key deletion. Some regulations, such as European Union (EU) General Data Protection Regulation (GDPR), Payment Card Industry Data Security Standard (PCI-DSS), and Health Insurance Portability and Accountability Act (HIPAA), require key management processes to be created and well-documented.

**Encryption Algorithms and Sharing Keys**

When an application needs to encrypt and decrypt data, it chooses standard encryption algorithms to communicate with other applications and with the other entity of the application. Examples of these encryption standards are RSA, DES, AES, ECC, and SHA. Each standard has a specific purpose for its use. When an application encrypts data, it typically uses multiple sets of encryption algorithms to make it more difficult for unauthorized parties to decrypt the data.

For example, if a database uses AES (Advanced Encryption Standard from NIST, the US National Institute of Standards and Technology) to encrypt its table data, the database also uses the RSA cryptosystem to encrypt the AES encryption key. As a result, the key can be shared with other applications in a secure way.

There are many variations and combinations of encryption algorithms that can be used and shared. Managing the keys for those methods also varies, but there are some popular standards such as the Key Management Interoperability Protocol (KMIP)

**IBM Security Key Lifecycle Manager - ISKLM**

IBM Security Key Lifecycle Manager (ISKLM) offers a simple integration to your enterprise applications and infrastructures to manage operational cryptographic keys. ISKLM centralizes, simplifies, and automates the encryption key management process to help minimize risk and reduce operational costs of encryption key management. It offers secure, robust key storage, key serving, and key lifecycle management for IBM and non-IBM storage solutions that use the OASIS Key Management Interoperability Protocol (KMIP). IBM Security Key Lifecycle Manager helps customers meet regulations such as the Payment Card Industry

---

1. KMIP is an open standard method to standardize the key management within the companies.
2. PKCS is a set of specifications that have been developed for public key cryptography, initially developed by RSA Data Security Inc.
Data Security Standard (PCI DSS), Sarbanes-Oxley, and the Health Insurance Portability and Accountability Act (HIPAA) by providing centralized management of encryption keys.\(^3\)

ISKLM runs on LinuxONE and supports Red Hat Enterprise Linux and SuSE Linux Enterprise Server. It is also integrated with both KMIP and PKCS\#11. So, you can use it to manage keys for storage devices such as IBM DS8000\(^\circ\), Spectrum Scale, Virtual Tape libraries, and also middleware like IBM Db2\(^\circ\) for LinuxONE.

As an example, if Db2 for LinuxONE is set up to use Db2 Native Encryption, it uses the IBM GSKit interface to exploit CPACF hardware in CPU to accelerate AES encryptions and decryptions. Db2 Native Encryption also supports KMIP, where ISKLM can be used to manage operational keys for the encryption key lifecycle management. If ISKLM is hosted on a LinuxONE server with the proper JAVA SDK level (in other words, JAVA SDK v7/v8), it uses IBMPKCS11Impl to exploit the Crypto Express Adapter to use hardware acceleration for PKCS\#11 functions. You can also use the `pkcsconf -m` command to display the supported mechanisms for each slot on a LinuxONE system.\(^4\)

Figure 3-1  ISKLM Overview

### 3.4.2 Master Key Lifecycle Management

The Trusted Key Entry (TKE) workstation is an optional feature of LinuxONE that manages cryptographic keys in a secure environment. The TKE workstation is an integrated solution that contains a combination of hardware, firmware, and software to provide a basic key management tool for the cryptographic coprocessors. TKE securely manages multiple cryptographic modules that run in Common Cryptographic Architecture (CCA) or IBM Enterprise PKCS\#11 (EP11) and use compliant-level hardware-based key management techniques from a single point of control.

---


Maximizing Security with LinuxONE

Trusted Key Entry (TKE) workstation is a platform that provides convenient way to manage Master Keys in the Hardware Security Module (HSM), in this case Crypto Express Adapter for LinuxONE.

**Master keys on LinuxONE**

When a Crypto Express Adapter is configured to wrap an encryption key (in other words, a key encrypting key, also known as *secure key*), the root key that is used to encrypt secure keys (also known as the *master key*) is stored in the hardware. This master key that is stored in the hardware never leaves its entity, and is processed only within the hardware to decrypt a key when called by a hardware function. Therefore a secure key (encrypted key) that transported into a Crypto Express adapter leaves the hardware unencrypted through the secure communication channel to the processor unit, and is used to encrypt or decrypt actual data with a specific encryption algorithm that is tied to that key.

Many organizations have security policies or specific requirements about how to set the master keys, and how often they should be changed/rotated. Most of the time, the master keys are divided into multiple pieces and distributed to multiple key custodians. At the time of rotating the master keys or generating new master keys, all the key custodians get together along with witnesses to set the new master keys. This is usually called 'Key Ceremony', where all the required persons get together to securely change the master key. This adds another layer of security into the process, making it harder for the keys to be exposed.

**TKE: Why do you need it?**

You can use a *panel.exe* software tool to change master keys in a Crypto Express Adapter for LinuxONE. It is a free command-line tool that provides a very basic interactive session to set up the master keys and change them when needed. For more information on using this command-line interface, see these resources:

- **Getting Started with Linux on Z Encryption for Data At-Rest:**
- **“The panel.exe utility” topic in IBM Knowledge Center:**

To make a new master key and/or change the current master key in the HSM, you must enter the current master key into the terminal by using 'panel.exe' to load the keys into the Crypto Express Adapter. As you might guess, this method has many potential security exposures because it is connected through the end-user's terminal to LinuxONE. The keys that are entered on terminal might be exposed as follows:

- In host memory on the system where they are entered
- On the network channel for communication to the LinuxONE host
- In host memory for the Linux partition

It also gets tricky to enter all the keys in hexadecimal number form in the terminal without having any errors - one mistake can lead to the loss of an encryption key, which can in turn lead to losing access to the data.

This is why IBM is offering the Trusted Key Entry workstation as a solution. It makes the master key management tasks easy and secure compared to the software-only method that is described above.

**TKE: how it works**

Trusted Key Entry is a specialized appliance that is built with custom hardware and software. The hardware is composed with x86 core-based workstation, which is equipped with a
PCI-express cryptographic co-processor. This is the same Crypto Express Adapter that is used in LinuxONE. It also has a pair of smart card readers that are attached, so that Master Keys are stored in the smart cards.

The TKE application is designed to give operators and key custodians the most secure way to manage master keys. Using six smart cards — two for Certificate Authority(CA) cards, two for module (domain) admin cards and two for split master keys — TKE gives maximum security for creating/chainging master keys and maintaining them in the most secure format.

A Linux server that runs on LinuxONE has a TKE daemon running that is called catcher.exe. This daemon allows the TKE workstation to be connected and communicate downward to the Crypto Express Adapters. Using the GUI on TKE workstation, each key custodian performs
the tasks that each person is assigned to. Figure on page 32 shows the TKE panel with the statuses of various master keys.

Because TKE does not allow a single user to change the master key or permit actions without proper procedures by key owners, this appliance requires the steps that are necessary to provide the most secure way to manage the keys.

The enterprise always must prepare service availability. With data encryption, the production servers and disaster recovery servers must maintain the same master keys, even though they might not be in the same data center. Therefore, after the master keys are created or regenerated in the production servers, they must be loaded into other servers that share/use the encrypted data, including cold-state disaster-recovery servers.

For more information on how to configure and operate TKE, see the IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/en/linuxonibm/com.ibm.linux.z.lxtc/ltke_document_outline.html
Use cases

In this chapter, we explore examples of encryption use cases that use the security features on LinuxONE. The chapter has more technical detail, to give you some understanding of what happens “under the hood” of the LinuxONE system.

- 4.1, “Containers and data encryption use case” on page 34
- 4.2, “Database and volume encryption use case” on page 41
4.1 Containers and data encryption use case

This chapter provides a description of how you might start to use LinuxONE hardware cryptographic features within a Docker environment and running on LinuxONE. The following sections describe how to benefit from CPACF hardware acceleration to secure client-server communications through an OpenSSL example:

- “Context and challenges” on page 34
- “Solution” on page 34
- “Implementation” on page 35
- “Summary” on page 40

4.1.1 Context and challenges

We are in an era where container technologies are gaining more and more popularity. The companies are increasingly adopting this virtualization technology to build and scale cloud-native applications. Undoubtedly containers provide the flexibility and rapidity to pursue business agility for speed-to-market. This fact has been understood and adopted by companies to improve their efficiency and competitiveness.

Docker is one of the most popular container platforms, and it has been adopted by companies across all industries to start the journey of cloud transformation. However, this new world of container infrastructure brings many questions about security. Again, IT decision-makers, IT architects and developers must address all the security concerns that might arise during each phase of containers lifecycle. That way, they ensure the success of the containerization strategy within the company. Some of these security concerns have already been addressed by different solutions.

If we look at image authenticity, it's possible to use a cryptographic signature mechanism: the Docker Content Trust and Notary functions. This mechanism ensures that a container image is not processed or modified by an unauthorized third party.

If we look at container isolation level, some Linux features such as namespaces, control groups, AppArmor, or SELinux can help to keep the containers isolated from each other.

But what about protecting container data? This is another security concern that is a key requirement for companies to comply with regulatory standards when they use containers. Protecting sensitive data in Docker containers can be addressed with encryption. However, this issue again brings us back to the challenges related to data encryption (performance overhead, application changes, and so on). Thus, how can LinuxONE help clients to overcome these challenges that come with the use of containers?

4.1.2 Solution

The first step toward a secure container ecosystem is to have a secure host environment with advanced security capabilities to run the entire set of containerized applications.

To help our clients with their journey toward secure hybrid cloud, LinuxONE extends its security features to containers. Thus, containers can leverage LinuxONE’s cryptographic hardware features. Encrypting data applications on containers is simplified and the encryption overhead is significantly reduced, after the containers are configured to use CPACF or cryptographic coprocessor cards.
To make your containers leverage CPACF for hardware accelerated encryption on every core, you must enable access to CPACF in the Docker host where Docker images are deployed. If the Docker host has access to CPACF, then the Docker containers that run on this host automatically can access CPACF without additional configuration.

Regarding the cryptographic coprocessor cards (Crypto Express adapters), it's the same approach. The Docker host must have access to the cryptographic cards in addition to loading the `zcrypt` device driver. Then, the container images are able to use the cryptographic cards, through the `/dev/z90crypt device node`.

In this document, we'll use the encryption of data in-flight with OpenSSL as an example to show how to enable and use CPACF function with containers. The following diagram shows the components that are part of the solution for encrypting data in flight with OpenSSL based on CPACF.

Figure 4-1 shows an example of containers that are backed by LinuxONE hardware-cryptographic acceleration with CPACF, and the components that are used for data in flight encryption with OpenSSL.

![Diagram of components used for data in flight encryption with OpenSSL](image)

**Figure 4-1  Components used for data in flight encryption with OpenSSL**

### 4.1.3 Implementation

In this section, we complete the following steps to enable a Docker container to use CPACF hardware encryption acceleration for OpenSSL.

- “Check the CPACF enablement in Docker host” on page 36
- “Install required packages in the container” on page 36
- “Configure OpenSSL” on page 39
For this demonstration, we use a CentOS Linux server release 7.6 for Docker host and container, running on a z/VM LPAR, version 6.4. The Docker image that we use is an image that supports s390x architecture. The 's390x' is used as a suffix by the image providers to specify that the Docker images are compiled for LinuxONE architecture.

**Note:** For more information about the different options to acquire Docker images for LinuxONE, see the following website: https://www.ibm.com/support/knowledgecenter/en/linuxonibm/com.ibm.linux.z.ldvd/ldvd_r_images_acquire.html

---

### Check the CPACF enablement in Docker host

CPACF can be used in your environment if the Licensed Internal Code (LIC) feature 3863 is installed in your LinuxONE. This feature is available at no extra charge. You can confirm that this cryptographic feature is enabled by going directly to the HMC console in one of the following ways:

- Follow the steps in the "Verification of installed LIC 3863 using the SE" section of the IBM Redbooks publication, *Security and Linux on z Systems*, REDP-5464. OR
- Run the following command in the Docker host:
  ```bash
  [root@openshiftmaster ~]# cat /proc/cpuinfo | grep features
  ```

  If the `features` list contains `msa`, as shown in Example 4-1, you know that the CPACF feature is enabled in the LinuxONE central processors.

*Example 4-1 Resultant features list*

| features | : esan3 zarch stfle msa ldisp eimm dfp edat etf3eh highgprs te vx vxd vxe gs sie |

---

### Install required packages in the container

Now that we have confirmed that CPACF is enabled in the Docker host, we know that it is also available automatically to the container. So now, we switch to the Docker container itself to allow OpenSSL to use CPACF hardware encryption acceleration. For this purpose, we install the following packages.

- `libica`
- `openssl`
- `openssl-ibmca`

**Install libica library**

As shown in Figure 4-1 on page 35, the `libica` library ensures communication between CPACF and OpenSSL. It contains CPACF interfaces that allow applications to use CPACF. Perform the following steps to install the `libica` library.

1. Connect to the Docker container with root user by using the following command, where `2371cdc247ee` is ID of the container image:
   ```bash
   [root@openshiftmaster ~]# docker exec -it --user root 2371cdc247ee bash
   bash-4.2#
   ```
2. Install the `libica` library by using the following command:
   ```bash
   bash-4.2# yum install libica-utils
   ```

   The results of this command are shown in Example 4-2.
Example 4-2  Transaction summary

Transaction Summary

==================================================================
Upgrade  1 Package
Total download size: 116 k
Is this ok [y/d/N]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
libica-3.3.3-3.el7.s390x.rpm | 116 kB    00:06
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating  : libica-3.3.3-3.el7.s390x                                     1/2
  Cleanup   : libica-3.2.0-3.el7.s390x                                     2/2
  Verifying : libica-3.3.3-3.el7.s390x                                     1/2
  Verifying : libica-3.2.0-3.el7.s390x                                     2/2
Updated:
libica.s390x 0:3.3.3-3.el7
Complete!

The installed **libica** library provides the **icainfo** command, which provides the list of encryption algorithms that are supported by the hardware and therefore available to the container. The command and its sample results are shown in Example 4-3.

**Example 4-3  Results of the icainfo command,**

```bash
bash-4.2# icainfo
icainfo installation command summary
  Cryptographic algorithm support

<table>
<thead>
<tr>
<th>function</th>
<th>hardware</th>
<th>software</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-1</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SHA-224</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SHA-256</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SHA-384</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SHA-512</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SHA3-224</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SHA3-256</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SHA3-384</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SHA3-512</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SHAKE-128</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SHAKE-256</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>GHASH</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>P_RNG</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>DRBG-SHA-512</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ECDH</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>ECDSA Sign</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>ECDSA Verify</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>ECKGEN</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>RSA ME</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>RSA CRT</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>DES ECB</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>DES CBC</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>DES OFB</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>DES CFB</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>DES CTR</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
```
Install OpenSSL library

OpenSSL is an open source cryptographic library that provides an implementation of the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. These protocols secure communication between two parties: client and server. Different applications rely on OpenSSL to perform the encryption requests.

Example 4-4 shows the command to install the OpenSSL library in your environment and the results after the command runs.

Example 4-4 Install OpenSSL library

bash-4.2# yum install openssl
OpenSSL installation command summary
Total 224 kB/s | 1.8 MB 00:08
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Updating : 1:openssl-libs-1.0.2k-16.el7_6.1.s390x 1/4
Installing : 1:make-3.82-23.el7.s390x 2/4
Installing : 1:openssl-1.0.2k-16.el7_6.1.s390x 3/4
Cleanup: 1:openssl-libs-1.0.2k-12.el7.s390x 4/4
Verifying: 1:make-3.82-23.el7.s390x 1/4
Verifying: 1:openssl-1.0.2k-16.el7_6.1.s390x 2/4
Verifying: 1:openssl-libs-1.0.2k-16.el7_6.1.s390x 3/4
Verifying: 1:openssl-libs-1.0.2k-12.el7.s390x 4/4
Installed:
openssl.s390x 1:1.0.2k-16.el7_6.1

Install the openssl-ibmca library

As shown in Figure 4-1 on page 35, OpenSSL needs the ibmca engine to communicate with the libica library. In this case, the encryption requests are transferred to the ibmca engine and not processed directly by OpenSSL. Then, the libica library communicates with CPACF to allow applications to start using hardware-based cryptographic acceleration.

Example 4-5 demonstrates the command to install the openssl-ibmca library and its results.

Example 4-5 Install openssl-ibmca library

bash-4.2# yum install openssl-ibmca
Installation Command Summary
Transaction Summary
Configure OpenSSL

Now that all required packages have been installed, we can explain how to prepare OpenSSL to use the \texttt{ibmca} engine. This is the last step to enable CPACF hardware acceleration of cryptographic functions in OpenSSL.

The \texttt{openssl-ibmca} package contains an \texttt{openssl.cnf} configuration file that we use to configure OpenSSL to use the \texttt{ibmca} engine. For this purpose, complete the following steps.

1. Locate the \texttt{openssl.cnf} file:
   
   ```bash
   bash-4.2# find / -name openssl.cnf /etc/pki/tls/openssl.cnf
   ```

2. Make a copy of the \texttt{openssl.cnf} file:
   
   ```bash
   bash-4.2# cp -p /etc/pki/tls/openssl.cnf /etc/pki/tls/openssl.cnf.v0
   ```

3. Locate the \texttt{openssl.cnf.sample.s390x} file. This file is in the \texttt{openssl-ibmca} package. It allows the loading of the \texttt{ibmca} engine for all the applications that have OpenSSL support.
   
   ```bash
   bash-4.2# find / -name openssl.cnf.sample.s390x
   ```

4. Add the content of the \texttt{openssl.cnf.sample.s390x} file to the \texttt{openssl.cnf} file.
   
   ```bash
   bash-4.2# tee -a /etc/pki/tls/openssl.cnf < /usr/share/doc/openssl-ibmca-1.4.1/openssl.cnf.sample.s390x
   ```

5. Insert a reference to \texttt{ibmca} engine in the \texttt{openssl.cnf} file by adding the line “\texttt{openssl\_conf = openssl\_def}” under “\texttt{RANDFILE = \$ENV::HOME/\_rnd}” line as shown in the following command:
   
   ```bash
   bash-4.2# vi /etc/pki/tls/openssl.cnf
   ```

   # Using the following parameters prevents the configuration file from hanging if \texttt{HOME} isn't defined.

   ```
   HOME = .
   RANDFILE = \$ENV::HOME/\_rnd
   openssl\_conf = openssl\_def
   ```

6. Confirm that the support of \texttt{ibmca} engine is enabled for OpenSSL. The \texttt{ibmca} engine must be listed in the result of the following command, as presented in the right column:

   ```bash
   bash-4.2# openssl engine
   (dynamic) Dynamic engine loading support
   (ibmca) Ibmca hardware engine support
   ```
Test the CPACF hardware encryption acceleration in the container

To verify that your container has been properly configured to use the CPACF hardware encryption acceleration with OpenSSL, run the following command:

```bash
bash-4.2# openssl speed sha1 -elapsed | tail -n 3
```

The results of that command are shown in Example 4-6.

**Example 4-6** Verify container is properly configured to use OpenSSL with sha1 encryption algorithm

You have chosen to measure elapsed time instead of user CPU time.
Doing sha1 for 3s on 16 size blocks: 6411136 sha1's in 3.00s
Doing sha1 for 3s on 64 size blocks: 5656337 sha1's in 3.00s
Doing sha1 for 3s on 256 size blocks: 4747020 sha1's in 3.00s
Doing sha1 for 3s on 1024 size blocks: 2938084 sha1's in 3.00s
Doing sha1 for 3s on 8192 size blocks: 612107 sha1's in 3.00s
The 'numbers' are in 1000s of bytes per second processed.

<table>
<thead>
<tr>
<th>type</th>
<th>16 bytes</th>
<th>64 bytes</th>
<th>256 bytes</th>
<th>1024 bytes</th>
<th>8192 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>sha1</td>
<td>34193.79k</td>
<td>120668.52k</td>
<td>405079.04k</td>
<td>1002866.01k</td>
<td>1671460.18k</td>
</tr>
</tbody>
</table>

To verify that the OpenSSL cryptographic calls are using CPACF, run the following command:

```bash
bash-4.2# icastats
```

Typical results of this command are shown in Example 4-7.

**Example 4-7** Results of icastats command

<table>
<thead>
<tr>
<th>function</th>
<th>hardware</th>
<th>software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ENC</td>
<td>CRYPT</td>
</tr>
<tr>
<td>SHA-1</td>
<td>34525344</td>
<td></td>
</tr>
<tr>
<td>SHA-224</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>SHA-256</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>SHA-384</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>SHA-512</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>SHA3-224</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SHA3-256</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SHA3-384</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SHA3-512</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SHAKE-128</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SHAKE-256</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Notice that the hardware column in Example 4-7 shows that the OpenSSL cryptographic calls are using CPACF.

### 4.1.4 Summary

In addition to the advantages related to scalability and performance, LinuxONE also provides your containerized applications with a secure host environment. By running your containers on LinuxONE, you can benefit from its hardware security features to secure and accelerate the encryption of your data. One of the main advantages of using these hardware security features is being able to encrypt your data without modifying your applications and with minimal performance overhead.
Using OpenSSL for data in-flight encryption is one of the various examples of what you can implement on LinuxONE. You can also use other Linux libraries, such as dm-crypt, to encrypt your data at-rest while you continue to benefit from the hardware cryptographic acceleration with CPACF or cryptographic cards.

With its differentiating security capabilities, LinuxONE is a great starting point in your transition to cloud-native applications. LinuxONE can also help you meet compliance requirements for protecting sensitive data in a container ecosystem.

### 4.2 Database and volume encryption use case

This section shows how you can use LinuxONE hardware cryptographic acceleration capabilities features to address the challenges that are related to protection of sensitive data that is stored in databases. This approach is illustrated through a database volume encryption use case with dm-crypt LUKS- and CPACF-protected keys to protect data at rest.

#### 4.2.1 Context and challenges

Databases contain different types of sensitive information, as in these examples:

- PII (Personally Identifiable Information), related to data that could help to identify a specific individual,
- Business Information that includes data that represents a risk to the company if disclosed to competitors or the general public.
- Classified Information that refers to top-secret government data.

All these types of sensitive data must be protected, especially in the light of data breaches that are affecting more and more businesses. Different regulations and laws exist to ensure that this sensitive data remains in good hands. Otherwise, companies could incur heavy financial repercussions.

All these factors make the protection of data that is stored in databases more urgent and important in our century. It becomes one of the highest priorities for companies around the world.

This context brings back a significant focus on data-at-rest encryption. This focus addresses some of the data security issues and complies with various regulations, such as Payment Card Industry Data Security Standard (PCI DSS) or the Health Insurance Portability and Accountability Act (HIPAA).

Consider the following scenario. Your company has a database that stores sensitive information. To meet the data protection requirements, you must encrypt all data in flight and at rest. How can you handle this without altering your applications’ response times, increasing your costs, or suffering from lack of skills in your organization?

For data in flight, these issues can be easily addressed by encrypting your data based on SSL encryption by using the OpenSSL Linux library, for example. As explained in the previous section, on LinuxONE, OpenSSL can benefit from the hardware cryptographic acceleration with CPACF or cryptographic cards. Thus, you can transparently encrypt connection by the database server, in addition to securing the Linux sessions through SSH, for example.

Regarding the protection of data at rest, one of the most common use cases is the use of Filesystem-level encryption, which encrypts data before it is written out to physical volumes.
However, this option can expose you to the same issues that were mentioned previously (costs, performance impact, and so on). Thus, how can LinuxONE help you to minimize the impact of encrypting your data at rest?

4.2.2 Solution

Encryption at the file-system level can be implemented by using `dm-crypt` LUKS, which is an encryption subsystem included in the Linux Kernel. With this transparent encryption mechanism, you can encrypt disks, software RAID volumes, partitions, and logical volumes to protect your data at rest. It allows the encryption of all data that is written to disk and decrypts all data that is read from disk. The data appears only in the clear in the application.

**Note: dm-crypt** provides plain format volumes or LUKS (Linux Unified Key Setup) volumes. In plain mode, **dm-crypt** doesn't add any headers or metadata to the volumes. The **dm-crypt** LUKS adds a metadata header to the encrypted volume data, and therefore offers more features than plain mode.

LUKS2 format is the preferred option for LinuxONE data at-rest encryption. For more information, see Section 3.9.3, “Volume format considerations” in *Getting Started with Linux on Z Encryption for Data At-Rest:* [http://www.redbooks.ibm.com/abstracts/sg248436.html?Open](http://www.redbooks.ibm.com/abstracts/sg248436.html?Open)

By implementing this encryption option on other platforms, you might suffer from increased performance overhead, which is always involved when you use software-encryption mode. On LinuxONE, **dm-crypt** LUKS can leverage the CPACF on-chip encryption co-processor to remove this overhead.

In addition to this, there is a unique security enhancement that LinuxONE provides to **dm-crypt**, which consists of using the CPACF protected keys support to resolve the security issues that are related to storing keys. Indeed, in case **dm-crypt** is used with clear key, if the memory system is dumped, then keys are readable. With a CPACF protected key, there is no encryption key that is stored in clear in the Operating System. But how does that work?

The **dm-crypt** cryptographic operations are performed by using CPACF. CPACF code generates the wrapping keys. These keys are unique to each LPAR, and they are stored in the Hardware System Area (HSA). HSA is a reserved memory area that is separated from client-purchased memory that is used for internal system functions. And HSA is accessible only through the firmware.

As shown in Figure 4-2 on page 43, at cryptographic card level, the secure key is encrypted by the master key. This secure key is used as the source key of the protected key. It is decrypted and sent to CPACF in clear text. Then at CPACF level, the key is wrapped by the CPACF wrapping key, and the protected value is stored in memory. As a result, protected keys can never be seen in plain text by the operating system or by applications. With each encryption or decryption operation, the protected keys are sent to CPACF to be unwrapped by the CPACF wrapping keys.

Figure 4-2 on page 43 gives an overview of data-at-rest encryption that uses **dm-crypt** on LinuxONE. It also shows that **dm-crypt** is located between the file system and the physical disks, which allows data encryption to happen when they are written to physical disks.

This diagram also shows the key wrapping process that is used to provide **dm-crypt** with CPACF protected keys.
4.2.3 Getting started

The following steps refer to the main steps that compose the data at-rest encryption process that uses dm-crypt LUKS with CPACF protected keys.

1. Check CPACF enablement in your system (Licensed Internal Code (LIC) feature 3863 must be installed).
2. Check the availability of Crypto Express adapters (use CCA mode), and add the domains that are needed to hold the master key and secure key. (You can do this through the HMC console.)
3. Load the master key into the Crypto Express adapter domain.
4. Install the pkey kernel module for protected key management.
5. Use the zkey utility to generate and manage secure key.
6. Set up dm-crypt as usual (install the cryptsetup package).
7. Create the encrypted volumes and set up the dm-crypt LUKS2 header in the volumes.

For more information about these steps, see Getting Started with Linux on Z Encryption for Data At-Rest: http://www.redbooks.ibm.com/abstracts/sg248436.html?Open

4.2.4 Summary

As an enterprise data-serving platform, LinuxONE provides a secure environment to protect the sensitive data stored in your databases. Protecting data at rest is one of the important components in the Data Security journey. To help clients with the different challenges that are related to the implementation of data-at-rest encryption, LinuxONE offers hardware cryptographic acceleration capabilities with CPACF.
Encryption of database volumes with **dm-crypt** LUKS is one of the solutions that can be implemented on LinuxONE to protect your data at rest. That type of protection is one of the most common use cases. Because this mechanism works at the kernel level, it can improve security in two fundamental areas:

- **Performance:** You avoid the overhead that is associated with traditional encryption.
- **Encryption:** You better protect the encryption keys against eventual attacks by using the protected keys that CPACF provides.
IBM Blockchain Platform with IBM LinuxONE

The IBM Blockchain Platform (IBP) provides an easy way to create and deploy Hyperledger Fabric servers either in the public cloud or in an on-premises data center.

In this chapter, we show how an enterprise client can deploy IBM Blockchain Platform and explain the security benefits of building a blockchain on LinuxONE.

- 5.1, “Blockchain, Hyperledger, and IBM Blockchain Platform” on page 46
- 5.2, “Details of IBM Blockchain Platform for LinuxONE” on page 47
5.1 Blockchain, Hyperledger, and IBM Blockchain Platform

Well known from its use for crypto-currencies such as Bitcoin, blockchain technology is a hot topic in the technology industry. The core components of blockchain technology are not really new. However, as computing power grew alongside the expansion of IoT and mobile devices, adopting blockchain technology into applications has become a feasible reality. Blockchain can provide a way for enterprises to share data among one another securely, but also apply permissions on who can create and view that data.

Blockchain technology provides a way to securely store data, most of the time as a metadata to point to more detailed database tables. The data that is stored in blockchain is distributed among participants' servers, also called 'decentralized' data stores.

Data that is stored in blockchain is also inherently encrypted with very strong cryptographic algorithms and key management procedures. When a blockchain stores data with these cryptographic procedures, it stores them with time stamps, sequence information that tells the order of the data blocks, and hash information that is derived from a prior data block. As a result, it can prove its position in the blockchain based on its parent block.

This strong cryptographic process makes blockchain 'immutable', which means that once data is stored, it stays in its original form and is unable to be modified. Unlike other commonly used databases, blockchain normally does not allow inserts or modification of its data. This is why blockchain is such a strong candidate for verified ledger applications between many distinct parties.

Large enterprises in industries such as banking, insurance, supply chain/logistics, and healthcare are thinking about how they can adopt blockchain technology. IBM started working with these customers on requirements for blockchain technology, but discovered that public network blockchain protocols, such as Bitcoin and Ethereum, were not a good fit for enterprise use cases.

IBM researchers and developers worked to create a foundation of enterprise-grade private blockchain protocol and donated the initial work as an open source software to the Linux Foundation, which became the Hyperledger Fabric. Even though it is still possible to run other types of open source blockchain protocols in LinuxONE, there is no formal support, thus we do not discuss it in this book.

IBM created IBM Blockchain Platform, an easy way to create and deploy Hyperledger Fabric servers either in public cloud, or in an on-premises data center.

As mentioned above, Hyperledger blockchain was born from the needs of the enterprise. It is a permission-based private blockchain network. That means that only allowed participants can see the data and access the data from the shared, distributed ledger (database).

Without going into too much detail, we describe the basic concept of blockchain data here.

Figure 5-1 on page 47 shows how each block of data on a blockchain network is stored in each peer node’s database. A peer node is served as data processing host for the blockchain network. It validates/Endorses a transaction that is requested from an application. Then, it ensures that the rules (business rules) that are described in chain code get checked when the transaction occurs.

The peer also communicates with the orderer unit, where it keeps the sequence of transactions and distributes to proper peer nodes when a transaction is requested.
There is also a certificate authority, which manages certificates for each member’s entity to verify that it is genuine and to validate authenticity.

On a peer node, the blockchain always starts with a genesis block, which is an initial data block that has a specially encoded Hash value. Its hash value is also transferred to the next block in sequence as an input. The next block generates a new hash value from the prior block’s hash and the payload of the new block’s data, then repeats for the next blocks.

Keeping the prior block’s hash and making a new hash from it makes blockchain data unable to be reversed or modified. When the contents of the data payload (world state) changes, blockchain changes the hash for the next block that already was created and impacts all the world states behind the block that was modified. This is the key idea of blockchain data being immutable and it requires a high-speed hash process to generate new blocks.

Hyperledger also uses an asymmetric cryptographic function to create certificate and validate the signature of the transactions. The workflow of those asymmetric keys can be a bit complicated in Hyperledger Fabric, because it involves multiple members and different channels for various application uses. If a person or an organization plans to develop Hyperledger applications, they must plan carefully where to host the service, because it can consume a good amount of resources depending on transaction sizes. IBM LinuxONE is a great choice for a large enterprise Hyperledger Fabric service platform, because it has the best cryptographic hardware in the industry. Also, it has a feature that is called On/Off Capacity on-demand. You use it to turn on extra CPU capacity to boost processing power in a few seconds for any need that is caused by spiking workloads.

For more information about Hyperledger blockchain network, see the official Hyperledger documentation:

5.2 Details of IBM Blockchain Platform for LinuxONE

To run a Hyperledger application for your enterprise, or even for personal projects, you need to have at least one Hyperledger Fabric Network running somewhere. It is an architecture form that consists of multiple servers and services. They require many different access controls, permissions, and business rules. In the Hyperledger world, these business rules are called Smart Contracts, and it is represented in the form of Chain Code (“chain” as in “blockchain”).

To familiarize yourself, you can choose to download the open source version of Hyperledger codes to your personal computer or to your choice of servers. If you are an enterprise blockchain application developer/architect, you need at least have one Hyperledger Fabric Network running across your team to develop and test more extensively.

When the actual business applications begin to interact with other business entities, the Hyperledger Fabric must serve those other entities over the internet and communicate to
handle the chaincode’s requests. This is often not a simple task. Even though you are running multiple servers, they must always be up and redundant to keep the data replicated and consistent among participating members of the blockchain network.

For this reason, IBM Cloud hosts Hyperledger Fabric services to enterprise customers with 24x7x365 support, as a solution called IBM Blockchain Platform (IBP). You can choose to start coding from IBP in IBM Cloud, but you also can download container images for IBP and run them on your on-premises server.

5.2.1 IBM Blockchain Platform

The IBM Blockchain Platform (IBP) is a Hyperledger Fabric blockchain protocol as-a-service that has following properties. (Also see https://www.ibm.com/downloads/cas/Q90GBLV7.)

Modularity
Blockchain networks must be able to incorporate a wide range of new and existing “pluggable” features depending on the enterprise and industry. As a result, Hyperledger Fabric was developed to be modular to support networks, as new features emerge. Modularity in Hyperledger Fabric allows the IBM Blockchain Platform to leverage industry-leading security practices to serve production-ready networks, including GDPR and HIPAA best practices.

Scalability
Organizations across industry sectors demand solutions that scale as they move past initial explorations and proof-of-concepts.

Hyperledger Fabric was built to support growing business networks, which need to dynamically add participants and support increasing amounts of transaction processing. Many aspects of scalability depend on network configuration of consensus, membership, and security. The IBM Blockchain Platform leverages Hyperledger Fabric to provide a modular platform that supports the ability to configure a network to support the throughput numbers and network growth that are required.

Consensus
An important feature in the security, scalability, and maturity of any blockchain framework is a clearly-defined and implemented consensus protocol. As mentioned above, consensus in Hyperledger Fabric is pluggable and fit specific enterprise use cases.

Therefore, Fabric allows you to choose the best consensus protocol to fit your specific business networks’ needs. Hyperledger Fabric’s success to date is driven by the massive amount of community support it has received through Hyperledger. Open governance of the code base with a clear purpose has allowed it to emerge as the industry-leading protocol and framework for enterprise production networks.

While IBM Blockchain Platform is offered in IBM Cloud and in third-party public cloud services, it is often beneficial for an enterprise to have an on-premises server for many reasons1.

Running the IBM Blockchain Platform components outside of IBM Cloud provides you with more flexibility to grow or join a blockchain network. It helps network initiators grow their networks by allowing new members to join while using the platform of their choice. It allows

organizations that are interested in joining blockchain networks to colocate their peers with their existing applications or to integrate with their systems of record.

Users of this offering manage their own security and infrastructure. IBM Cloud doesn't provide those services.

**IBM Cloud Private**

IBM Cloud Private (ICP)\(^2\) is a reliable and scalable cloud platform that runs on your on-premises infrastructure. It’s built on open source frameworks, like containers, Kubernetes, and Cloud Foundry. In addition, it offers common services for self-service deployment, monitoring, logging, and security, and a portfolio of middleware, data, and analytics.

ICP on LinuxONE brings all the benefits of reliability and scalability of IBM Cloud Private but also provides excellent security without any changes in the application nor containers. It also can scale vertically and horizontally, which helps process surges in workloads due to any business environment changes (for example, a stock market surge due to a catastrophic event).

You can find more about IBM Cloud Private for LinuxONE at the following website: https://www.ibm.com/us-en/marketplace/cloud-private-on-z-and-linuxone

**Architecture overview of IBP for ICP**

Figure 5-2 shows a simple use case for IBM Blockchain Platform in IBM Cloud Private.

\(^2\) https://www.ibm.com/cloud/private
5.2.2 How IBP for ICP exploits security hardware features on LinuxONE

**CPACF and Crypto Express**

IBM Blockchain Platform for IBM Cloud Private can use the cryptographic hardware that is built into LinuxONE. Starting with CPACF (CP Assisted Cryptographic Functions), when new blockchain data gets generated it uses a HASH function called (SHA-256) in CPACF, and then encrypts the data block in the file system with an AES block cipher also in CPACF. IBP automatically uses these hardware features when used with IBM Cloud Private for LinuxONE.

LinuxONE can also be equipped with the Crypto Express co-processor feature, which is a hardware security module (HSM) that is plugged into the PCI slot on the server. It meets the highest level of FIPS 140-2 standard, a level-4.

When equipped with Crypto Express co-processor, IBP can use protected key encryption for file systems. In this case, the encryption key for the block ciphers is always protected with another 'key-encrypting-key' in the HSM. This function dramatically reduces the chance of exposing the encryption keys, because it is protected by an HSM in the IBM Cloud and on IBM LinuxONE. In contrast, most of the other public cloud solutions store keys in memory as clear text.

The Crypto Express co-processor also helps accelerate blockchain processing times. It contains custom-built ASICs that accelerate asymmetric cryptographic functions such as RSA or Elliptic Curve Cryptography (ECC). When encryption and authentication are required for blockchain processes, Crypto Express hardware can handle them much faster, compared to just using CPU cycles for the computations.

Hyperledger Fabric uses ECDSA (Elliptic Curve Digital Signature Authentication) for certificate authority enrollment and for encrypting blocks of data. Thus, when you have ECDSA acceleration through an ECC unit in CPACF (**Note: LinuxONE III only**), you can process more data in a shorter time. Its TCP/IP communications are also SSL/TLS enabled. So, having RSA digital signature generation and verification helps to serve SSL/TLS traffic better with a Crypto Express co-processor.

**Currently supported features of IBP for ICP on LinuxONE**

At the time of writing this Redpaper, IBP for ICP supports the SOLO Ordering Service, Local Remote Peers, and certificate authority. The vision is to support the full Hyperledger Fabric functions. This approach enables organizations to start building Hyperledger applications and test the connections to a main Hyperledger Fabric that is shared with other organizations. It offers easy setup and integration to existing internal enterprise data stores. That way, you don’t have to worry about exposing corporate data sets to an external cloud and about maintaining full control over the entire lifecycle of a blockchain network.

It can be deployed as a containerized service, working with selected Kubernetes management suites including IBM Cloud Pak™. For more information about the offerings on IBM Blockchain Platform for LinuxONE, see this page: [https://ibm.biz/BdYaYi](https://ibm.biz/BdYaYi)
The section presents some of the common cryptographic hardware libraries, tools, and drivers that can be used on LinuxONE.

**Reference table**

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>libcsulcca</td>
<td>The libcsulcca library provides APIs for CCA and secure key cryptography functions that are provided by cryptographic express coprocessor.</td>
</tr>
<tr>
<td>libica</td>
<td>The libica library provides hardware support for cryptographic functions. It is part of the openCryptoki project in GitHub. It is primarily used by OpenSSL through the IBM OpenSSL CA engine or by openCryptoki through the ICA token. A higher level of security can be achieved by using it through the PKCS #11 API implemented by openCryptoki.</td>
</tr>
<tr>
<td>openCryptoki</td>
<td>openCryptoki is an open source implementation of the Cryptoki API defined by the PKCS #11 Cryptographic Token Interface Standard. Thus, openCryptoki provides support for several cryptographic algorithms according to the industry-wide PKCS #11 standards. The openCryptoki library loads the so-called tokens that provide hardware- or software-specific support for cryptographic functions.</td>
</tr>
<tr>
<td>z90crypt</td>
<td>z90crypt is a cryptographic device driver. It acts as the interface to the PCI cryptographic card coprocessor. This driver must be loaded to use CEX features.</td>
</tr>
<tr>
<td>virsh</td>
<td>You can create, delete, run, stop, and manage your virtual machines from the command line by using a tool called virsh. Virsh is particularly useful for advanced Linux administrators, interested in script or automating some aspects of managing their virtual machines.</td>
</tr>
<tr>
<td>(1) Cryptographic Hardware Libraries and Drivers</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>qemu</td>
<td></td>
</tr>
<tr>
<td>Qemu is a machine emulator that can run operating systems and programs for one machine on a different machine. Mostly it is not used as emulator but as virtualizer in collaboration with KVM kernel components. In that case, it uses the virtualization technology of the hardware to virtualize guests.</td>
<td></td>
</tr>
<tr>
<td>libvirt</td>
<td></td>
</tr>
<tr>
<td>While qemu has a command-line interface and a monitor to interact with running guests, it is rarely used that way for other means than development purposes. Libvirt provides an abstraction from specific versions and hypervisors and encapsulates some workarounds and best practices. Libvirt is an open source API, daemon, and management tool for managing platform virtualization. It can be used to manage KVM, QEMU, and other virtualization technologies. These APIs are widely used in the orchestration layer of hypervisors in the development of a cloud-based solution.</td>
<td></td>
</tr>
<tr>
<td>vfio</td>
<td></td>
</tr>
<tr>
<td>The VFIO driver is an IOMMU/device-agnostic framework for exposing direct device access to userspace, in a secure, IOMMU protected environment. In other words, this allows safe, non-privileged, userspace drivers.</td>
<td></td>
</tr>
</tbody>
</table>

### Linux Tools

| chzcrypt                                      |
| The chzcrypt command is used to configure cryptographic adapters that are managed by the cryptographic device driver and modify the AP bus attributes. |

| lszcrypt                                      |
| The lszcrypt command is used to display information about cryptographic adapters that are managed by the cryptographic device driver and its AP bus attributes. |

| icainfo                                       |
| The icainfo command is used to find out which libica functions are available on your Linux system. |

| icastats                                       |
| The icastats command is used to find out whether libica uses hardware acceleration features or works with software fallbacks. icastats collects the statistical data per user and not per system. |

| zcryptctl(KVM)                                 |
| The zcryptctl command is used to control access to AP queues and functions. |

### z/VM

| QUERY CRYPTO DOMAINS USERS                    |
| These command queries can be used to show the status of the cryptographic hardware. |
| When the DOMAINS operand is specified, the status of the installed AP domains is displayed. When the USERS operand is specified after the DOMAINS operand, the users who are authorized for CRYPTO APVIRT in the directory are listed. |