2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)

Sport Wearable Biometric Data Encrypted Emulation and Storage in Cloud

Nick Mcdonald, Daniel Atkinson and Youry Khmelevsky*
Computer Science, Okanagan College, Kelowna, BC, Canada
Emails: nick.mcdonald.94@gmail.com, daniel_atkinson@mail.com,
and ykhmelevsky@okanagan.bc.ca

Scott McMillan
XCo Tech Inc., Penticton, BC Canada
Email: scott@xco.io

*Also Affiliated with Mathematics, Statistics, Physics, and Computer Science
Irving K. Barber School of Arts and Sciences, UBC Okanagan, BC Canada

***Abstract*—We investigated different encryption algorithms for sport wearable devices by utilizing a newly developed data generator for the testing purposes. Additionally we investigated different data encryption algorithms for a NoSQL DBMS. Testing results for data generator, data encryption and NoSQL database stress testing are presented and discussed as well. The research project was conducted in support of NSERC grant "GAUGE: Exact Positioning Systems For Sport and Healthcare Industries".**

I. Introduction

XCo Tech Inc. (XCo) is engineering sensor systems for the collection, examination, and display of statistics for sporting and healthcare environments. Information is gathered from one or more sensors and is relayed to servers where it is collated and evaluated. This collection of data and the statistics derived from it can then be viewed in real-time by users on a mobile application and used to make instant decisions for players or patients.

A critical facet of XCo's platform and the value of it is the ability for the system to correctly calculate key statistics that enable a coach or trainer to make accurate decisions to maximize the effectiveness of their team. To this end a great deal of realistic player data was needed to simulate the movement and actions of players including acceleration, velocity, and hits so that the system could be made to successfully recognize such events. To facilitate this need, as part of this project we developed a data generator which could emulate the sensors developed by XCo and create a significant amount of data for them to be able to test the analytics of their system.

The second avenue of research conducted by our team and detailed in this paper are investigations into encryption algorithms with respect to NoSQL databases. Due to the potentially sensitive nature of the information that players or patients would be producing with the sensors it was important that some or all of the data be encrypted to protect against intruders or against malicious access. Therefore we sought to test the performance of encryption and decryption algorithms and their integration with databases.

Our main contributions are: 1) data generation with NoSQL database integration and performance optimization, 2) performance analysis of encryption algorithms with respect to a NoSQL database.

II. Related Works

In [1] authors present "cycling helmet that displays heart rate data". In [2] authors investigate "personal and wearable sports technology". In [3] authors discuss "Ambient Health Monitoring" as an example of self-monitoring system and "as part of a healthy lifestyle and as an enabler of appropriate healthcare services in Ambient Assisted Living (AAL)". They used different types of sensors, including "stationary sensor networks in AAL infrastructure, wireless medical device sensors, embedded mobile device sensors as well as virtual sensors". A prototype application for Android was presented as well.

In [4] authors discussed "sensor-based wellness data" collection for ice-hockey players. They inform, that "numerous sports tracking applications exist for mobile phones and smart watches, bracelets and other wearable sensors are becoming increasingly popular form factors for detecting location, physical activity and biometric data" [4].

M. Chebalier et al. (2015) state that "existing benchmarks rely mostly on the relational data model and do not take into account other models" [5]. They proposed an extension to Star Schema Benchmark (SSB) for NoSQL DBMS. They used data generators and experimentations as well. The HBase and MongoDbB were used with the data loader. E. Dede et al. (2013) evaluate performance of "document oriented NoSQL MongoDb and Hadoop" databases [6]. Additionally they evaluated scalability and fault-tolerance of these databases. They found, that in "a 32 node cluster, with 8 node failures, Hadoop-HDFS fails to complete the MapReduce job since the worker nodes also host chunks of the input data".

In [7] authors discuss benchmarking of big data analytics and a synthetic data generator for BigBench data model as well as Teradata Aster DBMS. In a case-study paper "Benchmarking cloud-based tagging services" [8] authors describe "a benchmark for tagging services, and propose benchmarking modules that can be used to evaluate the suitability of a database for workloads generated from tagging services".

Two NoSQL datastores based on HBase and Cassandra are studied in [9]. They proposed the use of a benchmarking studio to generate data for performance measuring. They found, that Cassandra's writing performance is much better than HBase's capabilities (for about 1 million writes per second).

978-1-4673-8721-7/16/$31.00 ©2016 IEEE

For optimized batched writes Cassandra is twice better than HBase. On the other hand, they found that HBase is about 30% better than Cassandra for low-density data read.

A "new encryption paradigm, referred to as asymmetric cross-cryptosystem re-encryption (ACCRE)" was presented in [10]. A "ciphertext conversion mechanism that allows an authorized proxy to convert a complicated IBBE ciphertext into a simple IBE ciphertext affordable to mobile devices, without leaking any sensitive information to the proxy" was used. An "EnCore mobile platform" is discussed in [11]. It's built "on secure encounters between pairs of devices as a foundation for privacy-preserving communication". A mobile development environment for Android was used for the EnCore. They evaluated EnCore's utility with more than 30 users. "EnCore relies on D2D radio communication, but incorporates an efficient periodic MAC-address change protocol that ensures users cannot be tracked using their MAC address".

S. Holla and P. Dala-Krishna (2012) proposed a data encryption apps mHealth for health mobile applications in [12]. The "construction, performance evaluation, and validation of a data encryption solution for mobile health apps (DE4MHA)" were performed. The proposed solution "did not deteriorate the overall network performance and the app, maintaining similar performance levels as without the encryption" [12].

AES, DES and Triple DES symmetric key algorithms were discussed by Er. A. Pansotra and Er. S. P. Singh (2013) in [13]. The asymmetric Diffie-Hellman Key Exchange, RSA and Homomorphic equations were discussed as well. They either don't compare or don't give any recommendation, but we discuss differences of encryption algorithms in our research.

Darpan D. Shah et al. (2015) discuss Blowfish and AES in [14]. They discuss "a new encryption technique named D's crypto-cipher technique (DCCT), based on random key generation", but this is out of scope of our research paper.

In [15] DES, 3DES, Blowfish and AES (Rijndael) performance comparison is presented. They used several encryption settings with different sizes of data blocks for encryption/decryption speed. The C# language was used. They found, that Blowfish has a better performance results when compare to other algorithms. They suggest the use of Blowfish as a standard encryption algorithm. They also note that AES requires more processing power.

III. Environment Setup

A. Amazon Server

We conducted our research in Cloud at Amazon Elastic Compute Cloud (Amazon EC2) [16]. The requirement for our short research project was to investigate possibilities of Hybrid or Private Cloud utilization for NoSQL database, including performance investigation with and without data encryption. We started with t2.micro instance, 1GB RAM and 1 vCPU. Ubuntu 14.04 linux server was selected as an operating system (OS) for the project. The selected platform allows scaling up by adding CPUs and RAM, if it's necessary. On the other hand, t2.micro instance is currently free for AWS users, but it can be changed by the service provider.

IV. Data Generator

The project sponsor is developing sensors that track a player's location and biological data such as heart rate. This data is relayed to a server which performs calculations including velocity and acceleration before forwarding the final data packet to a database. The data generator was designed to emulate all of this player data in a rapid, lightweight, and realistic manner to simulate a sensor being worn by an active player to produce data sets to test the server's analytics processes. To achieve this coordinates and the heart rate were defined as functions of time and then sine waves were used to modulate them. Using this data the remaining data could be calculated (ie. displacement, velocity, and acceleration) as they would be in the actual system. This data is converted to a JSON object and sent to the database.

Additionally, we used the generator to test the capacity of full system. For instance, we tested the viability of encrypting the data on the relay server before forwarding the data to the database. This encrypted data was written to a NoSQL database in sequence to ascertain how much inbound data could be processed.

A. Data Generator Performance

It was required that the data generator be capable of producing a minimum of 1000 samples per second (or 1 sample per millisecond) per emulated sensor, to effectively match the output speed the actual sensor. This condition constrained the number of sensors that could be emulated at a time as can be seen in Fig. 1. Without encryption the generator can adequately emulate approximately 120 sensors. AES encryption can generate at least 1000 samples/second for up to 50 sensors and Blowfish manages until 40 sensors. With Triple DES encryption the generator can produce the requisite number of samples for up to only 20 sensors. Since the generator was not implemented with multi-threading in mind it can be assumed that use of multi-core CPUs and multi-threading would improve sample generation and encryption. At 1000 samples/second, the emulation of 130 sensors generates data at approximately 54 MB/s (130 sensors * 1000 samples * 435 byte documents) and 21 MB/s when encryption is employed. For the purpose of testing a NoSQL database, it was determined that the output volume of data would be adequate.

B. NoSQL Database Performance

The data generator was tested to determine the speed at which data from 100 sensors could be emulated, encrypted, and inserted into the database, the results of which can be seen in Fig. 2. The generator, when not using any form of encryption, could insert into the database at a pace of approximately 10000 inserts/second or about 4 MB/s. Employment of AES and Blowfish encryption algorithms were concluded to not add much overhead to the process with them capable of 9100 and 8100 inserts/second respectively. Triple DES produced the most overhead and managing only 6300 inserts/second.

V. Choosing Encryption Algorithm

When investigating encryption algorithms for our system, we chose to look at the three most commonly used algorithms.

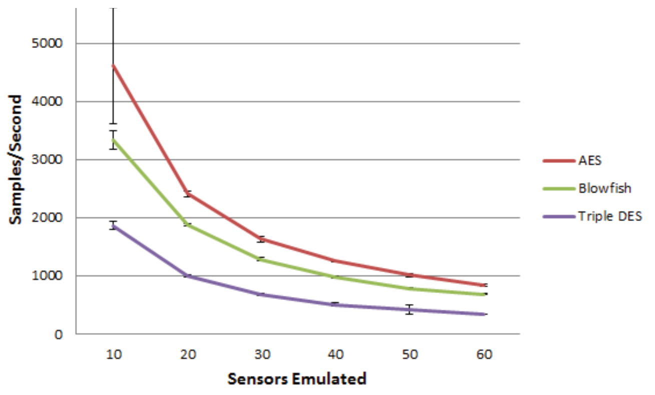

Fig. 1. Data Generator Performance

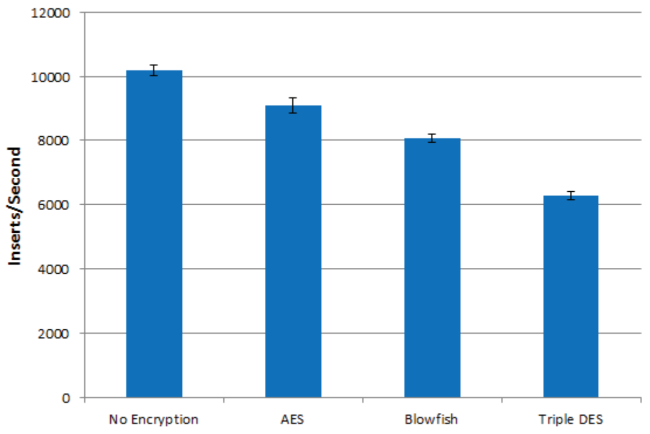

Fig. 2. NoSQL Database Performance

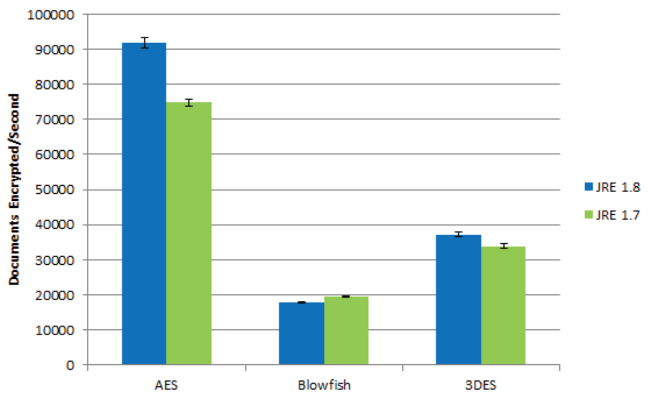

Fig. 3. Encryption Algorithm Performance for 100 Byte Documents

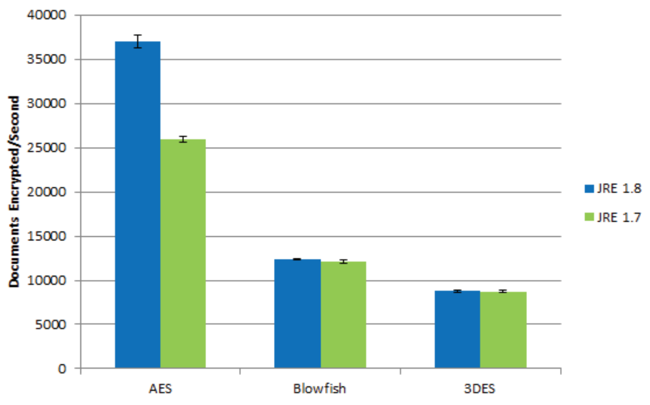

Fig. 4. Encryption Algorithm Performance for 1 Kilobyte Documents

In addition to being the most widely used encryption algorithms they were selected for their stability, security, and ease of implementation. The algorithms addressed in this paper are the AES, Blowfish, and 3DES algorithms.

A. AES vs Blowfish vs 3DES

1) AES Encryption: An AES encryption algorithm [17] utilizes two symmetric keys to code and decode data (meaning that the same key will encrypt and decrypt data). Keys constructed by this algorithm can be 128, 192, or 256 bits.

2) Blowfish: Blowfish [18], [19] employs keys with lengths 32 bits - 448 bits using a block size 64 bits. As of now there is no known attack which can successfully break Blowfish.

3) 3DES: "Data Encryption Standard (DES)" [20] makes use of a 56-bit encryption key with a block size 64 bits. To counter DES' vulnerability to brute-force attacks, due to a small key size and advancements in computer hardware, 3DES was introduced, which effectively repeated the DES algorithm three times to increase key length from 56 bits to 168 bits.

B. Encryption Algorithm Performance

The aforementioned algorithms were integrated into the system and each was tested to determine which one was the best fit for our system. In Fig. 3 and Fig. 4 we tested each algorithm by having the repeadedly encrypt JSON documents over a period of time to determine how many documents per second each algorithm could encrypt. The tests were ran for 5 seconds and repeated 20 times to provide an accurate average. We chose 100 Bytes and 1 Kilobyte as our document sizes because they best fit the size of data that this system and similar systems would be handling.

Our hypothesis was that Blowfish would encrypt faster than AES and 3DES, but surprisingly we found that AES encrypted more than twice the number of documents as Blowfish. Looking into this peculiarity we discovered that the answer lies in optimization. In recent iterations of Java there have been great efforts to optimize their AES implementation including most notably making use of Intel®'s AES-NI (AES New Instruction set) [21] which conducts AES encryption directly on hardware.

Fig. 3 and Fig. 4 show how each algorithm performed and different versions of Java and on different sizes of data. We found that AES out performed Blowfish and 3DES in both cases, and has improved greatly in recent versions of Java. Blowfish did not perform well with the small amount of data and was outperformed by 3DES, but Blowfish performed better than 3DES with larger amounts of data. Subsequent tests showed that as the size of the data got larger Blowfish performed even better than 3DES but AES was still by far the fastest.

VI. Conclusion

During our several small research and SW Engineering projects we developed a data generator to emulate biometric sensors as well as we investigated different data encryption algorithms for NoSQL DBMS systems, particularly for location and biometric data captured by sports wearable devices.

In our research we have found, that two encryption algorithms worked well. AES and Blowfish algorithms best fit for our purposes. We found, that Blowfish implementation is more secure than AES, but the investigated AES algorithm implementation is faster with large amounts of data, particularly with Intel® AES-IN.

We found, that 3DES has less protection and works slow to compare with the AES and Blowfish. It's outdated and we don't recommend it for the new systems implementation.

Acknowledgments

We would like to thank Computer Science students at Okanagan College A. A. Tayrah, M. Aldaej, K. Almutiri, Y. Asiri, C.K. Chiang, D. Leader, and J. Ohlhauser, who developed initial tests and data generator during a spike project within a capstone project course COSC 470 Software Engineering, 2015. This research was conducted in support of NSERC College Engage grant "GAUGE: Exact Positioning Systems For Sport and Healthcare Industries (GAUGE)".

We would like to thank Amazon, Inc. for awarding us by AWS Grants for Research and Education and by Educational grants. We would like to thank Atlassian for supporting us by community licenses for Jira project management and Bitbucket cloud tools.

Our thanks to IBM Education for supporting us by IBM Rational SW Architect licenses in our system design and SW engineering projects.

References

[1] W. Walmink, D. Wilde, and F. F. Mueller, "Displaying heart rate data on a bicycle helmet to support social exertion experiences," in *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, ser. TEI '14. New York, NY, USA: ACM, 2013, pp. 97–104. [Online]. Available: http://doi.acm.org.ezproxy.okanagan.bc.ca/10.1145/2540930.2540970

[2] J. Tholander and S. Nylander, "Snot, sweat, pain, mud, and snow: Performance and experience in the use of sports watches," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 2913–2922. [Online]. Available: http://doi.acm.org.ezproxy.okanagan.bc.ca/10.1145/2702123.2702482

[3] D. Burmeister, A. Schrader, and D. Carlson, "A modular framework for ambient health monitoring," in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, ser. PervasiveHealth '13. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 401–404. [Online]. Available: http://dx.doi.org.ezproxy.okanagan.bc.ca/10.4108/icst.pervasivehealth.2013.252132

[4] M. Alhonsuo, J. Hapuli, L. Virtanen, A. Colley, and J. Hakkila, "Concepting wearables for ice-hockey youth," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '15. New York, NY, USA: ACM, 2015, pp. 944–946. [Online]. Available: http://doi.acm.org.ezproxy.okanagan.bc.ca/10.1145/2786567.2794315

[5] M. Chevalier, M. E. Malki, A. Kopliku, O. Teste, and R. Tournier, "Benchmark for OLAP on NoSQL technologies comparing NoSQL multidimensional data warehousing solutions," in *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*, May 2015, pp. 480–485.

[6] E. Dede, M. Govindaraju, D. Gunter, R. S. Canon, and L. Ramakrishnan, "Performance evaluation of a MongoDB and Hadoop platform for scientific data analysis," in *Proceedings of the 4th ACM Workshop on Scientific Cloud Computing*, ser. Science Cloud '13. New York, NY, USA: ACM, 2013, pp. 13–20. [Online]. Available: http://doi.acm.org/10.1145/2465848.2465849

[7] A. Ghazal, T. Rabl, M. Hu, F. Raab, M. Poess, A. Crolotte, and H.-A. Jacobsen, "Bigbench: Towards an industry standard benchmark for big data analytics," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '13. New York, NY, USA: ACM, 2013, pp. 1197–1208. [Online]. Available: http://doi.acm.org/10.1145/2463676.2463712

[8] T. Malik, K. Chard, and I. Foster, "Benchmarking cloud-based tagging services," in *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on*, March 2014, pp. 231–238.

[9] A. K. Kalakanti, V. Sudhakaran, V. Raveendran, and N. Menon, "A comprehensive evaluation of NoSQL datastores in the context of historians and sensor data analysis," in *Big Data (Big Data), 2015 IEEE International Conference on*, Oct 2015, pp. 1797–1806.

[10] H. Deng, Q. Wu, B. Qin, W. Susilo, J. Liu, and W. Shi, "Asymmetric cross-cryptosystem re-encryption applicable to efficient and secure mobile access to outsourced data," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '15. New York, NY, USA: ACM, 2015, pp. 393–404. [Online]. Available: http://doi.acm.org.ezproxy.okanagan.bc.ca/10.1145/2714576.2714632

[11] P. Aditya, V. Erdélyi, M. Lentz, E. Shi, B. Bhattacharjee, and P. Druschel, "Encore: Private, context-based communication for mobile social apps," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 135–148. [Online]. Available: http://doi.acm.org.ezproxy.okanagan.bc.ca/10.1145/2594368.2594374

[12] S. Holla and P. Dala-Krishna, "Medical data encryption for communication over a vulnerable system," Sep. 4 2012, uS Patent 8,260,709.

[13] E. A. Pansotra and E. S. P. Singh, "Cloud security algorithms." *International Journal of Security and Its Applications.*, vol. Vol. 9, no. No.10, pp. pp. 353–360, 2015.

[14] D. D. Shah, A. Mittal, and K. K. Jani, "A new approach towards encryption technique: D's crypto-cipher technique (DCCT)," *Advances in Computer Science and Information Technology (ACSIT)*, vol. 2, no. 5, pp. pp. 446–449, April-June 2015.

[15] A.-K. A. Tamimi. Performance analysis of data encryption algorithms. [Online]. Available: http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/

[16] Amazon Web Services Inc. Amazon elastic compute cloud. [Online]. Available: http://aws.amazon.com

[17] I. C. Aleksey Ignatenko. Improved advanced encryption standard (AES) crypto performance on java with nss using intel® aes-ni instructions. [Online]. Available: https://software.intel.com/en-us/node/185207

[18] S. Inc., "Blowfish encryption." http://www.splashdata.com/splashid/blowfish.htm.

[19] B. Schneier. The Blowfish encryption algorithm. [Online]. Available: https://www.schneier.com/blowfish.html

[20] M. Rouse. Data encryption standard (DES) definition. [Online]. Available: http://searchsecurity.techtarget.com/definition/Data-Encryption-Standard

[21] I. C. Shay Gueron. Intel® advanced encryption standard (AES) new instructions set. [Online]. Available: https://software.intel.com/sites/default/files/article/165683/aes-wp-2012-09-22-v01.pdf