Stealth Apps for Secure Personal Data Analytics in the Cloud

Josef Spillner*, Lorenzo Miori†, Julian Sanin†
* Technische Universität Dresden, Faculty of Computer Science, 01062 Dresden, Germany
† Free University of Bozen-Bolzano, Faculty of Computer Science, Piazza Domenicani 3, 39100 Bolzano, Italy
Email: josef.spillner@tu-dresden.de, {Lorenzo.Miori,Julian.Sanin}@stud-inf.unibz.it

***Abstract*—Personal Data Analytics (PDA), also called Body Informatics, is a new computing trend driven by the ubiquitous presence of sensing mobile devices and cloud services. Participating persons collect body, health, sports and activity metrics and submit them to analytics portals with incentives such as training progress metering, social comparison and gamification. The uncontrolled cloudification of personal data makes the affected persons vulnerable to tracking, tapping and loss of valuable records. Stealth applications present an alternative software architecture paradigm which mitigates many of the risks. A distributed stealth platform, partially under the control of the user, enables analytics over encrypted and dispersed data in untrusted public cloud environments. In this demo, we showcase a secure public cloud stealth database and an e-sports application running on it.**

I. Background

The increased need to store, process and broker data streams produced by a variety of stationary and mobile sensors (see [1]) is insufficiently met by today's cloud architectures. In particular, basic security and reliability guarantees across organisational boundaries cannot strictly be covered by service level agreements due to a missing technical enforcement in storage and computation service usage. These services are potentially untrusted, volatile, quickly evolving and subject to many other risks. The DFG-funded research project DaaMob ('Dienstbasierte Plattformkonzepte für systemübergreifende Drittanbieteranwendungen mit mobilen Komponenten im Internet der Dinge') aims at narrowing the gap by introducing so-called *stealth platforms* which combine operation-specific data coding with fault-tolerant and adaptive platform features. As opposed to existing cloud platforms offered as services (PaaS), stealth platforms ensure that data is sufficiently encrypted and dispersed before it is transmitted to the cloud. Storage, retrieval and to some extent remote analytical processing will still remain possible. The main benefit for enterprises is that agreements continue to be valid, even though at times with a degraded quality profile, when cloud providers go bankrupt or expose customer data to the public [2]. For individuals, the selective sharing of otherwise well-protected data together with analytical results is a unique feature. PDA applications are therefore strong contenders for being run on stealth platforms [3] with database, complex event processing, pub-sub/messaging and storage services. We back up this claim with a demo consisting of a stealth database in the public cloud and an e-sports PDA application running on it.

II. Key Demonstrator Facts

StealthDB is a column-store native cloud database which spreads its data across in-memory areas, files and cloud resource services according to data distribution and modification strategies. In the demonstration, an e-sports application which gathers GPS locations and auxiliary data from conference participants will be shown in conjunction with StealthDB. The data is inserted into StealthDB and, as soon as the synchronisation has succeeded, will be safe concerning a potential loss or theft of the mobile device. The application can then, from any device, perform key analytics such as determining the total movement distance, average speed and other domain-specific metrics. Using information dispersal and structure-preserving encryption algorithms including homomorphic encryption, all processing happens safely in the untrusted cloud.

The demo will be shown with a mobile device to generate the participant's data stream and perform analytics, and a notebook which runs a couple of virtualised cloud stacks in the event of network issues. The sketch in Fig. 1 visualises the entire setup. A tracking application records sensor events locally, produces GPX files and other data, inserts them into a table which is dispersed into sub-columns and then retrieves aggregate results in real-time. Reproducibility and recomputability will be ensured by providing public access to the source code and reference data sets at http://lab.nubisave.org/stealthdb/.

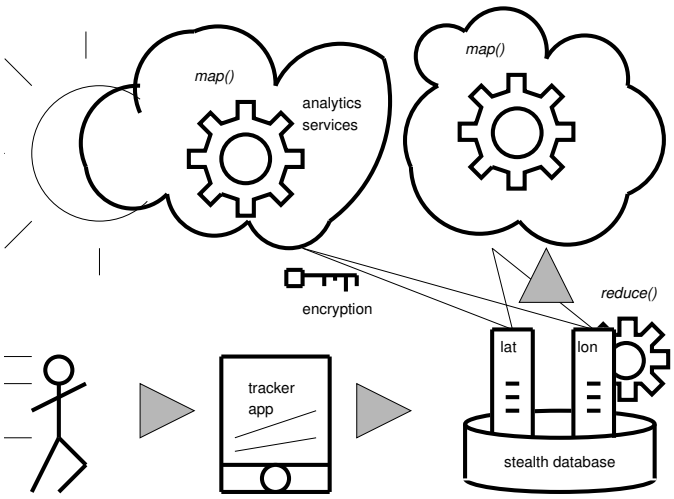

Fig. 1. Data acquisition, insertion and analytics process for e-sports

III. Detailed Demonstrator Description

A number of PDA applications running atop StealthDB have been designed and implemented already for sensed and manually inserted data with high privacy and security requirements. These *stealth apps* encompass a pulse rate and blood oxygen density collector and a flights statistics database.

The database schema of the e-sports application shown in the demo has been designed with typical features found in existing e-sports portals, such as the HeiaHeia social network, Endomondo or Sports Tracker. A first table, `locations`, contains discrete events with location points (longitude, latitude, elevation) and associated timestamps. A second table, `distances`, contains the deltas between these events and normalised date/time representations for easier aggregation. Its columns are of type `REAL` for spatial deltas in meters, temporal deltas in seconds, and corresponding speeds in m/s.

Before running the analytics functions, the data is imported from the user's device as extensible GPS Exchange Format (GPX) files. Generally, stealth databases are set up to store their data with a distribution scheme (full replication, dispersion, round-robin or hash ring) and additional coding (encryption, compression, ordering) depending on the application needs. For the e-sports case, we chose dispersion and encryption which is a suitable stealth coding scheme offering confidentiality and availability. All values are bitsplit into a high-bits part and a low-bits part, along with a redundant part, which requires either two or three cloud providers – two with computation capabilities beyond storage – to realise the scenario. All partial values are encrypted in the Paillier cryptosystem with 128 bit key pairs and additionally in the Boldyreva order-preserving scheme. A key feature of StealthDB is the delayed synchronisation. When no network connection to the cloud providers can be established, the values to be inserted are temporarily stored in memory. Depending on the application's needs, this trades consistency for rapid application development. In the domain of e-sports, it keeps the data reasonably safe and the application simple.

On request, the database system calculates five metrics from this data: Minimum, maximum and average speed (`MIN`, `MAX`, `AVG`), total distance and total time (`SUM`). One challenge is that due to high-precision location and time information, this data is natively provided as floating-point data. On the other hand, both processable encryption and processable dispersion are only working well with integer values. Therefore, the floating-point numbers are first converted to fixed-point integer representations which reduces the precision by a negligible amount. Another concern is the calculation of the average speed which can either be calculated directly over the database column `speed`, although only as approximation due to the different distance weights, or indirectly as a formula over the column `totaldistance` divided by the column `totaltime`.

The calculations are carried out using a map-(carry-)reduce algorithm involving all cloud service providers. For non-ordered queries, a parallel map-reduce function is called, whereas order-sensitive aggregators like `MIN` need an iterative map-carry-reduce implementation which subsequently eliminates all non-matching values across all cloud providers, starting from the one which carries the most significant bits of each value [4]. The capabilities of the stealth database imply a lean application design in which almost all calculations are offloaded to the cloud providers apart from local reduction steps. In order to increase the attractiveness of the application, the data can be projected on a graphical maps view. It can also be selectively shared with other users, although sharing is not currently the focus of our research.

From an application developer's perspective, the programming interface is simple. Listing 1 shows the main method calls for the e-sports application which roughly correspond to the SQL statements supported by StealthDB. Insertions and selections transparently perform query rewriting and chained data transcoding. Integers, for instance, are stored as base64-encoded byte arrays. For increased availability, the keyword `'auto'` chooses the best cloud providers from a registry, replicating or restoring data as needed in order to support cloud service evolution. Once again, the application code can remain free from such infrastructure and network concerns.

Listing 1. Python programming interface for stealth apps

```
cloudlist = \
  ["cloud://googlecompute", "cloud://tudzih"]
clouds = \
  [stealthdb.Cloud(x) for x in cloudlist]
loctable = stealthdb.StealthTable("locations",
  clouds, ["dispersion", "encryption"])
loctable.addcolumn("lat", "REAL")
loctable.addcolumn("lon", "REAL")
...
loctable.columns["lat"].addentry(lat)
...
avglat = \
  loctable.columns["lat"].aggregate("avg")
```


Acknowledgements

This work has been partially funded by the German Research Foundation (DFG) under project agreements SCHI 402/11-1 as well as a visiting researcher grant from the Free University of Bolzano.

References

[1] M. Franke, C. Seidl, and T. Schlegel, "A seamless integration, semantic middleware for cyber-physical systems," in *10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Evry, France, April 2013, pp. 627–632.

[2] L. Ramaswamy, V. Lawson, and S. V. Gogineni, "Towards a Quality-centric Big Data Architecture for Federated Sensor Services," in *IEEE International Congress on Big Data*, Santa Clara, California, USA, June 2013, pp. 86–93.

[3] L. Jalali, D. Huo, H. Oh, M. Tang, S. Pongpaichet, and R. Jain, "Personicle: Personal Chronicle of Life Events," in *Workshop on Personal Data Analytics in the Internet of Things (PDA@IOT) at the 40th International Conference on Very Large Databases (VLDB)*, Hangzhou, China, September 2014.

[4] J. Spillner and A. Schill, "Algorithms for Dispersed Processing," in *1st International Workshop on Advances in Cloud Computing Legislation, Accountability, Security and Privacy (CLASP)*, December 2014, pp. 914–921, London, UK.