GDPR Compliant Cloud Data Storage and Processing Engine as a Service

Cryptographic Protocols

Mats Kockmeyer
Mats.Kockmeyer2@stud.hs-flensburg.de

Felix Janke
Felix.Janke2@stud.hs-flensburg.de

Hochschule Flensburg — 28. Januar 2020

1 Projekt

Das Ziel des Projektes ist es, eine Möglichkeit zur GDPR Konformen Verarbeitung von personenbezogenen Daten in der Cloud zu finden. Nach Möglichkeiten sollen die personenbezogenen Daten anonymisiert außerhalb der Europäischen Union gespeichert und verarbeitet werden. Sind die Daten anonymisiert gespeichert, muss mit dem entsprechendem Cloud-Anbieter kein Datenverarbeitungsauftrag ausgehandelt werden, da dies nicht unter die Anwendungsbereich (Artike 2, weiter Artikel 4) der GDPR fällt.

2 Firmen Definition

Der Service soll Exemplarisch anhand einer Firma erklärt und durchgeführt werden. Zuerst muss die Firma dafür definiert werden.

Branche	KFZ Gewerbe
Land	Deutschland
Standorte	1
Betriebsgröße	120 Mitarbeiter
Kundenstamm	50.000
Datenbankgröße	100 GiB
Infrastruktur	Lokale
Internetanbindung	100 MBits Down & 40 MBits Up

Zum aktuellen Zeitpunkt werden die personenbezogenen Daten in der Firma, in einer getrennten Räumlichkeit, auf einem lokalen Server gespeichert. Um Kosten und eventuelle Ausfallzeiten zu minimieren, sollen die Daten zu einem Cloud Anbieter migriert werden. Aufgrund der hohen kosten deutscher Cloud Anbieter, würde Firma eine Speicherung in einer Ausländischen Cloud bevorzugen, zum Beispiel in Russland.

3 Anforderungen

In diesem Kapitel werden die Anforderung an den Cloud Service zusammengefasst.

- Einhalterung der GDPR.
- Speicherung der Daten außerhalb der EU.

- Verschlüsselung muss als Anonymisierung gelten.
- Die Daten müssen für externe Firmen als anonymisiert gelten, damit diese nicht unter die GDPR fallen.
- Bereits bei der Kommunikation von der Firma in die Cloud, müssen die Daten als anonymisiert gelten2.

4 Architektur

Für die Realisierung der GDPR konformen Datenverarbeitung ohne die Erteilung eines Verarbeitungsauftrages an das jeweilige Cloud Service Provider (CSP) soll Intel SGX eingesetzt werden. Intel SGX bietet die Möglichkeit gesicherte Programmbereiche auszuführen, sodass von extern kein Zugriff möglich ist. Diese Funktionsweise bildet die geschützte Datenverarbeitung innerhalb eines Rechenzentrums ab, ohne eine Zugriffsmöglichkeit direkt von anderen Instanzen auf der physikalischen Maschine sowie den indirekten Zugriff über die Infrastruktur durch das CSP.

In Abbildung 2 ist der Aufbau skizziert. Der grüne Bereich repräsentiert hierbei den vertrauenswürdigen Bereich und die direkte Zuständigkeit eines beliebigen Unternehmens. Auf der rechten Seite ist dieses Unternehm abgebildet, welches über verschiedene Clients verfügt und über eine gesicherte Verbindung auf den Datencontainer bei einen Hosting Container zugreifen möchte.

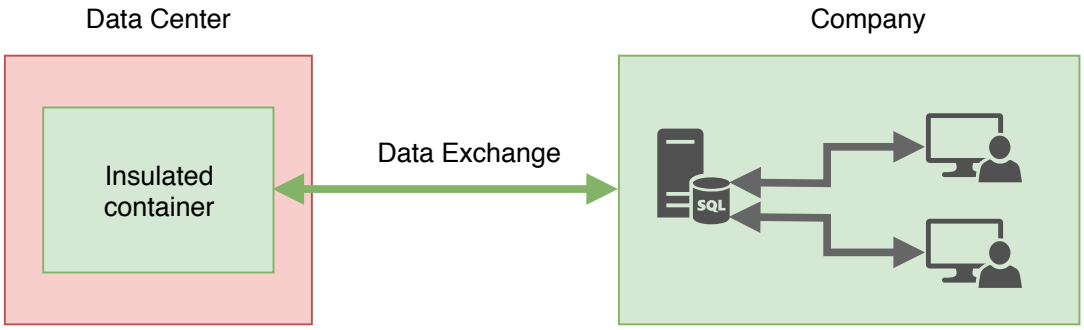

Abbildung 1: Skizzierung des Ziels einer gesicherten Umgebung bei einem CSP

4.1 Intel SGX

In diesem Abschnitt werden lediglich die wichtigsten Komponenten beschrieben, welche für das Verständnis der Architektur notwendig sind. Intel SGX ermöglicht die Bereitstellung von sicheren Programmumgebungen, die sogenannten *Enclaves*. Eine Enclave besteht aus zwei essentiellen Bestandteilen. Das Rahmenwerk einer Enclave definiert der *Unsecure Part*, welcher keinerlei Schutzfunktionen unterliegt, jedoch die Schnittstellen zu dem eigentlichen, gesicherten Container innerhalb einer Enclave beinhaltet.

Die Enclaven werden innerhalb dem Dynamic Random Access Memory (DRAM) abgelegt. In diesem existiert ein zusammenhängender Speicherbeich, welche den sogenannten Processor Reserved Memory (PRM) beinhaltet. Auf diesen Bereich kann durch die Software nicht direkt zugegriffen werden. Der PRM beinhaltet wiederum einen Speicherbereich, welcher Enclave Page Cache (EPC) bezeichnet wird. Der EPC enthält alle Enclaves, die auf dem jeweiligen System agieren [1]. Hierbei ist zu erwähnen, dass die parallele Ausführung mehrere Enclaves möglich ist.

Die maximale Größe des EPC beträgt 128 MiB. Hiervon können 93 MiB aktiv genutzt werden. Übersteigt eine Enclave die genannte Größe, so ist das kostspielige Swapping über den EPC notwendig. [2] Programme, die innerhalb einer Enclave ausgeführt werden können, sind in den Programmiersprachen *C* oder *C++* zu schreiben. Zudem ist es grundsätzlich möglich beliebige Programmbibliotheken der jeweiligen Programmiersprache zu importieren.

4.2 Struktur einer Enclave

In diesem Abschnitt die die Struktur der Enclave betrachtet werden, welche genutzt wird, um Anfragen zu bearbeiten. Der erwähnte Use-Case ist das Handling von Datenbankoperationen auf einer verschlüsselten

Structured Query Language (SQL) Datenbank. Das Interface für die gesicherte, externe Kommunikation wird zunächst abstrahiert. Es wird davon ausgegangen, dass die SQL Instruktionen über ein sicheres Verfahren die Enclave erreichen. Diese stellt hierfür ein External Interface bereit. Diese Interface wird innerhalb des *Unsicheren Bereichs* ausgeführt. Die Datenkommunikation dient hierbei lediglich als Relay und wird an das *Communication Interface* des sicheren Bereichs weitergeleitet.

Anschließend wird mit der $EncryptionEngine_{External}$ die Anfrage entschlüsselt und die extrahierten Instruktionen an den *SQL Handler* weitergereicht. Parallel wird eine Anfrage an die $EncryptionEngine_{Storage}$ gestellt, welche die erforderlichen Daten anfragt. Auf Grund der möglichen Dimensionen einer Datenbank, muss diese außerhalb des sicheren Container gespeichert werden. Daraus resultiert, dass die Datenbank verschlüsselt werden muss, sodass keine Daten im Klartext die sichere Enclave verlassen. Die $EncryptionEngine_{Storage}$ fordert daher die Datenbank über den *Perstistence Storage Handler* an und übernimmt die Entschlüsselung der Daten. Die Daten werden anschließend dem *SQL Handler* im Klartext zugeführt, sodass die angefragte Instruktion darauf angewendet werden kann.

Nach der erfolgreichen Anwendung werden die neuen Datensätze über die $EncryptionEngine_{Storage}$ und den *Perstistence Storage Handler* gesichert. Ergänzend dazu werden die Ergebnisse der Anfrage über die $EncryptionEngine_{External}$ und das *Communication Interface* an den Kunden zurück gesendet.

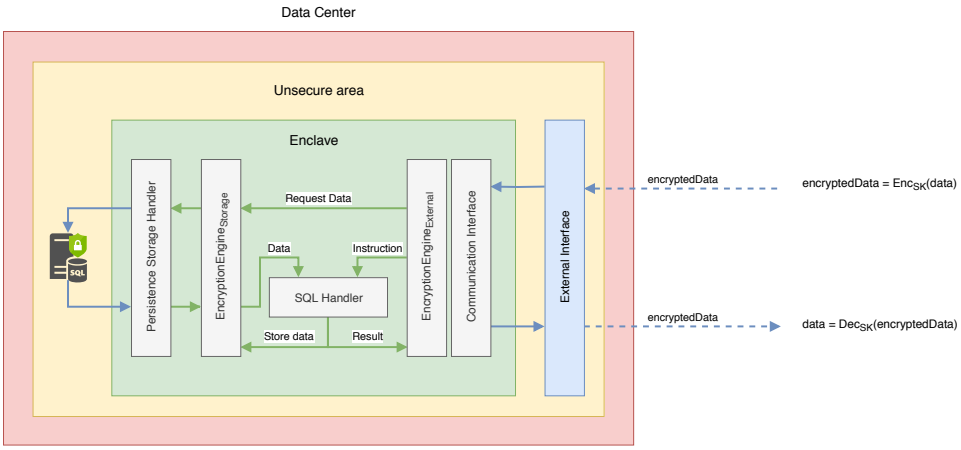

Abbildung 2: Skizzierung des Ziels einer gesicherten Umgebung bei einem CSP

Literatur

[1] Costan, Victor, and Srinivas Devadas. 2016. Intel SGX Explained. http://eprint.iacr.org/2016/086 (November 8, 2019).

[2] Weichbrodt, Nico, Pierre-Louis Aublin, and Rüdiger Kapitza. 2018. "Sgx-Perf: A Performance Analysis Tool for Intel SGX Enclaves." In Proceedings of the 19th International Middleware Conference on - Middleware '18, Rennes, France: ACM Press, 201–13. http://dl.acm.org/citation.cfm?doid=3274808.3274824 (January 21, 2020).