Biometric Data Emulation and Encryption for Sport Wearable Devices (A Case Study)

Nick Mcdonald, Daniel Atkinson, Corey Frank and Youry Khmelevsky*
Computer Science, Okanagan College, Kelowna, BC Canada
Emails: nick.mcdonald.94@gmail.com, daniel_atkinson@mail.com,
corey.a.frank@gmail.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*—This paper investigates the biometric data emulation and encryption for the sports wearable devices, including data generation performance with different data encryptions for a NoSQL document database. We discuss more deeply a specific topic, related to testing data generation and data encryption for the performance and stress testing of our NoSQL database.**

I. Introduction

XCo Tech Inc. (Xco), based in Penticton BC, Canada is developing an agnostic sensor platform for enabling interconnectivity, analysis and integration of information for sports, fitness and healthcare. The company's software system collects data from multiple sensors and transmits that data to servers where the data is integrated, synchronized, and analyzed. The data and derived analytics are then transmitted to other devices or persons where an app can use the data and analytics to present valuable real-time information to the user.

Critical to the value-add proposition of the system is the ability to measure a person's location with cm level precision indoors and outdoors. For example, basketball coaches are interested in analyzing the cuts, jumps and bursts performed by a player during play. Therefore, XCo implements algorithms to analyze data from location systems and other sensors to determine such analytics. A second aspect requiring investigation is the synchronization of data from different sensors or systems such that the data can be integrated. For instance, positions obtained from the positioning system could be integrated with data from MEMS (micro-electro-mechanical systems) accelerometers and gyroscopes to better detect and classify maneuvers if the data from the separate sensors and positioning system can be synchronized.

This research is a small part of a gamification/real-time solution and state those requirements within a research project "GAUGE: Exact Positioning Systems For Sport and Healthcare Industries", conducted by undergraduate students at Computer Science department (COSC) at Okanagan College (BC, Canada) with XCo Tech Inc (Kelowna, BC Canada), which was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) in 2015. Our designed system includes a NoSQL database. The emulated data are related for each individual player (personal statistics) as well as between players to provide a competitive aspect.

In this paper we discuss two parts of the project, related to different sensors data emulation as well as data encryption on the way from sensors to a NoSQL database. We investigate different encryption/decryption algorithms and related database performance issues.

Our main contributions are: (1) new area of NoSQL database implementation; (2) data generation for a NoSQL database performance optimization and load testing; (3) performance analysis of different encryption algorithms for use with NoSQL databases: (4) a practical example of Internet of Things.

II. Related Works

In [22] authors discuss the potential of heart rate displays in a social context, by means of an augmented cycling helmet that displays heart rate data. They studied how pairs of cyclists engaged with this setup and found that access to another person's heart rate data can result in social interplay which in turn supports engagement with the exertion activity.

Authors in [21] have conducted interviews with ten elite and recreational athletes to understand their experiences and engagement with endurance sport and personal and wearable sports technology. "Technology played both an instrumental role in measuring performance and feeding bio-data back to them, and an experiential role in supporting and enhancing the sport experience". They suggested two interrelated ways of looking at sports performances and experiences, firstly through the notion of a measured sense of performance, and secondly as a lived-sense of performance.

In paper [5] authors discuss Ambient Health Monitoring, which is becoming increasingly important for supporting proactive self-monitoring as part of a healthy lifestyle and as an enabler of appropriate healthcare services in Ambient Assisted Living (AAL). They describe a modular system approach for integrating heterogeneous context sources, including: stationary sensor networks in AAL infrastructure; wireless medical device sensors; embedded mobile device sensors; as well as virtual sensors. The use cases of the system described by presenting a prototype lifelogging application for Android, which integrates several sensor types into a personal health record, with a special focus on activity recognition. "The application also demonstrates the usage of gamification methods as a persuasive means of enhancing the intrinsic motivation of users towards a personalized healthy lifestyle" [5].

978-1-4673-9549-9/16/$31.00 ©2016 IEEE

In the workshop paper [3], authors present work in progress where they utilize sensor-based wellness data to benefit teenage ice-hockey players in their hobby. They created an application concept and mock-ups of wearable sensors, and conducted a service design workshop with a teenage ice-hockey team. "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" [3].

In paper [16] authors discuss a new model of using NoSQL databases as a storage systems. The tell, that the "new generation of database systems with weaker data consistency models is content with using and managing locally attached individual storage devices and providing data reliability and availability through high-level software features and protocols". They examines the behavior of several NoSQL DBs: HBase and Cassandra. In Summary they conclude, that I/O profile does not differ greatly from traditional RDBMes, but what differs most is their approach to managing data.

On the other hand, authors in [12] investigated three NoSQL database (MongoDb V2.2, Cassandra V2.0 and Riak V1.4) performances for a large, distributed healthcare organization. In their testing, a typical workload and configuration produced throughput that varied from 225 to 3200 operations per second between database products, while read operation latency varied by a factor of 5 and write latency by a factor of 4. They found, that Cassandra DBMS provided the best throughput performance, but with the highest latency.

A Parallel Data Generation Framework (PDGF), a generic data generator is described in [7]. As they inform, "an extremely time and resource consuming task in the creation of new benchmarks is the development of benchmark generators, especially because benchmarks tend to become more and more complex". They presented PDGF Version 2, which contains extensions enabling the generation of update data as well.

Additionally to biometric data emulation and transmission we investigated different types of encryption algorithm for the secure data transmission and storage within a NoSQL database.

In [6] authors propose a new encryption paradigm, referred to as asymmetric cross-cryptosystem re-encryption (ACCRE) by leveraging the asymmetric capacity of the participants to meet the security requirements in mobile access to sensitive data. They use 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.

In [1] authors introduce EnCore, a mobile platform that builds on secure encounters between pairs of devices as a foundation for privacy-preserving communication. Using an Android implementation of EnCore and an app for event-based communication and sharing, they evaluated EnCore's utility using a live testbed deployment with 35 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" [1].

In [9] authors propose a data encryption solution for mobile health apps in cooperation environments. The proposed mHealth system include the use of mobile devices and apps that interact with patients and caretakers. They proposed construction, performance evaluation, and validation of a data encryption solution for mobile health apps (DE4MHA) as well. The data encryption algorithm DE4MHA allow users to safely obtain health information with the data being carried securely. The proposed security mechanisms did not deteriorate the overall network performance and the app, maintaining similar performance levels as without the encryption.

In [14] authors describe symmetric key algorithms DES, AES, and Triple DES, in which a single key is used for encryption and decryption as well as RSA, Diffie-Hellman Key Exchange and Homomorphic equations, which are asymmetric. In the last type of algorithms two different keys are used for encryption and decryption. But the authors don't give any comparison or any recommendation, which we discuss in our current research paper.

In [18] the authors inform, that some of the encryption algorithms are very popular in achieving data security at a great extent like Blowfish and AES, but because as security level is increased the time span and complexity of algorithm to perform various operations is also increased. The authors proposed a new encryption technique named "D's crypto-cipher technique (DCCT)" with random key generation which enhances the security level as well as increasing the speed and efficiency of an encryption system. In our paper we don't investigate this technique.

"When companies are testing cloud applications, e.g. for storage or databases, they use generated data for fear of data loss", as it is stated in abstract of [11]. They give an overview of the architecture and performance benchmarks on a prototype which maybe used for practical adoption. On the other hand, in [13] authors describe encryption implementation for light and temperature sensors with very limited system resources.

III. Environment Setup

A. Amazon Server

For our research we used a development server that was outside of the college network so it was accessible from outside the network without having to deal with the colleges firewall. We chose an Amazon Web Services (AWS) EC2 instance because it gave us the flexibility to do whatever we needed with the server, and also the ability to easily scale up if we needed more resources. This server is used to host a database as well as an automated testing suite. The server is a t2.micro instance with 1GB of memory and 1 vCPU running Ubuntu 14.04. The Architecture of the Data Generator with our encryption/decryption application on the Application Server is shown on Fig. 1.

Our application supports both Android and iOS clients. Since it is only possible to compile and test iOS applications on Apple computers we used an Apple server and install all of the software that we needed to build our application and run it on an emulator.

For version control we used Bitbucket. Bitbucket was a good choice for us because it is free for educational institutions, and it is capable of linking to both our planning and collaboration tool (Jira) and our automated testing suite (Jenkins/Karma).

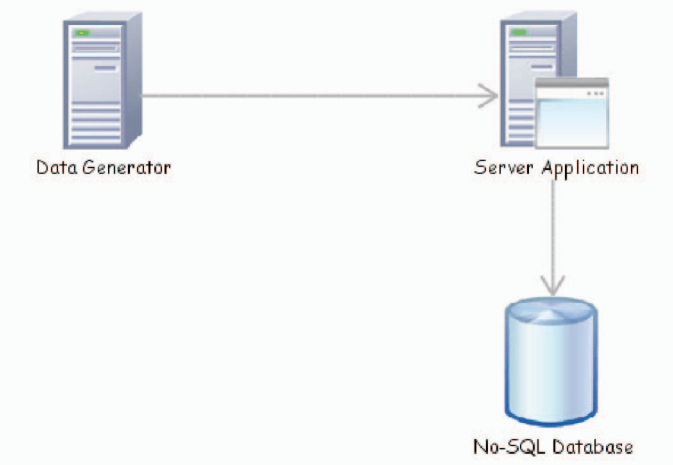

Fig. 1. Architecture of the Data Generation and Data Encryption/Decryption within NoSQL DBMS server.

Jira is a planning and collaboration tool that we used to plan our work and keep track of what needs to be done, and who is working on what. It is based around agile methodologies, and has a plug in for SCRUM which makes it a very good fit for our team.

A group of 9 students in COSC 470 Software (SW) Engineering at Okanagan College (OC) completed a 5 week spike project, in which a data generator was designed and implemented. The data generator emulates sensors data, and sends data across a network and into the database. The research goals of the spike project were to experiment with different encryption methods to use in the system, as well as stress testing the systems to test the limits of the system. A NoSQL database, network, and encryption/decryption algorithms needed to be tested to show what kinds of technological constraints and bottle necks the system might have.

IV. Data Generator

The data generator was designed to emulate the sensors used to track a players motion (xyz coordinates, velocity, and acceleration) and collect biological data such as heart rate. All of these fields needed to be generated in a way that would be at least semi-accurate in simulating the motion of a sensor being worn by a player so that the data could be used to test the analytics of the system. We accomplished this by defining the key fields (x, y, z, heart rate) as functions of time and using sine waves to modulate them. We then calculated the remaining fields (displacement, velocity, acceleration) from those key fields as they would be in the real system. Another purposes of the data generator project was to test the capabilities different parts of the system. For instance we tested how much overhead encryption added, to see if it was viable to encrypt all or part of the data that was transferred to and stored in our NoSQL database. We also tested it against of our NoSQL database to see how much incoming data it could handle.

A. Data Generator Performance

A requirement of the data generator designed by the COSC 470 class was for it to generate data at a rate of at least 1000 samples/second (1 sample/millisecond) for each sensor being emulated, to keep up with the speed of the actual sensors. The data generator was constrained by the number of sensors it could emulate without dropping below 1000 samples/second as shown in Fig. 2. The number of samples/second drops below 1000 at around 130 sensors without encryption and around 40-50 with AES or Blowfish encryption. If the emulation was multi-threaded it would be able to make better use of multi-core CPUs and increase the amount of data we could generate and encrypt, but that isn't necessary for our purposes. Emulating 130 sensors at 1000 samples/second generates approximately 54 MB/s (130 sensors * 1000 samples/second * 435 Bytes document size) and 21 MB/second with encryption enabled. This volume of data should is sufficient for testing against a single instance of the NoSQL database.

B. NoSQL Database performance

We tested the data generator against our database to see how fast we could insert the generated data into the database. Fig. 3 shows the results of a test of 100 sensors being emulated and encrypted using different encryption algorithms and inserting into a NoSQL database running on the same machine. Without encryption the data generator was able to insert at a rate of just over 10000 inserts/second, which is approximately 4 MB/second. AES and Blowfish encryption did not add much overhead with 9100 and 8100 inserts/second respectively. Triple DES gave the most over head with 6300 inserts/second.

V. Choosing Encryption Algorithm

We investigated three of the most commonly used encryption algorithms that could be implemented in our system. The algorithms discussed are the AES, Blowfish, and Triple DES (3DES) encryption algorithms. We outline each algorithm and describe the pro's and con's of each; as well as compare the algorithms performance.

A. AES vs Blowfish vs 3DES

Used Terms:

- *Brute-Force Attack*: When hackers try to decrypt data by trying every possible key in the encryption algorithm.
- *Block Cipher*: A block cipher is a function that encryption algorithms use to encrypt and decrypt files. Block ciphers take two inputs, a key and text. Next it applies the encryption algorithm it was designed for on the text using the provided encryption key [4].
- *Encryption Key*: The encryption key is a string of characters that are fed into the encryption algorithm. These characters act as a key to unlock the algorithm, they will be used backwards through the algorithm to decrypt the data. For every bit used in the key the possible key combinations increases, so longer keys provide a harder code to decrypt. Any key size below 96 Bytes are vulnerable to brute force attacks [20].

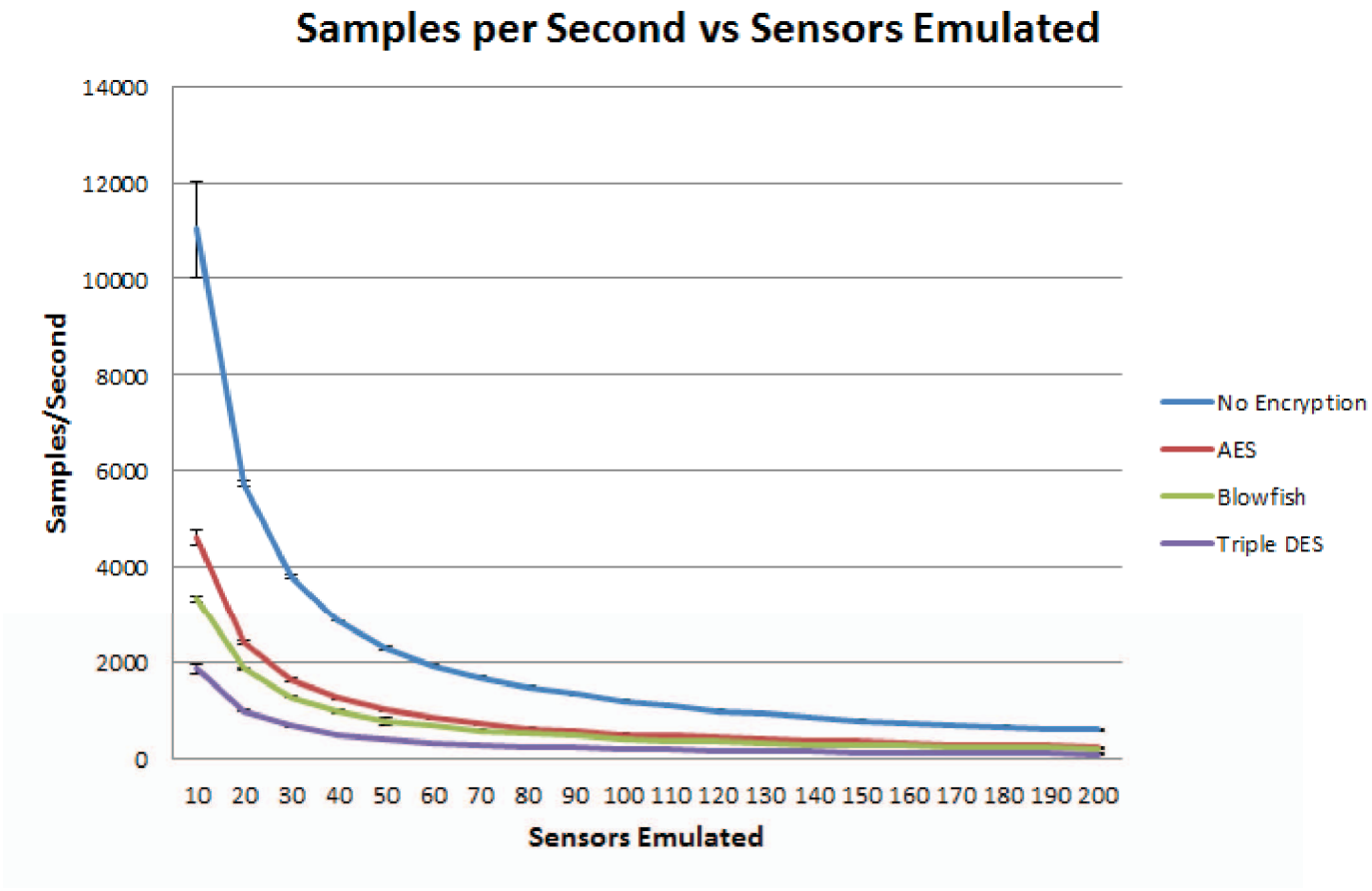

Fig. 2. Data Generator Performance

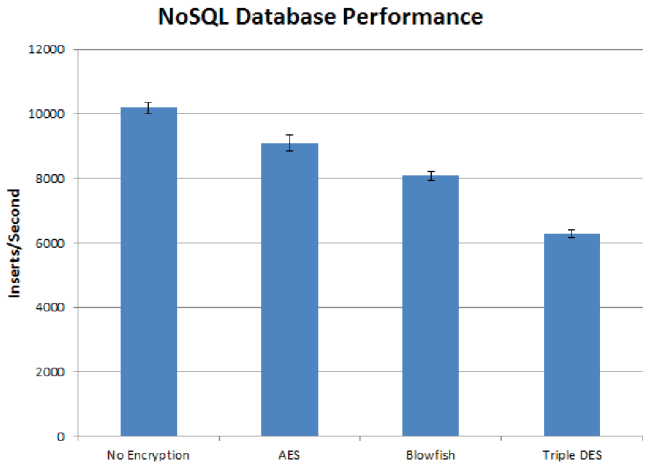

Fig. 3. Database Performance

1) AES Encryption: Advanced Encryption Standard (AES) was developed in 1997 by NIST (National Institute of Standards and Technology) to replace the outdated DES algorithms; which were becoming more vulnerable to brute-force attacks [15]. The AES uses two symmetric keys to encrypt and decrypt data (this means the key to encrypt the data will also be used to decrypt the data.) The keys in this algorithm can vary from 128 - 256 bytes. AES encryption is so secure that in 2003 the United States Government announced that AES encryption could be used to protect classified information [15].

2) Blowfish: The Blowfish algorithm was developed by Bruce Schneier in 1993 and has been examined by many cryptographers. Bruce developed Blowfish to be a free alternative to the existing algorithms at the time. Blowfish uses encryption keys with lengths up to 448-bits and currently no known attack has been successful against the Blowfish algorithm. Blowfish is currently not patented, license-free and has become one of the most commonly used algorithms [17]. Blowfish has a 64-bit block cipher size and a key length of anywhere from 32 bits to 448 bits. Blowfish has been subject to a significant amount of cryptanalysis, and full Blowfish encryption has never been broken [10].

3) 3DES: Data Encryption Standard (DES) is a symmetric key encryption algorithm originally developed in the early 1970's and was the standard for encryption for many years. It used a 56 bit encryption key with a block size of 64 bits. In its time it was considered a very secure algorithm, but with the advancements in computer hardware its small 56 bit encryption key left it vulnerable to brute-force attacks. In the late 1990s 3DES was introduced as a temporary alternative to DES while a new standard was developed. 3DES is essentially the DES algorithm repeated 3 times. This increases the length of the of the key to 168 bits (3 * 56). Many people are suspicious

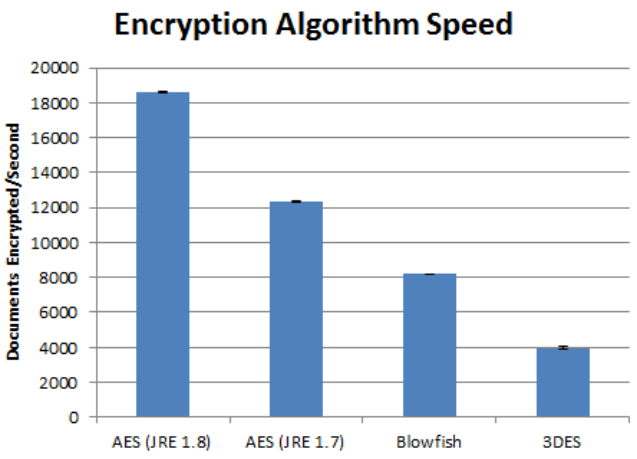

Fig. 4. Encryption Algorithm Performance

of 3DES because the DES algorithm was not designed to be used in this way, but no serious flaws have been found in its design, and it is still a widely used algorithm [8].

B. Encryption Algorithm Performance

We have implemented the previous three algorithms into the system and ran tests on them to see which one would be the best choice for us to implement in our system. In Fig. 4 we ran each encryption algorithm for a set amount of time to see how many 2491 Byte Json documents each of them could encrypt per second.

Theoretically Blowfish is a faster algorithm than AES, so we were surprised when tests were showing that AES more than twice as fast as Blowfish. The answer to this question of why AES is faster that Blowfish is optimization. In recent versions of Java there has been a lot of work done to optimize their implementation of AES and make use of Intel®'s AES-NI (AES New Instruction set); a set of instructions used to do AES encryption directly on the hardware which was introduced in 2010 [2] [19].

VI. Conclusion

In this paper we looked at how to generate data to emulate biometric sensors and investigated the effectiveness of different data encryptions for NoSQL document data bases for location and biometric data captured by sports wearable devices. Choosing an encryption method to use can be difficult, however through this research we have discovered 2 encryption methods that work well. The AES and blowfish algorithms seem to be the best choice for the system implemented. Blowfish can be implemented to be more secure than AES, however AES is faster when encrypting very large amounts of data, especially when using Intel® AES-IN. They outperform 3DES in both speed and security, 3DES is an outdated algorithm, and should not be implemented in new systems.

Acknowledgments

The research paper is based on the SW engineering spike project, which was developed by Computer Science students at Okanagan College within capstone project course COSC 470 "SW Engineering" in the Fall 2015, Bachelor of Information Systems (BCIS) program. The student project was in support to the NSERC CCI Engage College grant "GAUGE: Exact Positioning Systems For Sport and Healthcare Industries" (GAUGE). Our thanks to the COSC 470 students: Ahmed Abu Tayrah, Mohammed Aldaej, Khalid Almutiri, Yasir Asiri, Cheng-Kao Chiang, David Leader, Jon Ohlhauser and to the student research assistant Mikiko Koga for their participation in development and testing of the software applications.

We would like to thank NSERC CCI Engage College program of Canada for supporting our GAUGE research project application in 2015. Our thank to Amazon Web Services, Inc. for supporting our capstone student software engineering and our student research projects by AWS Grants for Research and Education.

References

[1] Paarijaat Aditya, Viktor Erdélyi, Matthew Lentz, Elaine Shi, Bobby Bhattacharjee, and Peter Druschel. Encore: Private, context-based communication for mobile social apps. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pages 135–148, New York, NY, USA, 2014. ACM.

[2] Intel Corporation. Aleksey Ignatenko. Improved advanced encryption standard (aes) crypto performance on java with nss using intel® aes-ni instructions. https://software.intel.com/en-us/articles/improved-advanced-encryption-standard-aes-crypto-performance-on-java-with-nss-using-intel.

[3] Mira Alhonsuo, Jenni Hapuli, Lasse Virtanen, Ashley Colley, and Jonna Hakkila. Concepting wearables for ice-hockey youth. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '15, pages 944–946, New York, NY, USA, 2015. ACM.

[4] M. Bellare. Block ciphers. http://cseweb.ucsd.edu/mihir/cse107/w-bc.pdf.

[5] Daniel Burmeister, Andreas Schrader, and Darren Carlson. A modular framework for ambient health monitoring. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '13, pages 401–404, ICST, Brussels, Belgium, Belgium, 2013. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[6] Hua Deng, Qianhong Wu, Bo Qin, Willy Susilo, Joseph Liu, and Wenchang 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*, ASIA CCS '15, pages 393–404, New York, NY, USA, 2015. ACM.

[7] Michael Frank, Meikel Poess, and Tilmann Rabl. Efficient update data generation for dbms benchmarks. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, ICPE '12, pages 169–180, New York, NY, USA, 2012. ACM.

[8] A.A.Zaidan Hamid A.Jalab M.Shabbir Hamdan.O.Alanazi, B.B.Zaidan and Y. Al-Nabhani. New comparative study between des, 3des and aes within nine factors. *Journal of Computing*, 2, 2010.

[9] Seema Holla and Praveen Dala-Krishna. Medical data encryption for communication over a vulnerable system, September 4 2012. US Patent 8,260,709.

[10] SplashData Inc. Blowfish encryption. http://www.splashdata.com/splashid/blowfish.htm.

[11] Florian Kerschbaum. Searching over encrypted data in cloud systems. In *Proceedings of the 18th ACM Symposium on Access Control Models and Technologies*, SACMAT '13, pages 87–88, New York, NY, USA, 2013. ACM.

[12] John Klein, Ian Gorton, Neil Ernst, Patrick Donohoe, Kim Pham, and Chrisjan Matser. Performance evaluation of nosql databases: A case study. In *Proceedings of the 1st Workshop on Performance Analysis of*

Big Data Systems, PABS '15, pages 5–10, New York, NY, USA, 2015. ACM.

[13] S. Marchesani, L. Pomante, F. Santucci, and M. Pugliese. A cryptographic scheme for real-world wireless sensor networks applications. In *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, ICCPS '13, pages 249–249, New York, NY, USA, 2013. ACM.

[14] Er Ashima Pansotra and Er Simar Preet. Singh. Cloud security algorithms. *International Journal of Security and Its Applications.*, Vol. 9(No.10):pp. 353–360, 2015.

[15] Margaret Rouse. Advanced encryption standard (aes) definition. http://searchsecurity.techtarget.com/definition/advanced-encryption-standard.

[16] Jiri Schindler. Profiling and analyzing the i/o performance of nosql dbs. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '13, pages 389–390, New York, NY, USA, 2013. ACM.

[17] Bruce Schneier. The blowfish encryption algorithm https://www.schneier.com/blowfish.html.

[18] Darpan D Shah, Anamika Mittal, and Kuntesh K Jani. A new approach towards encryption technique: D's crypto-cipher technique (dcct). *Advances in Computer Science and Information Technology (ACSIT)*, 2(5):pp. 446–449, April-June 2015.

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

[20] A Al Tamimi. Performance analysis of data encryption algorithms. *Retrieved October*, 1, 2008.

[21] Jakob Tholander and Stina 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*, CHI '15, pages 2913–2922, New York, NY, USA, 2015. ACM.

[22] Wouter Walmink, Danielle Wilde, and Florian 'Floyd' 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*, TEI '14, pages 97–104, New York, NY, USA, 2013. ACM.