The Scalability and the Strategy for EMR Database Encryption Techniques

David Shin, Tony Sahama, Steve (Jung Tae) Kim, and Ji-hong Kim, *Member, KIMICS*

***Abstract*— EMR(Electronic Medical Record) is an emerging technology that is highly-blended between non-IT and IT area. One of methodology to link non-IT and IT area is to construct databases. Nowadays, it supports before and after-treatment for patients and should satisfy all stakeholders such as practitioners, nurses, researchers, administrators and financial department and so on. In accordance with the database maintenance, DAS (Data as Service) model is one solution for outsourcing. However, there are some scalability and strategy issues when we need to plan to use DAS model properly. We constructed three kinds of databases such as plain-text, MS built-in encryption which is in-house model and custom AES (Advanced Encryption Standard) – DAS model scaling from 5K to 2560K records. To perform custom AES-DAS better, we also devised Bucket Index using Bloom Filter. The simulation showed the response times arithmetically increased in the beginning but after a certain threshold, exponentially increased in the end. In conclusion, if the database model is close to in-house model, then vendor technology is a good way to perform and get query response times in a consistent manner. If the model is DAS model, it is easy to outsource the database, however, some technique like Bucket Index enhances its utilization. To get faster query response times, designing database such as consideration of the field type is also important. This study suggests cloud computing would be a next DAS model to satisfy the scalability and the security issues.**

***Index Terms*— Data Encryption, Electronic Health Record, DAS, Bucket Index, Bloom Filter.**

I. INTRODUCTION

EMR is an emerging technology that is highly-blended between non-IT and IT area. The medical area now depends on and utilizes all IT technology from network to software. In addition, governments nowadays assures that IT area can help medical area and tries to amend the acts related to the medical area to conform with IT area more suitably. By using EMR consistently and seamlessly, many stakeholders such as practitioners, nurses, researchers, administrators and financial departments can improve their performances and make synergies which link all the stakeholders systematically. Despite successful application of information technology in other information-intensive industries, the current situation to share information across systems and between care organizations encounters many obstacles with efficiency and cost-effectiveness in health care (Grimson, et al., 2009)[1]. To support this methodology to deliver all the needs successfully, there needs to have a system that holds all the status for all stakeholders. Conventional technology to carry out this methodology is to construct databases. Nowadays, it supports before and after-treatment for patients. In the future, the information stored in one hospital can be sent out to another hospital electronically without hastiness. As for the financial perspective, EMR itself saves a lot of money to conduct all the necessary tasks compared to manual hand-writing and by human to human. To save more financial budgets, outsourcing is a solution to that. However, if the database can be outsourced to a third-party organization, we should consider all the security levels.

II. CONCEPTS OF THE SERVICE MODEL

A. DAS model and Query Processor

Wei et al. [2] mentions that the DAS model is a new data management model that allows users to outsource their data to database service providers (DSP). Since data is stored in cryptographic form at DSP, query efficiency becomes a critical problem. Existing solutions for this problem concentrate mostly on cryptograph index technology. The outsourcing database to a third party aims at decreasing the cost of maintenance of DBMS.

B. Bucket Index

The Bucket index is identical and useful for character type data. The construction of the index should follow two principles; firstly the index should filter false records efficiently, and secondly it should be safe enough not to leak (expose) the true value. Numeric data needs equations and a range query with “between”, “and” terms. An index supporting all computations does not exist, thus, it is possible to create different types of indexes according to the data type and their purposes. The index tries to translate the character string into numeric data on which the primary query will be processed to filter the records

Manuscript received July 24, 2011; revised August 1, 2011; accepted August 11, 2011.

David Shin and Tony Sahama are with the Computer Science Discipline, Faculty of Science and Technology (FaST), QUT, Australia

Steve Kim is with the Dept. of Electronic Eng., Mokwon University, Korea

Ji-hong Kim is with the Dept. of Info.& Com. Eng., Semyung University, Jecheon, Chungbuk, 390-711, Korea (Email : jhkim@semyung.ac.kr)

roughly. Only the rest of the records need to be decrypted and it will save a considerable amount of time.

C. Bloom filter algorithm

Zhong et al. [3] demonstrated that bloom filters are compact set representations that support set membership queries with small, one-sided error probabilities. Standard bloom filters only support elemental insertions and membership queries. The Bloom filter is used to speed up answers in a key-value storage system.

D. Secured Electronic Medical Records (EMR) Requirements and Design under the DAS model

According to Essin & Lincoln[4], the EMRs should have certain requirements such as atomicity, authenticity, persistence, flexibility of representation and retrieval, semantic integrity, interoperability, process ability, performance and security. In the context of security, there are legal and ethical requirements that the records should be kept secure and confidential so that each individual's privacy is preserved.

III. MAIN SIMULATION DESIGN

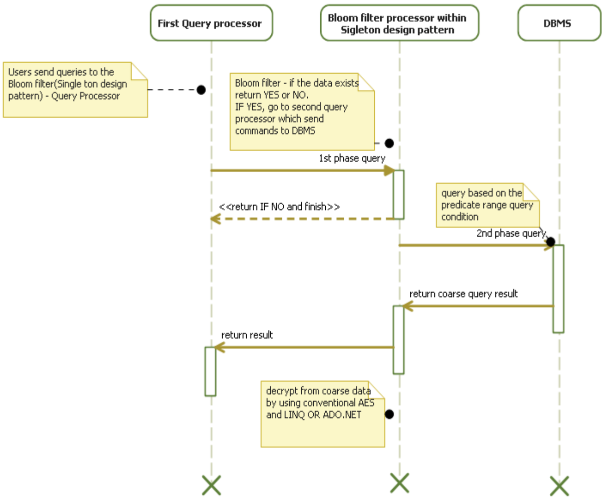
First Query processor
Bloom filter processor within Sigleton design pattern
DBMS
Users send queries to the Bloom filter(Single ton design pattern) - Query Processor
Bloom filter - if the data exists return YES or NO. IF YES, go to second query processor which send commands to DBMS
1st phase query
<<return IF NO and finish>>
query based on the predicate range query condition
2nd phase query
return coarse query result
return result
decrypt from coarse data by using conventional AES and LINQ OR ADO.NET

Fig 1. Sequence Diagram for Query Flow in Bucket Index using Bloom filter under AES-DAS model

The main idea behind the bucket indexing engine is to include key information such as range query conditions that has already been physically partitioned in tables. It also initially loads bucket key information into Bloom filter as a Singleton design pattern as shown in Figure 1.

TABLE 1.
LIST OF THE DATABASES FOR THE SCALABILITY OF BULK INSERTION

Database name	Total(Hrs)
Plain_Hospital_5	0.01
Enc_Hospital_5	0.09
Enc_Hospital_BI_5	0.01
Plain_Hospital_10	0.01
Enc_Hospital_10	0.18
Enc_Hospital_BI_10	0.01
Plain_Hospital_20	0.02
Enc_Hospital_20	0.34
Enc_Hospital_BI_20	0.04
Plain_Hospital_40	0.08
Enc_Hospital_40	0.72
Enc_Hospital_BI_40	0.08
Plain_Hospital_80	0.23
Enc_Hospital_80	1.44
Enc_Hospital_BI_80	0.20
Plain_Hospital_160	0.44
Enc_Hospital_160	2.83
Enc_Hospital_BI_160	0.48
Plain_Hospital_320	1.03
Enc_Hospital_320	6.03
Enc_Hospital_BI_320	1.07
Plain_Hospital_640	1.87
Enc_Hospital_640	11.35
Enc_Hospital_BI_640	2.10
Plain_Hospital_1280	4.13
Enc_Hospital_1280	24.85
Enc_Hospital_BI_1280	4.95
Plain_Hospital_2560	10.53
Enc_Hospital_2560	49.18
Enc_Hospital_BI_2560	14.28

Table 1 depicts the lists of the databases for the scalability of bulk insertion. The colored part is the DAS model using BI (Bucket Index) and the numbers after "Enc_Hospital_BI_" means K records for the size of the databases. The DBMS and the techniques for the implementation are Microsoft SQL Server 2008 and .net framework.

TABLE 2.
DATABASE ENCRYPTION MODELS REGARDING DAS

Kind	Encryption?	DAS model?
Plain-Text	No encryption	Not recommended to DAS
MS built-in	Combination of Master database key, Triple DES, Asymmetric key	Can be recommended to DAS, however, is the database administrator trustworthy?
AES-DAS	Advance Encryption Standard	Recommended to DAS, all encryption / decryption logic is on the application side not on the database side

The simulation environment is below.

TABLE 3.
SIMULATION ENVIRONMENT

Development tool	Database	H/W & OS
MS Visual Studio 2010 (C#)	MS SQL Server 2008	Core 2 duo, Windows XP

The scalability was tested from 5K records to 2560K records. The table below shows each kind of databases, its size and time to build up. It shows AES-DAS shows much faster bulk insertion than MS built-in.

IV. SIMULATION RESULTS AND ANALYSES

A. Main Pseudo Logic

TABLE 4.
STAGE1 SIMULATION – PSEUDO CODE & ANALYSIS

AES-DAS	Process n:10,000 records, # of partition: 70 partitions, R: the record satisfying the partition & predicates, DECPartitionFunc():the bucket index manipulated by Vigenere cipher Bloom filter: in a Singleton design application heap	Coefficients
Range	#of Qurey: 1 n* decryption	10,000 * decryption = 156ms

Semi-Bucket	# of Query: 1 or 2
	IF R <= unit of partition
	SELECT the Ranged data
	(n/70) * decryption
	ELSE IF R > unit of partition
	(n*# of used partition / 70) * decryption
	ELSE
	Consume 1^{st} query and return FINISH
		R: Range
		Best Case:
		(10,000/70) * decryption = 16 ms

Bucket	# of Query : 1 or 2
	IF R <=
	DECPartitionFunc(unit of partition)
	SELECT the Ranged data
	(n/70) * decryption
	ELSE IF R > unit of partition
	SELECT the Ranged data
	(n*# of usedpartition/70)* decryption
	ELSE
	Consume 1^{st} query and return
		R: Range
		DECPartitionFunc(): bit operation || Vigenere cipher
		Best Case :
		(10,000/70) * decryption=(<= 47 ms)

Bucket Including Bloom filter	# of Query : 0 or 1
	IF BLOOM FILTER Contains
	SELECT DECPartitionFunc (unit of partition)
	(n* # of used partition/70) * Decryption
	ELSE
	Return
		BLOOM FILTER : in a Singleton designed application heap
		DECPartitionFunc(): bit operation || Vigenere cipher
		Best Case:
		(10,000/70) * decryption=(<=47ms)

Table 4 shows the detailed algorithm comparison for the encrypted range query which is efficient under DAS model. For simulation, the Bloom filter as a singleton design pattern in a Bucket Index saves database connection time in the case of checking the wanted data existence. In addition, for the security in the database, the Bucket Index as a field in a table in DBMS can be presented as a bit strings manipulated by a simple Vigenre cipher.

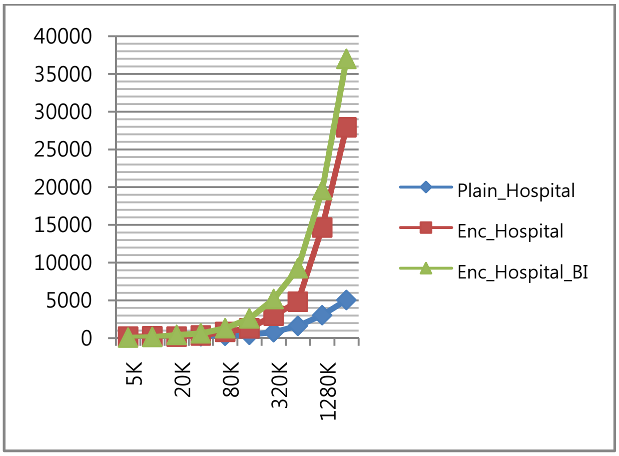

Fig. 2. Range select query scalability test

Figure 2 depicts Range select query scalability test. When the record size is increased from 5K to 2560K, the processing times are drawn noticeably in Enc_Hospital (MS built-in encryption scheme) and Enc_Hospital_BI (Bucket Index) whereas Plain_Hospital (non-encrypted) increases in arithmetic scale.

The Y axis stands for processing time (milliseconds). In this scalability, the results show that the MS built-in encryption scheme is slightly better than AES-DAS model. Thus, the MS built-in encryption technique can be useful when the data is valuable as a statistically compared with AES-DAS model. However, in this simulation of AES-DAS model, using Bucket index with singleton designed Bloom filter shows faster results than just using the Bucket Index field only.

B. Main Scalability Test Results

Followed by query optimization approach we conducted an experiment about the database scalability and query response times by using 6 different types of 'select queries'. These 6 types of queries are for string, numeric, aggregate, normal range, normal range-count, and range using Bucket index respectively. There are 3 types of databases (e.g., PT, MSE and DAS) been utilized. Plain-Text (PT) which has no encryption/decryption functionality. Microsoft built-in Encryption (MSE) which is an in-house model with the encryption/decryption logic built in already. The DAS can be an outsourcing model which allows encryption/decryption logic can be outside the database.

TABLE 5.
CATEGORISED SCALABILITY TEST RESULTS
(IN MILLE-SECONDS)

DB	Record Size	String	Numeric	Aggr.	RQ	RBI
PT	5K	73	0	3	186	N/A
PT	10K	100	0	6	233	N/A
PT	20K	160	0	10	263	N/A
PT	40K	283	20	36	246	N/A
PT	80K	293	13	123	313	N/A
PT	160K	493	30	153	453	N/A
PT	320K	1030	16	276	793	N/A
PT	640K	1753	36	476	1633	N/A
PT	1280K	4236	100	820	3040	N/A
PT	2560K	7426	50	1983	5060	N/A
MSE	5K	106	16	70	200	N/A
MSE	10K	183	23	56	240	N/A
MSE	20K	316	46	60	210	N/A
MSE	40K	570	100	116	350	N/A
MSE	80K	2543	200	333	830	N/A
MSE	160K	13453	410	576	1306	N/A
MSE	320K	40806	796	1220	2970	N/A
MSE	640K	136600	1656	2423	4830	N/A
MSE	1280K	330713	3193	4686	14656	N/A
MSE	2560K	718150	26580	8906	27936	N/A
DAS	5K	47	16	31	63	16
DAS	10K	266	94	63	156	16
DAS	20K	344	188	250	422	109
DAS	40K	1109	234	438	641	203
DAS	80K	3813	344	968	1313	297
DAS	160K	12110	453	1328	2610	453
DAS	320K	40690	656	2235	5172	828
DAS	640K	111585	938	2672	9266	1531
DAS	1280K	289345	1203	5344	19595	2781
DAS	2560K	660851	1875	8704	36987	5226

Practically, select queries are more heavily used than insert or delete queries. Therefore, we had the scalability simulation from 5K record-size to 2560K record-size to measure select queries respectively.

Table 5 represents columns as the categories of String type, Numeric type, Aggregation query (Aggr.), Range query (RQ) and Range query using Bucket Index (RBI) which was applied only for AES-DAS model (i.e.: DAS DB). The databases with PT (Plain-Text) and MSE (MS built-in Encryption) resulted in null values (e.g., N/A – Not Available) in RBI query type were a compromise to the database encryption technique or a default strategy for a DAS model used. Furthermore, existence of the RBI (Range query using Bucket Index), is in the encryption/decryption modules that are in the application side not inside the database.

C. Simulation Analyses using R

Fig 3 shows the distribution of range query response time data in DAS model by depicting the distribution between range query, range count query, and Bucket Index range query respectively. The Size denotes the database sizes used for the simulation experiment.

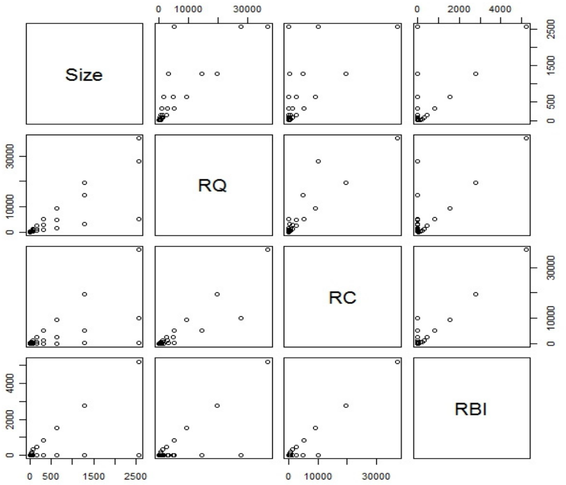

Fig. 3. Distribution of Range Query Data in DAS model

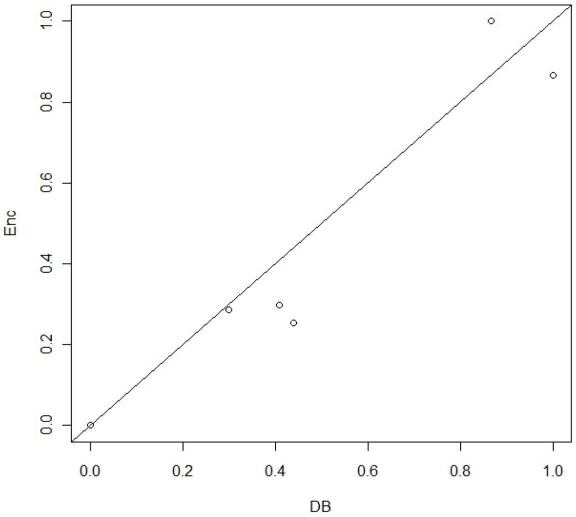

Fig. 4. Correlation of Databases, Encryption & Range query types

Fig 4 depicts correlation of database types, encryption and range query types. QRange denotes normal Range Query. BRange represents Bucket Index Range Query while RCount depicts Range Count Query. Both Sizes and normal Range Query have consistent with the encryption process and the database sizes.

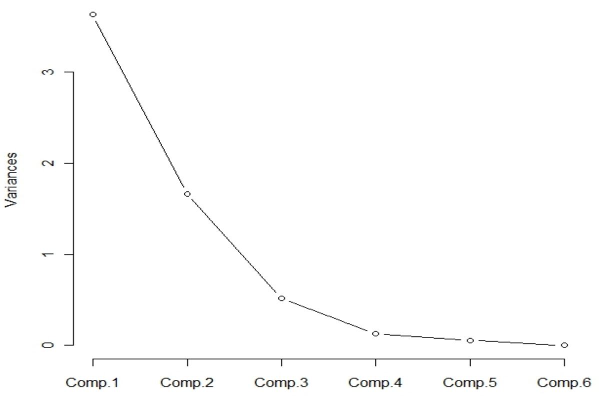

Fig. 5. Variance among components in DAS model

To understand the affect of database sizes and query types we performed Principal Component Analysis. The X-axis in Fig 5 above, depicting from Comp.1 to Comp.6 is the database types such as plain-text, MS built-in encryption and AES-DAS model. Furthermore, Comp.2 represents the encryption types such as “No” or “Yes”. Comp.3 consist of the size of the databases from 5K to 2560K records. Comp.4 is the normal range queries. Comp.5 represents the counts from the range queries. Comp.6 represents the range queries using Bucket Index. It is not unusual to have lower variances among the Bucket Index queries. The variance analysis from principal components shows the little variance among the other components and the variance became smaller when we use Comp.6 (Bucket Index Range).

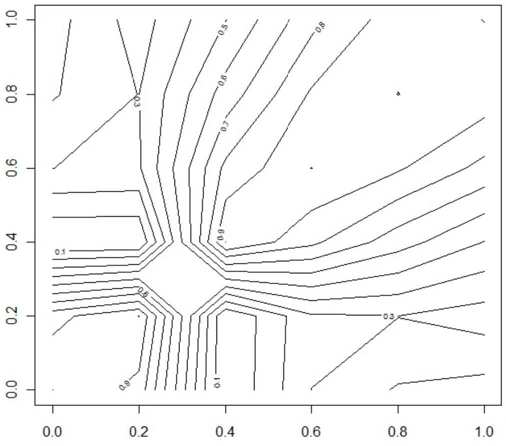

Fig. 6. Contour of the correlation for the simulation

The Bucket Index component represents positive correlation compared to other methods used. Fig 6 depicts the contour of the correlation of the simulation conducted for Bucket Index embedded queries. The relationships between Bucket Index and query types used were positively correlated however it is not clear whether the database sizes have been contributed to this correlation affect.

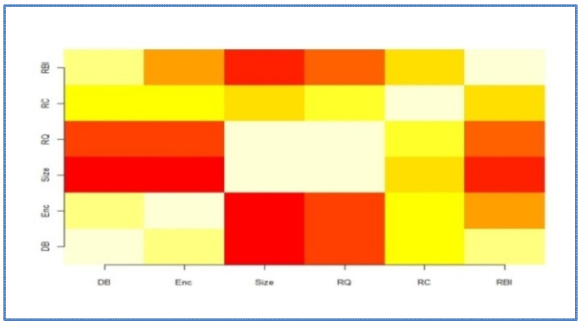

Fig. 7. Relational Image Map for Range Query Simulation

We conducted a correlation density analysis of the Bucket Index queries in order to understand the affect of database sizes. The results presented in Fig 7, depicts that the RC (Range count query) is not much related to other categories such as DB types and Encryption types. However, RQ (Range query) is much related to DB types and Encryption types. Furthermore, RBI (Range query using Bucket Index) is much related to the Size of the databases in long run. While we established the relationship between correlation densities of the query types the analysis is somewhat cloudy.

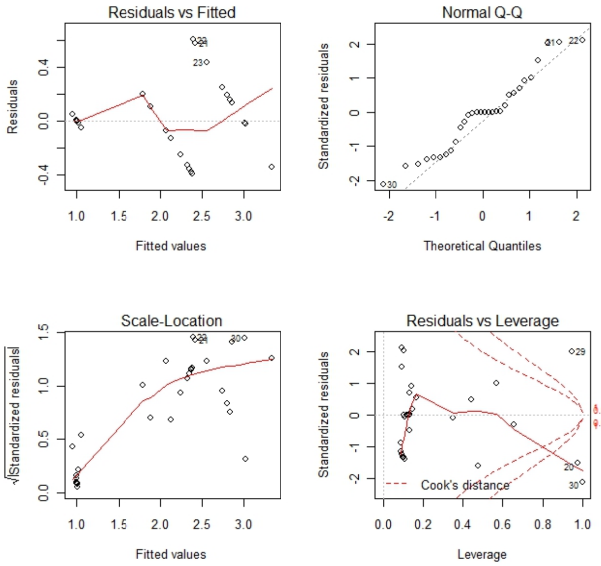

Fig. 8. Analysis of Variance (ANOVA) for the simulation

To understand the relationship better we examined the residuals of the variables utilized. Fig 8 depicts the diagnostic plots. In order to compare the affect of the database sizes, it is apparent that size of the database requires increasing with replications. This approach would be an extension of this simulation experiment should be addressed in future experiments.

Through this scalability simulation on EMR databases, we found some rules and reported below. This result is related to PC and MS SQL Server environment; however, this will be a starting point to design large scaled EMR databases with or without encryptions using a DAS (Data as Service) model.

The outcome from the scalability test with different categories is as follows.

- Numeric type shows faster select query response time than Ch. type as the size of the database become bigger up to 2560K records and encrypted (Numeric data: Medicare card number, Ch. sample: Suburbs name in address)
- In retrieving a record for simple Ch. type or numeric type, custom AES-DAS is slightly better than MS built-in encryption in SQL Server
- In retrieving range queries, MS built-in encryption is slightly better than custom AES-DAS model.
- In range queries under the AES-DAS models, the Bucket Index technique shows 7 times faster response than a normal Range Query when they are scaled up to 2560K records and the number of Bucket partitioning was 70 (date time: birth date)
- In aggregate queries, the custom AES-DAS model shows inconsistent query response time compared to MS built-in encryption as the size of the database become scaled up to 2560K record (blood type)

V. CONCLUSION

We presented the scalability ranges from 5K to 2560K record-sized. In the simulation experiment, the results are usually incremental arithmetically according to the record-size. However, there are threshold points in each category – string, numeric, range and aggregate queries. As an example, in MS built-in encryption, from 1280K record-sized to 2560K record-sized numeric query, the response time increases suddenly in a large step whereas in the case of AES-DAS model, the response time increases also exponentially after reaching a certain threshold. In the category of normal range operation, MS built-in encryption is slightly faster than AES-DAS model. However, MS built-in encryption is not a proper AES-DAS model. Among the presented AES-DAS models above, the Bucket Index model including Bloom filter can save unnecessary connection time and its bucket is operated more securely compared to Semi-Bucket models and so on. For future research experiment, the "cloud computing" environment can be considered for e-health decision-making systems to share information freely because the scalability and the security issues are handled as a top priority. Moreover, data- warehousing and web services among the e-health database providers can maximise the e-health data communication. Further research on those environments is useful.

ACKNOWLEDGMENT

This research presented in this paper is supported by the Ministry of Education and Science Technology of Korea (No. 2010-0023648).

REFERENCES

[1] Grimson J, Grimson, W, and Hasselbring W. The SI Challenge in Health Care. Communications of the ACM 2009: 43(6): 49-55.

[2] Wei, Z., et al. *A tuple-oriented bucket partition index with minimum weighted mean of interferential numbers for DAS models*. in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*. 2010.

[3] Zhong, M., et al., *Optimizing data popularity conscious bloom filters*, in *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*. 2008, ACM: Toronto, Canada. p. 355-364.

[4] Essin, D.J. and T.L. Lincoln, *Healthcare information architecture: elements of a new paradigm*, in *Proceedings of the 1994 workshop on New security paradigms*. 1994, IEEE Computer Society Press: Little Compton, Rhode Island, United States. p. 32-41.

David Shin is a Research Master student in computer science discipline, faculty of science and technology, Queensland University of Technology. He received the Masters' degree in Computer Science from QUT in 2009. His interest includes database modelling and simulations, developing mobile and cloud computing applications in e-health and gaming platforms.

Tony Sahama is a Senior Lecturer at the School of Information Technology, Faculty of Science and Technology, Queensland University of Technology. He received the PhD degree in Computer Science from Victoria University, Melbourne in 1999. His interest includes Computer Experiments, Modeling and simulation, Medical Informatics and Information Technology application in Health care.

Jung-Tae Kim received his Ph.D. degrees in Electronic Engineering from the Yonsei University in 2001. From 1991 to 1996, he joined at ETRI, where he worked as senior member of technical staff. In 2002, he joined the department of electronic engineering, Mokwon University, Korea, where he is presently professor. His research interest is in the area of information optical security technology that includes network security system design, RFID&USN and wireless security protocol.

Ji - Hong Kim is a professor at the Department of Information and Communication, University of Semyung. He received the M.S. and Ph.D. degrees in Electronic Communication engineering from Hanyang University in 1984 and 1996, respectively. His interests include PKI, Database security, and cryptographic applications.