Confidential Mode for Hyperdisk - DEK Protection Analysis

Google

Version 1.0 – May 8, 2024

Prepared By
Catalin Visinescu

Prepared For
Jo Hellwig
Vladimir Yuzhikov

1 Executive Summary

Synopsis

During the spring of 2024, Google engaged NCC Group to conduct a design review of Confidential Mode for Hyperdisk (CHD)[1] architecture in order to analyze how the Data Encryption Key (DEK) that encrypts data-at-rest is protected. The project was 10 person-days and the goal is to validate that the following two properties are enforced:

- The DEK is not available in an unencrypted form in CHD infrastructure.
- It is not possible to persist and/or extract an unencrypted DEK from the secure hardware-protected enclaves.

The two secure hardware-backed enclaves where the DEK is allowed to exist in plaintext are:

- Key Management System HSM - during CHD creation (DEK is generated and exported wrapped) and DEK Installation (DEK is imported and unwrapped)
- Infrastructure Node AMD SEV-ES Secure Enclave - during CHD access to storage node (DEK is used to process the data read/write operations)

Scope

NCC Group evaluated Confidential Mode for Hyperdisk - specifically, the secure handling of Data Encryption Keys across all disk operations including:

- disk provisioning
- mounting
- data read/write operations

This assessment was based on architecture documentation; source code review was not performed. While reviewed by NCC Group in depth, some details were excluded from this report at Google's request to protect their intellectual property.

Findings

NCC Group found that CHD was designed from the start with security in mind. Using hardware enclaves for the sensitive operations is a sound choice as they are carefully designed to protect the confidential material and go through rigorous validation processes.

An analysis of cryptography used in CHD was also performed. NCC Group was not able to find weaknesses that could compromise the DEK. For both symmetric and asymmetric cryptographic operations, only strong cryptographic primitives are used. For instance:

- The DEK is stored encrypted in Disk Metadata in the Control Plane and needs to be loaded to the HSM. The key used to wrap DEK is generated in the HSM and never leaves the HSM.
- When DEK is in transit between the HSM and SEV-ES enclaves, strong ECC P384 ephemeral public/private key pairs are used to ensure the DEK is not exposed.
- The DEK is used with AES-GCM-256 to encrypt data in persistent disk.
- The DEK only persists in SEV-ES until the disk operations are complete, the DEK is removed when the session is closed.

NCC Group has verified that Google complies with the requirements of the two properties stated above. No issues were identified with the respect to the handling of the DEK.

1. https://cloud.google.com/compute/docs/disks/hyperdisks

2 Overview

Introduction

Hyperdisk is Google's newest generation of network block storage service in Google Cloud Platform (GCP). Designed for the most demanding mission-critical applications, Hyperdisk[2] offers a scalable, high-performance storage service with a comprehensive suite of data persistence and management capabilities. Present in many regions of the world, it replaces the older Persisted Disk solution. Google also offers *Confidential Mode for Hyperdisk (CHD)*, which is the subject of this report, for added protections of encryption keys by processing the data with keys only available in secure hardware enclaves. Therefore, when this mode is selected, the encryption keys:

- will not reside on the VM host machine, hypervisor, or VM instance,
- are separated from the general purpose compute,
- are never present in plaintext in shared infrastructure.

Secure Enclaves

To secure data, the CHD solution uses data encryption keys that are only present in hardware-backed storage. These keys are used to protect customer's data-at-rest. This section briefly describes the two enclaves where the data encryption keys can be found in plaintext, and discusses why their use is a sound choice. References are provided herein for more in-depth discussion.

Marvell HSM

Marvell LiquidSecurity is a cloud-optimized Hardware Secure Module (HSM) used in Google's Cloud Key Management System[3] HSMs.[4] Connected to the host's PCIe slot, the HSM is certified for FIPS 140-2[5], Common Criteria, eIDAS and PCI-PTS compliance.[6] It is a mature solution that implements security-relevant features expected for a modern HSM, a few listed here:

- It implements role-based access control (RBAC). The RBAC allows HSM-related tasks to be split, with minimal responsibilities given that allows the engineers to perform the required work. Credentials are required to perform any changes, and some of these roles are:
  - partition crypto officer (PCO) - user can perform management operations such as creating and deleting users
  - crypto user (CU) - user can perform key management and cryptographic operations such as key creation/deletion, import/export, as well as use using the keys for encryption, signing, etc.
  - appliance user (AU) - user is allowed to perform cloning operations and retrieve audit logs.
- Only signed firmware updates are allowed.
- Sensitive keys are stored (data-at rest) and operated in tamper-resistant hardware.
- Data-in-motion is encrypted.

As discussed in the next section, the HSM security is crucial in protecting the data encryption key.

2. https://cloud.google.com/compute/docs/disks/hyperdisks
3. https://cloud.google.com/kms/docs/key-management-service
4. https://cloud.google.com/docs/security/cloud-hsm-architecture
5. https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp3521.pdf
6. https://www.marvell.com/products/security-solutions.html

AMD SEV-ES

The platform security processor (PSP) is an isolated ARM processor that runs independently from the main X86 core complex, it provides support for Secure Encrypted Virtualization (SEV).[7] The PSP has a built-in cryptographic coprocessor that is used to generate and protect the guest's memory encryption keys and to accelerate cryptographic operations.

SEV enables running encrypted VMs in which where the content of the memory (data and code) is only available to the virtual machine. Unprotected data from the CPU arrives at the memory controller (MC) and it is encrypted before it is written to memory. When accessing the memory, the opposite happens, and the CPU receives decrypted data that it can use. Not all memory can be encrypted, for instance shared memory that is used when communicating with peripherals through I/O operations does not go through the encryption process. The VMs marks most pages as private by setting the most significant bit (enCrypted bit or C-bit) to 1. A small number of pages that are used for outside communication are not marked as private, that is C-bit is 0.

A high-level functional diagram is shown below:[8]

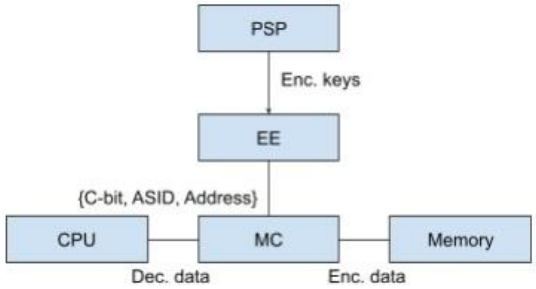

Figure 1: Overview of Memory Encryption Process with SEV

In May of 2022 the Google Project Zero group performed a code-assisted security audit[9] of the SEV firmware (Milan platform). The project extended over few months and all the issues found were quickly addressed by AMD.

In August of 2023 NCC Group also reviewed the Milan platform, in a 90-person day effort. Similarly, this audit was code-assisted, and identified issues were addressed by AMD. While the review had multiple goals, one that is directly relevant in this context was to evaluate threats that could undermine SEV/SEV-ES memory/register encryption or gain the ability to reveal the plaintext contents of data pages belonging to a VM.

AMD has announced[10] that it is publishing the source code for the SEV technology,[11] demonstrating their transparency and allowing for the technology to face additional scrutiny.

As discussed in the next section, the SEV-ES security is essential in ensuring the data encryption key is protected while processing data from and to storage.

7. https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/solution-briefs/amd-secure-encrypted-virtualization-solution-brief.pdf
8. https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/AMD_GPZ-Technical_Report_FINAL_05_2022.pdf
9. https://googleprojectzero.blogspot.com/2022/05/release-of-technical-report-into-amd.html
10. https://ir.amd.com/news-events/press-releases/detail/1154/amd-shares-the-technical-details-of-technology-powering
11. https://github.com/amd/AMD-ASPFW

3 Cloud Hyperdisk Architecture and Analysis of DEK Protection Mechanism

Goal

The Data Encryption Key (DEK) is used to protect customers' data, therefore it is crucial to ensure this key is safely protected and never exposed. The main goal of this document is to give a brief overview of the Confidential Mode for Hyperdisk architecture and to show that the solution conforms to the following statements:

- The DEK is not available in an unencrypted form in CHD infrastructure.
- No capability to persist/extract unencrypted DEK from the enclave

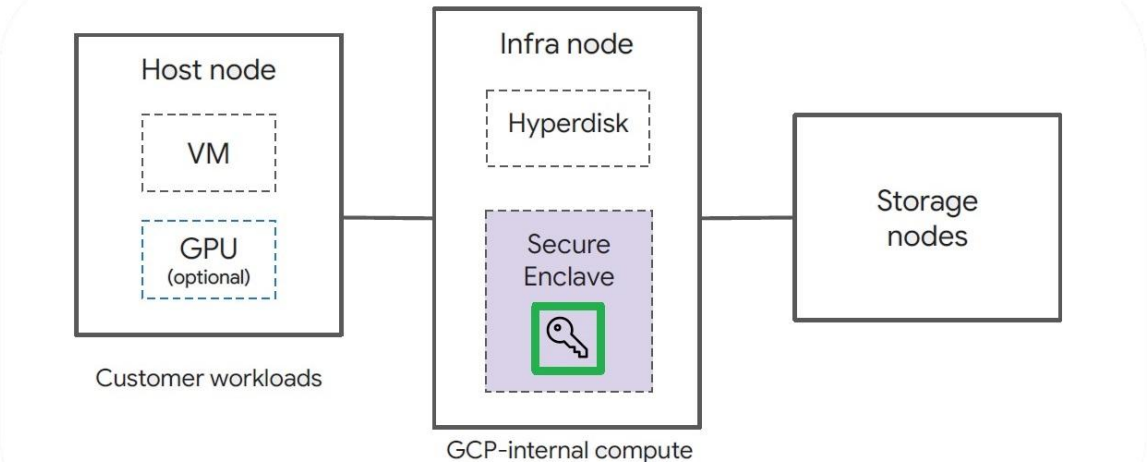

Figure 2: Location of Data Encryption Key Used by CHD

The assessment was performed based on internal Google documents. Some details were excluded from this report at Google's request to protect their intellectual property.

Key Use Scenarios

There are three situations that require the presence of the DEK in plaintext, each described below. While not in scope for this assessment, it must be noted that communication between various CHD components such as Cloud KMS and Infrastructure Nodes running the Hyperdisk process is secured using standard Google controls to authorize jobs, end-to-end binary provenance and authorization, and PSP transport encryption.[12]

Disk Creation

Before initiating a disk creation operation, the user must create a Customer-managed Encryption Key (CMEK).[13] The customer key is managed by the Cloud KMS which uses the Cloud HSM service to manage hardware-backed keys, and it therefore relies on HSMs' security properties for protection. The CMEK, which is a symmetric key, is generated in an HSM using the internal RNG engine, and it is never exposed outside the secure enclave. The CMEK scope may be regional or multi-regional and could be replicated across HSMs. While Cloud KMS-authenticated users can utilize the CMEK, they lack the ability to see what it is.

When a disk creation operation is requested, the Control Plane requests Cloud KMS to generate a DEK, an AES-256 key, and return it encrypted with the CMEK. Again, the Cloud KMS relies on an HSM to generate the DEK, using the RNG engine present on the HSM. The Control Plane stores the wrapped DEK into the Disk Metadata storage area and an entry is added to the Audit Logs signaling that the disk was created. Note that the HSM does not

12. https://cloud.google.com/blog/products/identity-security/announcing-psp-security-protocol-is-now-open-source
13. https://cloud.google.com/kms/docs/cmek

manage the DEK and will no longer store it after the generation step; the only way to get it back to the HSM is by loading the wrapped DEK.

Compliance:

- The HSMs use a multitude of sources to continuously collect entropy from the system, and the entropy generation engine goes through strict audits to show they conform to applicable security standards.[14] This ensures that it is impractical to guess CMEK and DEK, which are the foundation of CHD security.
- The DEK is only available outside the HSM wrapped with CMEK which, in turn, is only available in the HSM. This satisfies the first property.
- HSMs go through rigorous testing to ensure that keys are protected against attacks, physical and remote. This satisfies the second property.
- A vulnerability that exposes the Control Plane's Disk Metadata content will only allow an attacker to access the CMEK-wrapped DEK without the possibility to decrypt the DEK.

Disk Mounting

Before accessing the stored content, the disks must be mounted. For CHD, the Hyperdisk process running on the Infrastructure Node must be capable of encrypting data received from users' VMs and decrypting data received from disk. To accomplish this, the Hyperdisk process makes use of the local SEV-ES enclave to perform the cryptographic operations. The SEV-ES enclave does not have persistent storage, so the enclave must first be provided the DEK. This key provisioning operation occurs in few steps described below.

When the user makes a request to the Control Plane to mount a disk, the wrapped-DEK is read from Disk Metadata storage area and loaded into Cloud KMS. As part of the same transaction, the Hyperdisk process requests an ECC P384 public/private key pair to be generated in local SEV-ES, and the public part is exported outside the enclave and sent to the Cloud KMS service. The Cloud KMS returns the DEK wrapped with the public key to the Hyperdisk process which decrypts the DEK in the SEV-ES enclave. Once no longer required, the public/private key pair is deleted. Entries are added to the Audit Logs on completion of mounting and unmounting operations.

Compliance:

- During the DEK installation into the KMS HSM, the CMEK-wrapped DEK travels from Disk Metadata in the Control plane to the HSM in KMS. Only in Cloud KMS's HSM it is possible to retrieve the DEK, therefore the first property is met.
- The private key used to unwrap the DEK in the SEV-ES enclave is generated in and never leaves the SEV-ES enclave. The DEK is securely transferred from the HSM to the SEV-ES enclave on Hyperdisk. This satisfies the first property.
- The SEV-ES firmware has been thoroughly reviewed to ensure it generates strong random keys.
- Both the Marvell HSM and the AMD PSP platform that enables SEV-ES have been designed to protect against extraction of cryptographic keys. This satisfies the second property.

Read/Write Access to Disk

At this point the Hyperdisk is ready to process data going to and coming from the user's VM instance. The Hyperdisk process running on the Infrastructure Node is able to use the key stored in the SEV-ES enclave (see Figure 2) without being able to view it. Before writing it to disk, the Hyperdisk first passes the data to the enclave to be encrypted with DEK. Similarly,

14. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf

before providing it to the VM, the Hyperdisk sends the data to the enclave for decryption. For these cryptographic operations, AES-GCM-256 is used.

Compliance:

- During the read and write session, the DEK is never exposed and only used inside the secure enclave. This satisfies the first property.
- The DEK is only available in the secure enclave during the data access session, to reduce the likelihood of an incident. Additionally, there is no SEV-ES key export capability, which ensures the second property is met.
- Companies' hard drives sometimes end up on online marketplaces.[15] Since it is encrypted with the DEK, the users' data is protected even in case an improbable event as such occurs. It needs to be mentioned, however, that Google already has a process for removing used hard drives from data centers, and it includes either shredding or disk re-encryption.
- While Infrastructure Nodes and storage disks are owned by Google, users can rest assured that only they have access to their data since the SEV-ES enclave was designed especially for this use case.

15. https://www.computerworld.com/article/1689056/idaho-utility-hard-drives-and-data-turn-up-on-ebay.html

4 Contact Info

The team from NCC Group has the following primary members:

- Catalin Visinescu – Consultant
  catalin.visinescu@nccgroup.com
- Gage Polonsky – Project Manager
  gage.polonsky@nccgroup.com
- Nate Russo – Account Manager
  nate.russo@nccgroup.com

The team from Google has the following primary members:

- Jo Hellwig
  johellwig@google.com
- Vladimir Yuzhikov
  yuvladimir@google.com