Integrity Coded Databases

Dan Kondratyuk, Jake Rodden, Elmer Duran | Dr. Jyh-haw Yeh | Boise State University

Background

Recently, cloud database storage has become an inexpensive and convenient option to store information; however, this relatively new area of service can be vulnerable to security breaches [1]. Storing data in a foreign location requires the owner to relinquish control of their information. This opens the possibility for internal, malicious attacks that can involve the manipulation, omission, or addition of data [2].

Our research tests a potential solution for retaining data as it was intended to be stored in these cloud databases: by converting the original databases to Integrity Coded Databases (ICDB) [3]. ICDBs utilize Integrity Codes (IC): cryptographic codes created for the data by a private key that only the data owner has access to. When the database is queried, an integrity code is returned along with the queried information. The owner is able to verify that the information is correct and fresh [3]. Consequently, ICDBs also incur performance and memory penalties. In our research, we explore, test, and benchmark ICDBs to determine the costs and benefits of maintaining an ICDB versus a standard database.

emp_no	birth_date	first_name	last_name	gender
10001	1953-09-02	Georgi	Facello	M
10002	1964-06-02	Bezalel	Simmel	F
10003	1959-12-03	Parto	Bamford	M
10004	1954-05-01	Chirstian	Koblick	M

Relational Databases are organized like a spreadsheet: columns are attributes, and rows are instances.

Objectives

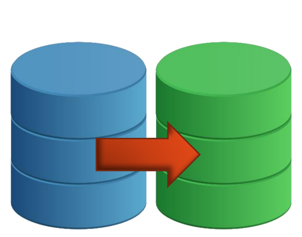
- Implement an Integrity Coded Database (ICDB)
- Verify that the data owner is able to detect malicious changes
- Test the performance of an ICDB
- Compare the performance to a standard database

Procedure

Implementation

MySQL was used to set up and configure several databases [4]

Conversion



Java conversion modules generated all integrity codes

Testing

MySQLSlap and Workbench tested a variety of queries [5]

Analysis

Results were analyzed and compared for relative memory and performance

Benchmarks

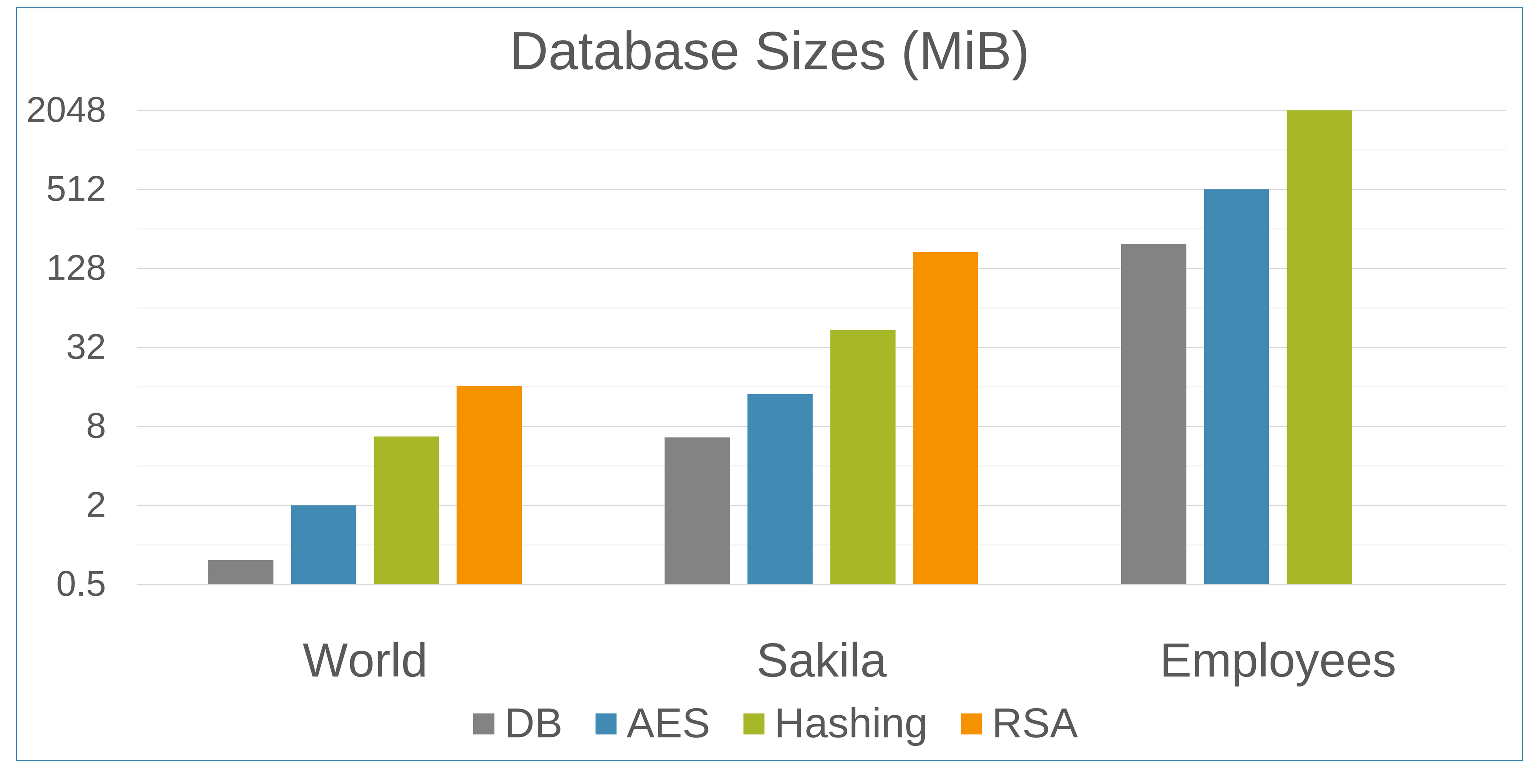


Figure 1. Database size relationships between 3 databases converted with AES, Hashing, and RSA. This chart uses a logarithmic base 2 scale, measured in Mibibytes.

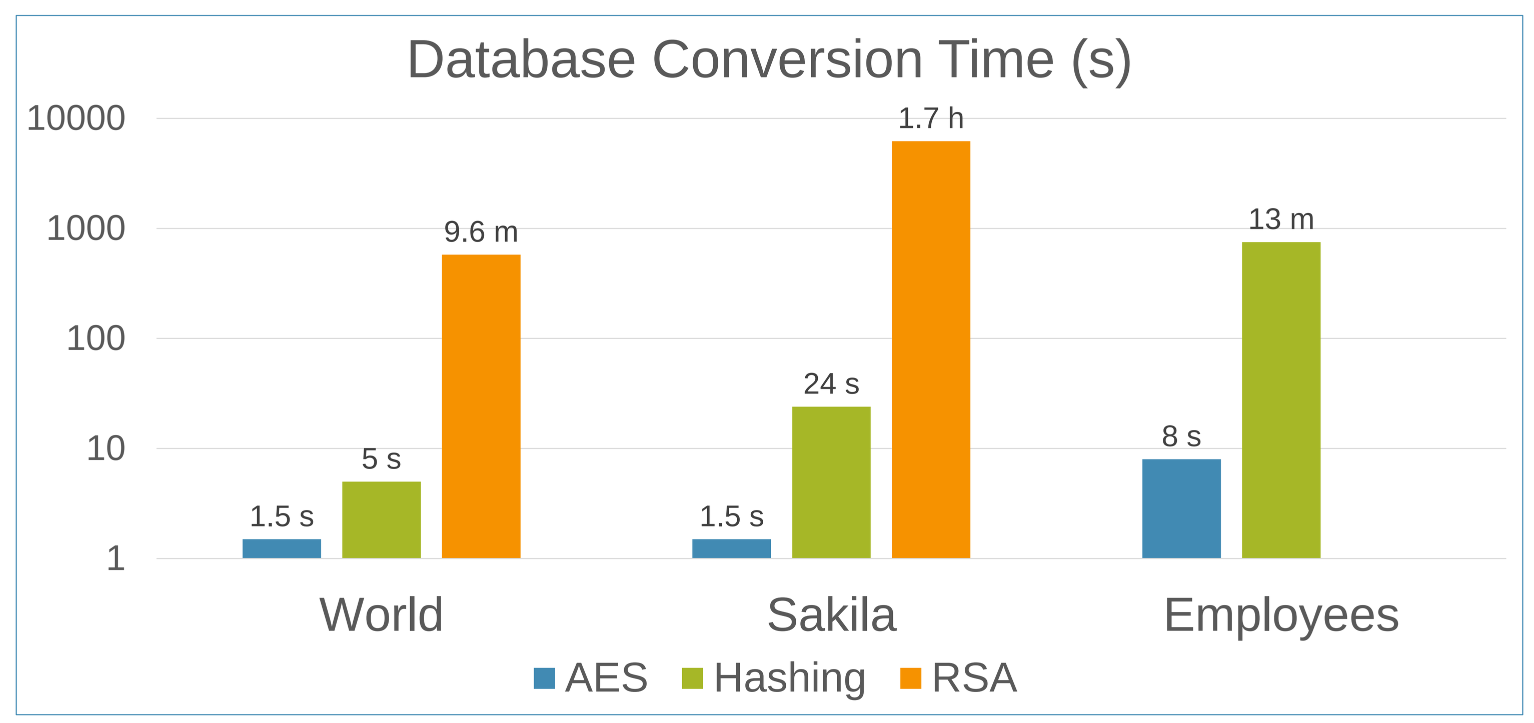


Figure 2. Database avg. conversion time relationships between 3 databases converted with AES, Hashing, and RSA. This chart uses a logarithmic base 10 scale, measured in seconds.

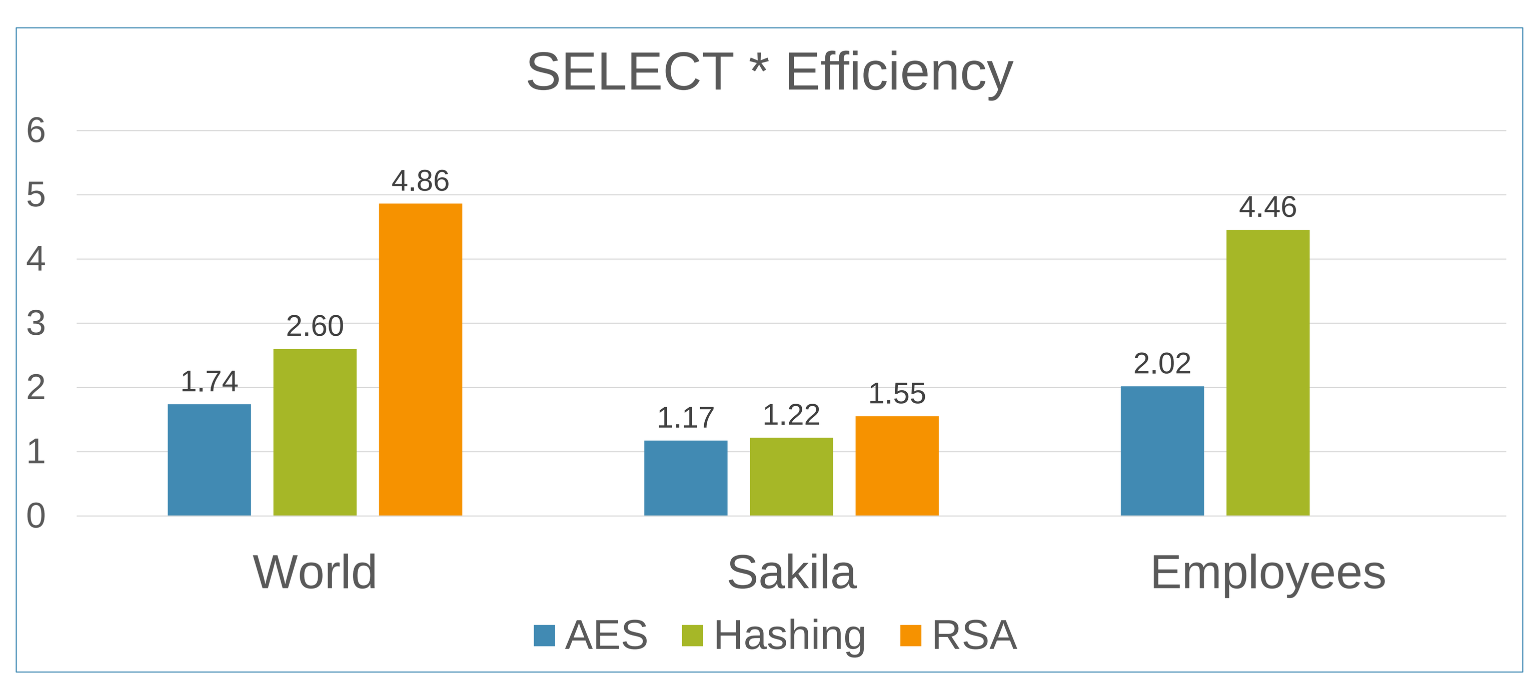


Figure 3. Database avg. query efficiency is measured by dividing the ICDB execution/retrieval time by the standard database execution/retrieval time. Each data point is a multiple of the query execution on a standard database.

emp_no	emp_no_svc	birth_date	birth_date_svc
10001	7ab91676255a...	1953-09-02	156ba6874f769...
10002	6b949c6cd5d2...	1964-06-02	2566e6adb0f4fd...
10003	1d6d0773eea2...	1959-12-03	548b2d56fb7bb...

ICDBs can contain integrity codes along each data entry

Results

- ICDBs are much larger than their standard database counterparts, by a factor of at least 2
- Different implementations (AES, Hashing, RSA) offer unique approaches for an ICDB solution
- AES used the least memory, while RSA used the most
- AES converted in seconds, while RSA can take hours
- AES queried the fastest, while RSA queried the slowest
- Queries can take 1.2 – 5.0 times as long to execute, depending on the complexity of the query and the size of each integrity code
- ICDBs are able to verify against data forgery, data substitution, and old data attacks

Conclusion

- Correctness and Freshness can be verified, but not Completeness
- ICDBs incur heavy memory and speed performance penalties
- RSA is infeasible for practical use, as hashing and AES provide much better results
- AES provides the best ICDB implementation due to its low memory cost, quick conversion time, and great query efficiency

ICDB Scheme	Size Increase	Conversion Speed
RSA	23x	1 KiB/s
Hashing	9x	250 KiB/s
AES	2.5x	25 MiB/s

These data points show the increase in memory cost and conversion speed of the 3 different ICDB implementations

References

[1] Lou, Wenjing, Cong Wang, Qian Wang, and Kui Ren. Ensuring Data Storage Security in Cloud Computing. Illinois Institute of Technology, n.d. Web. <https://eprint.iacr.org/2009/081.pdf>.

[2] Angeles, Sara. "Cloud vs. Data Center: What's the Difference?" Business News Daily. N.p., 26 Aug. 2013. Web. <http://www.businessnewsdaily.com/4982-cloud-vs-data-center.html>.

[3] Yeh, Jyh-haw. *Integrity Coded Database (ICDB) - Protecting Data Freshness and Correctness for Outsourced Databases in Clouds*. Boise State University, n.d. Web. June-July 2015.

[4] MySQL. MySQL Help Tables Documentation. MySQL Documentation: Other MySQL Documentation. Vers. 5.5 - 5.7. Oracle, n.d. Web. <http://dev.mysql.com/doc/index-other.html>.

[5] Oracle. *MySQL Workbench*. Computer software. *MySQL - The World's Most Popular Open Source Database*. Vers. 6.3. Oracle, 2008. Web. June-July 2015.

Acknowledgements

We would like to thank our mentors Dr. Yeh and Dr. Xu for organizing our research project. This work is supported by US National Science Foundation (NSF) under grant CNS 1461133.