Distributed Shuffle Index: Analysis and Implementation in an Industrial Testbed

Enrico Bacis*, Alan Barnett†, Andrew Byrne†, Sabrina De Capitani di Vimercati‡, Sara Foresti‡,
Stefano Paraboschi*, Marco Rosa*, Pierangela Samarati‡
*Università degli Studi di Bergamo, 24044 Dalmine - Italy Email: firstname.lastname@unibg.it
†Dell EMC - Ireland Email: firstname.lastname@dell.com
‡Università degli Studi di Milano, 26013 Crema - Italy Email: firstname.lastname@unimi.it

***Abstract*—The protection of content confidentiality as well as of access and pattern confidentiality of data moved to the cloud have been recently the subject of several investigations. The distributed shuffle index addresses these issues by randomly partitioning data among three independent cloud providers. In this paper, we describe the implementation of the distributed shuffle index in the high-performance Dell EMC platform. We first illustrate the main characteristics of the industry testbed and then show the results of our experiments, confirming the limited performance overhead and the practical applicability of the distributed shuffle index in industrial environments.**

I. Introduction

The problem of protecting data confidentiality has been widely addressed by the research community. Encryption is typically employed to protect data at-rest and data in-transit. However, by observing accesses, a cloud provider could infer sensitive information about the user performing the access and the possibly sensitive content of the outsourced dataset. Recently, several approaches have been proposed for protecting access and pattern confidentiality (e.g., [1], [2], [3], [4], [5], [6], [7]). Among them, the *distributed shuffle index* [7] is a B+-tree index structure that enables efficient key-based data retrieval, while guaranteeing content, access, and pattern confidentiality. It relies on the presence of three independent cloud providers to improve the protection guarantees offered by the single-provider shuffle index [2]. The distributed shuffle index is based on the combined adoption of *data distribution* and *swapping* protection techniques. Data distribution consists in allocating the nodes composing the shuffle index at three different and independent cloud providers. Each provider will then store and have visibility on accesses only on a portion of the index structure. Swapping consists in continuously changing the physical allocation of accessed data, which are moved to a different cloud provider after each access. The combined adoption of data distribution and swapping guarantees protection of access confidentiality also in case of collusion among the cloud providers. With this configuration, a cloud provider may not even know that data have been distributed, and observes accesses to only a portion of the shuffle index, while assuming that it is observing the whole dataset.

In this paper, we investigate the practical applicability of the distributed shuffle index within a real-world industrial environment. The shuffle index has been then implemented and deployed in an industrial testbed across three data centers. The experimental results confirm its limited performance overhead.

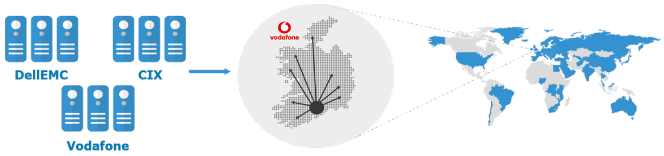

Fig. 1. INFINITE testbed

II. Industrial prototype of the distributed shuffle index

The distributed shuffle index has been implemented in a prototype and deployed in the Dell EMC's INFINITE (INternational Future INdustrial Internet TEstbed - www.iotinfinite.org) platform, a comprehensive IoT innovation platform built for the development of Industrial IoT products and solutions across a wide and diverse range of industries and sectors. It is a strategic initiative led by Dell EMC, Vodafone Ireland and partners, and it is the first-of-its kind in Europe. In the following, we first describe the architecture of the testbed and then present the deployment model.

A. Architecture of the testbed

The testbed is composed of a full mobile network (2G to LTE) covering the island of Ireland and a cloud infrastructure with a distributed data center architecture (Figure 1). In particular, the data center architecture spans three geographically diverse sites: *i) Dell EMC* datacenter that contains compute, storage and networking resources (provided via a VxBlock 1, which is configured as a VMWare cluster), and analytics platforms; *ii) Vodafone* datacenter that contains compute, storage, and networking resources; *iii) CIX* datacenter that hosts compute, storage, and networking resources.

The core network is an MPLS dark fiber ring offering 10 Gbps capacity. Each site is connected to the dark fiber ring through a MPLS router and a DWDM Mux/Demux optical device. The core network extends a Layer 2 domain across the three sites by creating VPLS (virtual private LAN service) between the MPLS routers. MPLS technology enables the flexibility to add new sites in the future or to change their connectivity.

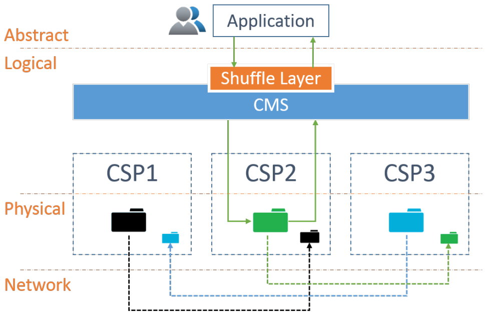

Fig. 2. Shuffle index deployment model

	Plain encrypted index	*Distributed shuffle index*
ECS	$0.04210s$ $\sigma = 0.01322s$	$0.19674s$ $\sigma = 0.08075s$
Commercial Providers	$0.56777s$ $\sigma = 0.25588s$	$1.07609s$ $\sigma = 0.42817s$

Fig. 3. Access times and their standard deviation σ

B. Deployment model

Elastic Cloud Storage (ECS) is a turnkey software-defined, cloud-scale, object storage platform selected as a target application for integration with the shuffle index. This environment for the prototype replicates a multi-cloud data storage service. The prototype implementation is based on Python due to the wide availability of production-ready libraries that support the interaction with multiple cloud providers. The deployment consists of four VMs (Figure 2). The shuffle index was deployed on a VM running a Content Management Service responsible for the distribution of data across ECS nodes. ECS nodes were deployed to the testbed in three installations configured on CentOS 7 VMs. ECS was set up as three single-node ECS configurations rather than one multi-node distributed ECS cluster. In this way, we simulate three separate and isolated cloud providers that are not 'aware' of each other. Once operational, each single-node configuration of ECS was migrated to a different physical location of the testbed.

III. Experimental results

We evaluate the impact on performance of the protection techniques adopted by a distributed shuffle index using two different cloud providers, ECS and widely used public cloud providers, referred in the following as "Commercial Providers". We omit their names because what matters is on one hand the comparison between an infrastructure that offers high performance in the interconnectivity among systems controlled by independent entities and, on the other hand, generic public cloud providers. Our performance analysis utilizes two configurations, one for the ECS - INFINITE testbed deployment and one for the Commercial Providers. The data structure used for the experiments is a 2-level index with fan-out 27 and we performed 100 accesses over the distributed shuffle index. Figure 3 shows a comparison between the distributed shuffle index and a plain encrypted index with the same static structure (i.e., a 2-level B+-tree with fan-out 27). The figure reports the average access time and the standard deviation σ.

Conducting this experiment in a public physically distributed network would be subject to several performance penalties due to network latency and congestion factors that will delay the the interactions. Additional factors such as the prioritization of traffic, communication protocols requiring encryption and routing protocols will also affect transfer speeds as data is shuffled across the network. These factors have been minimized in our experiments by deploying the distributed shuffle index to the INFINITE testbed, a dedicated, comprehensive platform for development and validation of applications and services. With dedicated high speed 10 Gbps connections, the issue of traffic congestion is significantly reduced. Comparing the performance of ECS with the performance of Commercial Providers, we can observe one order of magnitude deterioration in performance. This is in part due to the lower transfer rate (100 Mbps), greater physical distance between sites, and number of hops in the Commercial Provider experiment, compared to the INFINITE testbed. The INFINITE testbed results represent a dedicated network and private cloud infrastructure with dedicated resources that was developed as an innovation platform for Industrial IoT solutions. Further experiments are required to evaluate the performance of the Shuffle Index in a wider variety of real world configurations under different workloads

IV. Conclusions

We have reported our experience in the deployment of the distributed shuffle index on the INFINITE testbed that offers a dedicated high speed network between three server sites. The evaluation of the performance of the distributed shuffle index has demonstrated its practical deployment in an Industrial innovation platform, integrating with a commercial object storage application. This is the first step towards achieving the potential of a distributed shuffle index in a multi-cloud infrastructure.

V. Acknowledgment

This work was supported by the EC within the H2020 under grant agreement 644579 (ESCUDO-CLOUD).

References

[1] E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa, and P. Samarati, "Distributed shuffle index in the cloud: Implementation and evaluation," in *Proc. of IEEE CSCloud*, New York, USA, June 2017.

[2] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati, "Efficient and private access to outsourced data," in *Proc. of ICDCS*, Minneapolis, Minnesota, USA, June 2011.

[3] S. Devadas, M. van Dijk, C. Fletcher, L. Ren, E. Shi, and D. Wichs, "Onion ORAM: A constant bandwidth blowup oblivious RAM," in *Proc. of TCC*, TelAviv, Israel, January 2016.

[4] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati, "Shuffle index: Efficient and private access to outsourced data," *ACM TOS*, vol. 11, no. 4, pp. 1–55, October 2015, article 19.

[5] E. Stefanov, M. van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, and S. Devadas, "Path ORAM: An extremely simple Oblivious RAM protocol," in *Proc. of CCS*, Berlin, Germany, November 2013.

[6] E. Stefanov and E. Shi, "Multi-cloud oblivious storage," in *Proc. of CCS*, Berlin, Germany, November 2013.

[7] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati, "Three-server swapping for access confidentiality," *IEEE TCC*, 2015, pre-print.