Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

COMPARISON OF APPROACHES TO ENCRYPT DATA FOR SUPPLY CHAIN SIMULATION APPLICATIONS IN CLOUD ENVIRONMENTS

Kai Gutenschwager
Marcel Theile
Bastian Wilhelm

Ostfalia University of Applied Sciences
Hochschule Braunschweig/Wolfenbüttel
Am Exer 2
Wolfenbüttel, 38302, GERMANY

Markus Rabe

Department IT in Production and Logistics
TU Dortmund University
Leonhard-Euler-Str. 5
Dortmund, 44227, GERMANY

ABSTRACT

A main characteristic in the field of supply chain simulation is that typically several independent organizations are involved. However, most simulation studies only consider a very limited number of different organizations. One main reason is that suppliers usually do not want to publish sensitive data, such as resource capacities or cost rates to their customer. In this paper we present an overall architecture to tackle this problem of multi-organizational simulation studies storing all relevant data at an independent third party, which also carries out all experiments and distributes the results as a cloud service. In order to define scenarios, different rights to use the provided data can be granted to other participants. The necessary user-specific encryption of datasets needs to be established on the underlying data base structure. In this paper we focus on three approaches for storing and encrypting data along with a detailed comparison in terms of performance.

1 INTRODUCTION

Supply chain simulation is typically highly data-driven covering both the information and the material flow between different locations, such as customers, warehouses, production sites, hubs and suppliers in overall networks. Within the field of supply chain design, one focus is often to find good parameter settings for all operative processes, such as forecasting, production, inventory and distribution planning.

A major problem is that the data required for an overall view and optimization of supply chains, e.g. resource availability or cost rates, would be visible to at least the company that carries out the respective simulation study, referred to as the model owner. This is a fundamental challenge, as suppliers usually do not want to give insight into their sensitive data. However, useful results can still be obtained from such overall models even if key figures for costs and local resource utilization are not presented to the model owner as absolute values. It is often already sufficient to gain knowledge whether certain changes to the supply chain (as, e.g., increasing the safety stocks at a given location) will lead to an improvement of the overall service levels toward the final customers and to what degree costs will increase. For such studies, an independent service provider is necessary being responsible for data acquisition, modeling, experimentation and result presentation, such that all requirements concerning sensitive data are met. Some of these tasks could be automatized or still be taken care of by the model owner, using a cloud-based modeling and simulation environment, where all users (model owner and data providers) upload and validate their data themselves. In this approach, optimization and simulation models are set up as generic models using all data without

any user having access to all datasets at any time. However, in order to define (new) scenarios the model owner will need to have rights to use or to access the other users' data. Here, two different approaches are possible:

- A user (typically a data provider) grants access to the respective datasets to another user (typically the model owner), enabling the other user to make absolute parameter changes for new scenarios.
- A user grants access to the respective datasets to another user, enabling the other user to make relative parameter changes, e.g. increasing safety stocks for certain stock keeping units (skus) by 5 %.

To our knowledge, no commercial simulation tool exists addressing these points (see Section 2). Current IT solutions for simulation and optimization of logistic systems are typically installed locally by the model owner or a respective service provider, who is often not independent from the model owner, whereas all other companies serve as data providers for modeling the overall supply chain. The data collection with various companies involved is typically very time-consuming and cost-intensive, since it often means high manual effort. The necessary model validation is also rather complex, since, in case of inconsistencies or the occurrence of unexpected results, the respective data suppliers have to be consulted. The existing cloud-based solutions (Anylogistix and SimChain) offer a more structured handling of data, but do not differentiate users for a simulation project, i.e. different companies need to share an account to work on the same model.

This paper roots from research conducted in the SimChain MONSTER project. This project, targeting the optimization and simulation supply chains, has started in March 2017 and will last until February 2019. The work presented here covers the results of the first project year. It addresses some of these critical issues, intending to outline the development of a software tool to facilitate the analysis and simulation-based optimization of supply chains, in which several legally independent companies are involved. In this paper, we provide solutions for possible data models to address the given problems, which results in three primary goals:

- User data have to be encrypted, since the application will be used in a cloud environment with multiple users assigned to a single simulation project.
- Users should not be allowed to access the data of other users, if not given the explicit permission of the respective users, i.e. such rights need to be administrated by the tool.
- Every user should be able to grant access to another user on his own data, where the data owner can individually determine the rights for every other user.

The rest of this paper is organized as follows. Section 2 gives a short overview over simulation tools in the field of supply chain design. Section 3 briefly describes our data model and the general problem of data security regarding the storage of data. Section 4 presents the three concepts which have been developed and implemented to fulfill the given requirements. Section 5 shows the differences in terms of performance of the three concepts. Section 6 summarizes the test results and gives a short outlook for future research.

2 COMMERCIAL TOOLS

There are several vendors of discrete event simulation tools that originated from the field of intralogistic systems. Some of those tools offer the possibility to simulate supply chains, mostly by providing specific elements within class libraries. Examples are Arena (Rockwell Automation), Enterprise Dynamics (Incontrol), FlexSim (FlexSim Software Products), Tecnomatix Plant Simulation (Siemens) or Witness (Lanner Simulation Technology). Most of these tools do not offer a generic

modeling approach where all elements are defined in a database. Instead, each model object needs to be parameterized within the modeling environment itself. However, for importing data, standard interfaces of spreadsheet applications and database systems can be used in most tools.

A discrete event simulation tool for supply chain simulation is the IBM Supply Chain Analyzer (Bagchi et al. 1998; Archibald et al. 1999). The simulation library comprises, similar to other libraries, a set of classes like customers, manufacturing, distribution (warehouses and outlets), transportation, inventory planning and forecasting. Supply Net Simulator (SNS) is another specialized tool developed by Daimler (Stäblein et al. 2007). There are also some tools for supply chain modeling and analysis, usually originating in the field of analytical models and optimization, such as 4flow vista, Logistics Designer (LOCOM) and the Supply Chain Guru (LLamasoft), which offer discrete event simulation components. But, also these tools do not offer a generic modeling approach or multi-user functionalities. A relatively new tool is Anylogistix (XJ Technologies), which combines optimization and discrete event simulation based on Anylogic.

SimChain is a discrete event simulation tool, which has originally been developed as a class library for the simulation tool Plant Simulation (Gutenschwager and Alicke 2004). The data basis is a MySQL database, which is used to automatically generate the simulation model utilizing a SimChain-specific building block library and to store all result data. It has also been extended for environmental analysis (Rabe et al. 2013). A cloud-based implementation is available, but does not support multi-user modeling either.

The data model of SimChain consists of more than 60 tables. It allows for modeling a great variety of processes within a supply chain. Here, different nodes are distinguished in the simulation component, namely customers, production sites, distribution centers, hubs, and plain suppliers. The nodes are connected by direct transport relations or routes for modeling tours visiting more than one customer. Sourcing routes are defined as sequences of transport relations (including a given number of hubs) or routes to model multi-modal delivery strategies. Due to its flexible, data-driven structure, SimChain has been chosen as a basis for the data model described in this paper.

3 DATA MODEL

In this section we will describe the basic structure of our data model. The overall concept of the system with two different databases is given in Figure 1.

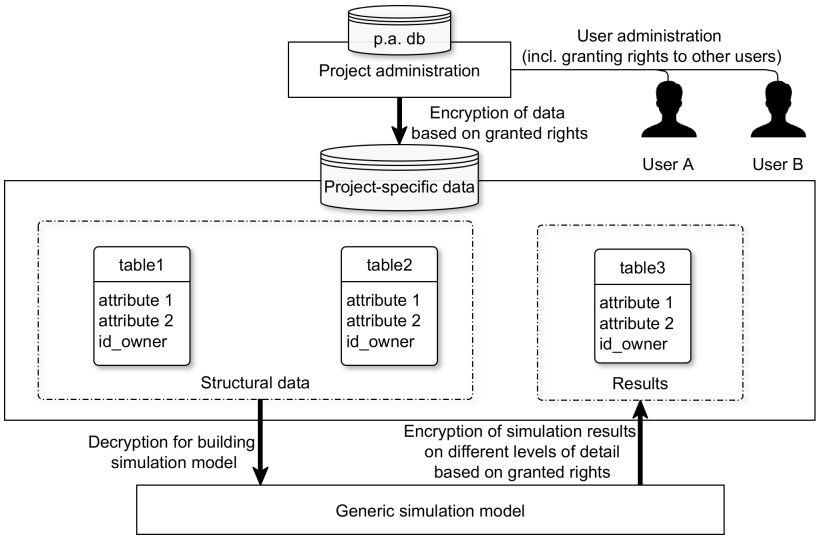

Figure 1: Overall system concept.

The *project administration database* stores all information about the users (such as name and password) as well as all granted rights to other users' data. The ownership of a dataset is defined by the additional column *id_owner* that identifies the user. Once a user has received access rights from another user, he can modify the respective datasets for defining new scenarios, according to the privileges he owns. When starting a simulation experiment, all necessary data will be decrypted, but will only be visible for the simulation engine itself. After the simulation experiment has ended, the results are encrypted, based on the granted rights, and will be inserted in the corresponding result tables. An excerpt of the project administration database is given in Figure 2.

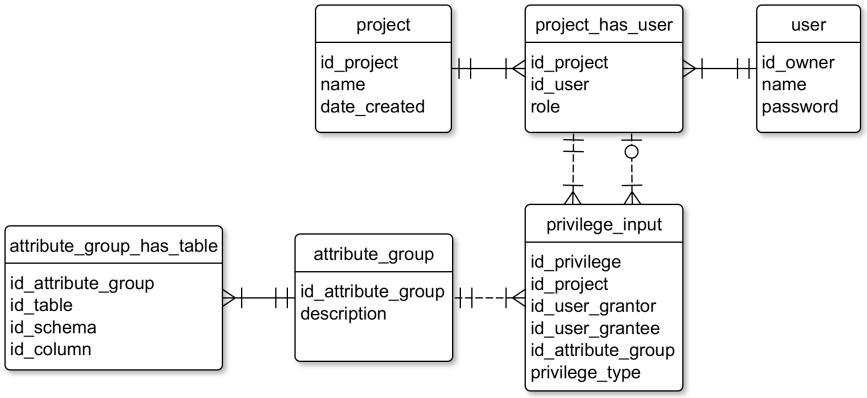

Figure 2: Table structure of the project administration (crow's foot notation).

Generally, a project consists of a group of users who share the same data, whereas setting up scenarios to be simulated is typically only conducted be the model owner. Every user is mapped to all projects he participates in. In each project a user receives a project-specific role. Within a single project a user can grant access rights to his datasets (organized as predefined *attribute groups*) to other users. Attribute groups are introduced to the data model as these represent logical units of attributes that belong to each other and should, therefore, not be handled differently with respect to access rights. Here, it is also more efficient and less error-prone to encrypt a whole set of columns in the same manner rather than encrypting each column individually. The access type a user owns for an attribute group is defined in the column *privilege_type* in the table *privilege_input*. There are three different types of privileges to be distinguished:

- absolute
- relative
- none

None is assumed to be the default value and ensures that only the owner (and no other member of a project) can access the respective datasets. This can be achieved as the simulation model is generic without giving any user the possibility to have insights of the internal model data. With privilege type *absolute*, a user can grant the right to freely access and modify copies of the respective datasets to another user without any restrictions, whereas *relative* grants the right to define modifications of the original data as a relative change without revealing the original data.

In this context, the application supports to create new scenarios on already existing ones. When creating a new scenario, a user needs to specify a base scenario from which the new scenario inherits all data. Basically, a user can only modify the data from the base scenario with the concept of so-called changesets, since the stored date of the base scenario must not be modified directly in any way by other users. The rights for the base scenario always remain at the original owner, and only

the changesets are assigned to the grantee (and are encrypted respectively). Because all changes that define the new scenario need to be persisted, respective changeset tables are defined in the schema for the project-specific data (cf. Figure 1). Figure 3 shows the general structure for a simplified example: The table *distribution_center_has_sku* has two different tables for defining changesets, which directly refer to the rights *absolute* and *relative*. The table for relative changesets does not contain all attributes of the corresponding base table, since only numeric attributes can be changed relatively.

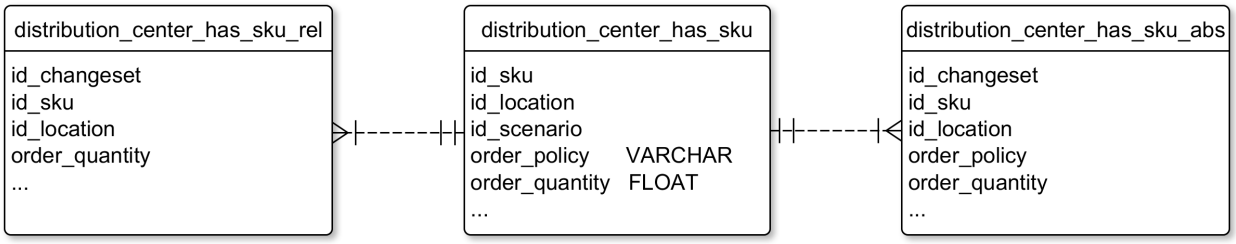

Figure 3: Concept of changesets.

The main challenge is to find an encryption policy that allows for granting rights based on the defined privilege types, encrypting the data accordingly and still being able to deliver a sufficient performance with respect to the runtime for encrypting and decrypting. As simulation runs themselves may need a relatively high amount of computation time, the additional computational times for data handling should be as low as possible.

4 ARCHITECTURE VARIANTS FOR STORAGE AND ENCRYPTION

Because our data model enforces to provide individual rights from one user to another based on a given attribute group, it is not possible to clearly identify a role for every user allowing to define a single access right of a user. Therefore, Role-Based Access Control (RBAC) models (Sandhu et al. 1996; Ferraiolo et al. 1995; Ferraiolo et al. 2001; Mönkeberg and Rakete 2000) are not suitable for our specific problem.

Another approach is to define so-called *encryption attributes*, that describe the rights a user is granted to access certain data. This concept differs from our attribute groups. The basic concept is called Attribute-Based Encryption (ABE) and is used in various fields of application (Bethencourt et al. 2007; Goyal et al. 2006; Roeckle et al. 2000). Following this approach, an encryption attribute needs to be defined for each combination of an attribute group and the two users involved. The ABE approach could, therefore, easily lead to a much higher number of encryption attributes than the number of attribute groups themselves. For reasons of efficiency, we have focused on three basic concepts, that only need to administrate user-to-user combinations:

- DB-based storage with a cell-based encryption
- DB-based storage with a table-based encryption
- File-based storage

These approaches have been selected as they have different advantages and disadvantages with respect to the necessary implementation effort to adjust the given data structure of SimChain, the flexibility to use different algorithms for encryption, and the computational runtime.

4.1 Cell-based Encryption

The first approach for safely storing data aims on encrypting cells of each database table individually. The basic concept of the encryption model is based on the models described by De Capitani Di Vimercati et al. (2010). Every user owns a pair of keys consisting of a private and a public key. If data shall only be visible to the user himself, then the data are encrypted with his own private key. Should a user grant access on his data to only one single other user, than these two users use the Diffie-Hellman (DH) key exchange (Diffie and Hellman 1976) to generate a shared secret key. With this key, the two users can encrypt data for one another. Apart from the DH keys, every other key is stored in a token. A token is basically a piece of information which has a source, a destination, and a so-called value. An encrypted key is stored in such a value. The source of a token describes which users are able to use this token, whereas the destination shows which users are able to access data encrypted with the derived key of the given token. Besides tokens there are resources. A resource consists of an id, an owner, a label and an encrypted piece of data. The owner defines to whom the stored piece of data belongs. The label is needed to derive which users are allowed to view the encrypted data. An example for a token graph is given in Figure 4.

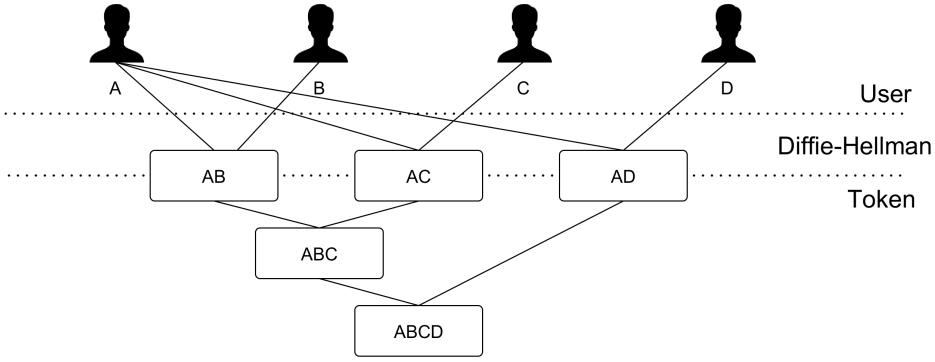

Figure 4: Example of a token graph for four users; cf. De Capitani Di Vimercati et al. (2010), p. 4.

This graph results from A granting B and C access to some of his data and to all users including D to some other data. In a first step, the DH keys for the two pairs (A,B) and (A,C) are created. Afterwards, a new key is generated that will be stored in a token only available to A, B and C. Next, A grants D access on some of his data, which are only a portion of the data available to B and C. Therefore, a DH key for those two users is computed. Then, a second secret key will be generated and stored in two new tokens, which are accessible from the key stored in the token of (A,B,C) and by the DH key of (A,D). The creation of tokens for more users proceeds in the same way, i.e. tokens are created once a user grants access to part of his data to another user, that leads to a key structure which can be represented as a graph.

Figure 5 illustrates the basic concept for users inserting their data via the GUI of the website. These datasets are sent to the database using REST calls (Richardson and Ruby 2008). Within the REST methods the encryption is carried out, based on the token graph that can be derived from the rights each user has been granted. Afterwards, the data are inserted into the corresponding tables of the database. Selection and decryption of data are analogue to this process. The tokens with all their information are encrypted and stored in files. For each token a separate file is created. These are decrypted once a user wants to upload data in several steps following the respective path within the token graph.

The basic structure of the original project-specific data model of SimChain only needs to be extended by a further column identifying the owner of each dataset, which is less modeling and implementation effort than for the other two approaches described next.

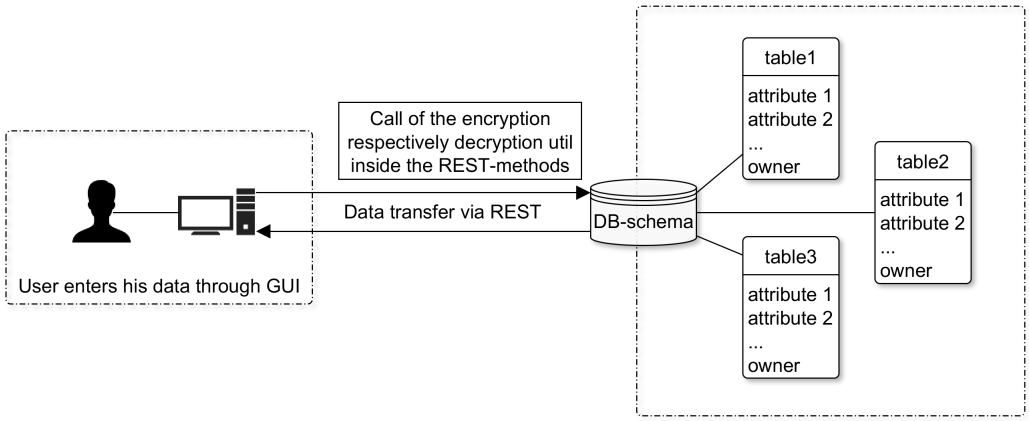

Figure 5: Basic concept of for data upload.

4.2 File-based Implementation

Our second approach is derived from the Data-Vault Model (Linstedt and Olschimke 2015; Casters et al. 2010). The main idea is to separate the individual data to be hidden from the other users in user-specific files. In order to identify all datasets uniquely, a relational database is used. Here, a table holding all primary and foreign keys is defined for each table of the original data model. In this paper, these are referred to as *basic tables* in the project-specific data base, which means there exists a number of tables only containing key attributes to maintain the basic structure of the original data model. The externally stored files contain all datasets and are mapped to the corresponding table containing the keys (Figure 6). Each file corresponds to an attribute group (defined in the database *project administration*).

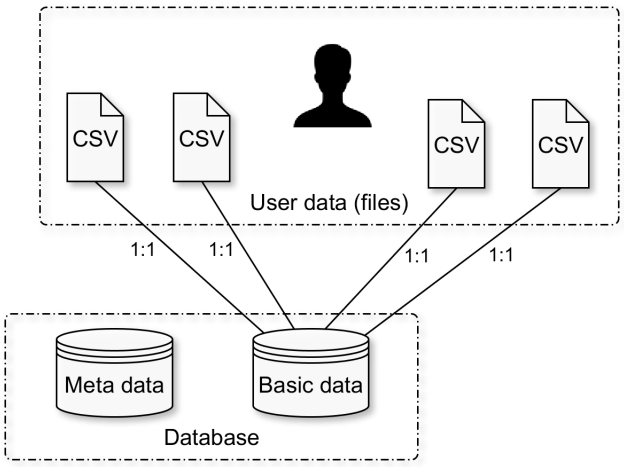

Figure 6: Concept of file-based encryption.

The idea behind the file-based approach is to safely store sensitive data in a cloud environment in greater blocks than then cell-based encryption. By encrypting the files containing data of the users, it is ensured that even if an unauthorized user gains access to the cloud he will most likely not be able to decrypt the data stored in the file. Because some data are not sensitive and publicly known to others, it is unnecessary to encrypt this type of data. The basic concept of our encryption algorithm uses cipher streams from Java to encrypt and decrypt the corresponding files. The file format is a modified version of the CSV format. The separators for lines and attributes had to be

adjusted to the unicode control character null (U+0000) and start of heading (U+0001) as attribute separator respectively line separator. There is no specific order of the datasets within a file, as these are simply appended with each insert. However, the attributes have a given order, always starting with the primary key attributes. The attribute order is defined in the meta data. The basic folder structure is defined as *user-home/schema-name/table-name/attribute-group-name*. Here, *user-home* describes the root path for each user. *Schema-name*, *table-name* and *attribute-group-name* are set depending on the corresponding name of the project (schema), tables and attribute groups. The folder *attribute-group* contains the encrypted data, where each file corresponds to an attribute group. The user management ensures that each user is able to grant access to his data (attribute groups and files). The basic structure of this concept is shown in Figure 6. Because the concept of attribute groups is not supported in standard databases, we defined an extension of the standard SQL dialect to handle such groups.

4.3 Database-based Implementation

The third approach is also based on the Data-Vault Model, but uses only a database for data storage and no additional files. *Basic tables* with all primary and foreign keys are defined according to the file-based implementation. For each basic table, a user-specific table with all other attributes of the respective entities is created within the project-specific database schema. Between the basic table and each user table a one-to-one dependency is defined. Figure 7 shows the basic layout of the resulting database schema. Each basic table stores all primary keys and foreign keys in order to keep the structure of the original data model. All data attributes of an entity that do not have to be encrypted are stored in another table with the public data. Tables holding public data are also assigned to the respective basic table.

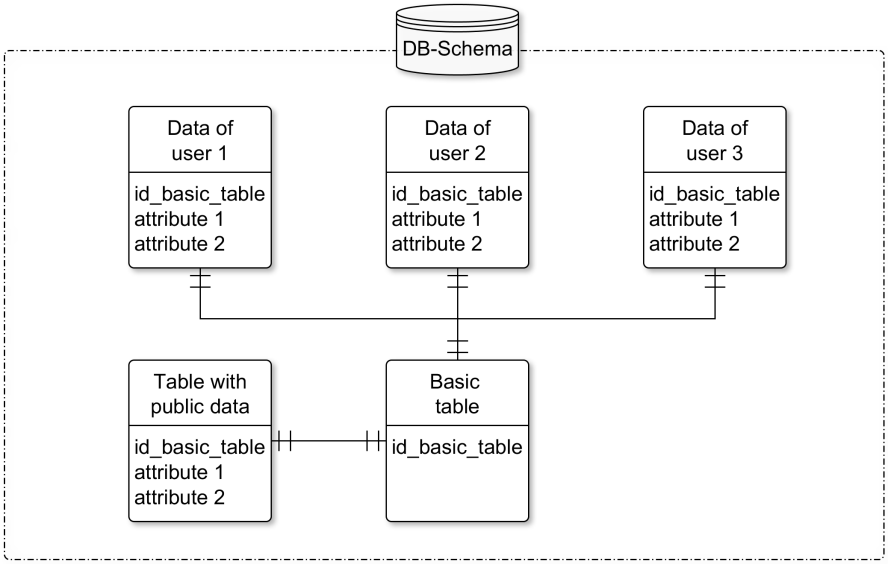

Figure 7: Concept of database-based encryption with five tables (crow's foot notation).

For granting permissions to other users, we use the default permission system of MySQL. This is possible because MySQL has the functionality to directly grant permissions on individual columns. Therefore, no further right management with respect to attribute groups needs to be implemented. To encrypt the data we use the MySQL *keyring_file* plug-in which uses one single key to encrypt all data, but this key is never used outside of MySQL. Hence, the safety of the key is dependent on the general security standard of MySQL, and the encryption algorithm cannot be replaced individually.

For a much simpler and more comfortable use of this approach we further created an SQL pre-processor enabling to specify all SQL-statements only on the basis of the respective basic tables. The pre-processor analyses the users' SQL query and translates it according to the structure of the basic table and the corresponding user-specific table applying necessary join operations (for select statements) or sequential operations (for insert statements).

5 PERFORMANCE TEST

We implemented the solutions presented in Section 4 and compared them with respect to inserting and reading performance. All tests were carried out with an Intel® Core™ i5-644HQ quad core CPU with 2.6 GHz, 16 GB DDR4 dual-channel RAM and a Samsung PM961 NVMe solid state drive on Windows 10 Pro. Two scenarios were analyzed:

1. Fixed number of datasets to be inserted and selected (1,000 and 50,000 datasets)
2. Increasing number of datasets to be inserted and selected to gain further insights on the runtime behavior of the three approaches.

For both experiments we differentiate two types of insertion: Single and bulk. "Single" means we insert each dataset with one individual statement. "Bulk" on the other hand means we group datasets to so-called bulks and insert a bulk with only one statement.

5.1 Scenario 1: Fixed Number of Datasets

In a first step we created 1,000 datasets, which were then to be inserted into the database and selected after the insertion. Each dataset contains 18 attributes and a size of 188 byte. The results of this test are given in Figure 8. The cell-based implementation is by far slower than the other two approaches. Compared to the database-based approach it is slower by a factor of 3.8 for the single insertion, by a factor of 50.6 for bulk insertion, and by a factor of 74 for reading. This results mainly from the fact that each cell of the respective database table is encrypted independently. For a larger number of datasets the cell-based approach was therefore not further considered.

Comparing the other two implementations we can see that the database-based approach is faster by a significant portion. The bulk insert is faster by a factor of about three, the single insert is about 1.6 times faster. The factor for the selection of the datasets is about 2.4 compared to the file-based implementation. Increasing the number of datasets to 50,000 leads to different factors comparing the file-based and database-based approach. Regarding the insertion, the factor is approximately 11.6, while the factor for reading datasets is about 4.8, i.e. both factors increase significantly. One reason for this behavior is most likely that the entire read and save mechanism of the file-based solution is only a prototypical implementation whereas the database-based solution makes use of a well-engineered storage engine (MySQL).

5.2 Scenario 2: Increasing Number of Datasets

The previous scenario illustrates that the insertion of datasets takes up more time than the reading the same datasets. It is, therefore, the main factor for the overall performance of the test example. In a second scenario (including the cell-based implementation) the insertion times for an increasing number of datasets are compared for both single and bulk insertion. The results are given in Figure 9. For each approach the computational time increase has been approximately linear for single insertion (Figure 9(a)). For the cell-based approach the increase is significantly higher. The file-based approach has a worse performance for both insertion types (single and bulk) compared to the database-based solution. In conclusion, for an increasing number of datasets, the amount of time needed for a single insert compared to a bulk insert grows notably larger, whereas the computational time per bulk

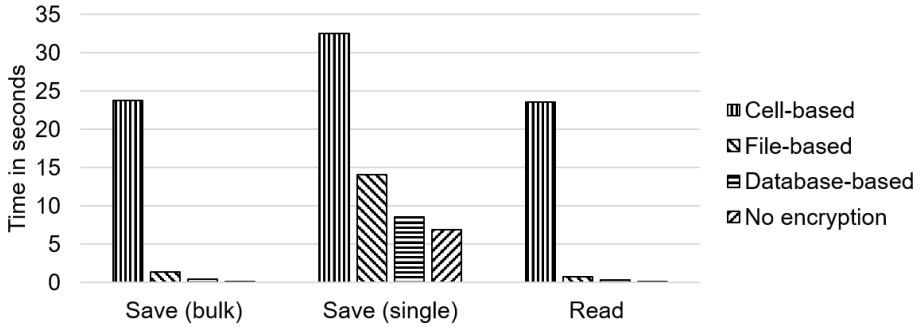

Figure 8: Test scenario 1 (1,000 datasets).

insert seems to be almost constant independent of the size of the bulk (ranging from 1 to 281 datasets) for the database-based approach (as well as the case of no encryption (Figure 9(b)). The cell-based approach also has the worst performance for bulk insertion with a factor of about 10 compared to the database-based approach and is not given in Figure 9(b) for reason of comprehensibility. For further implementations, bulk inserts need to be used whenever possible, as the computational time is nearly constant per bulk.

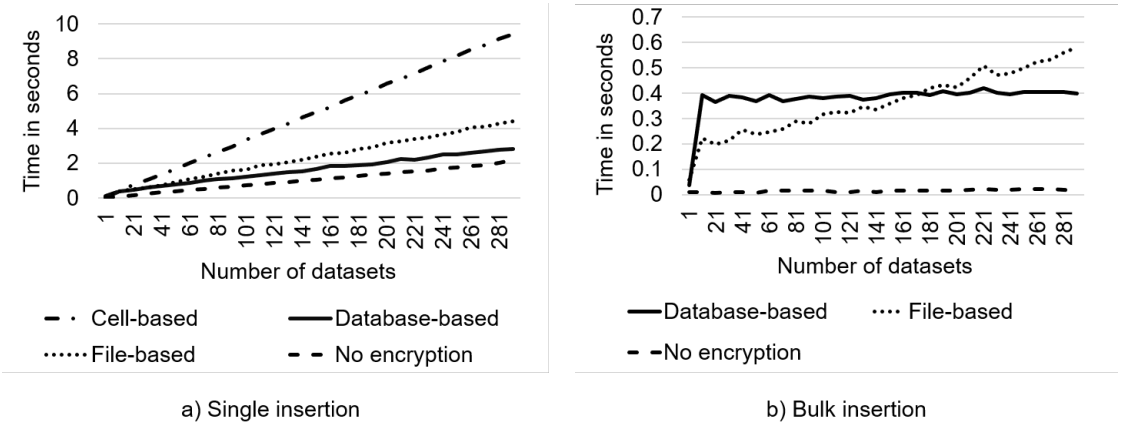

Figure 9: Test scenarios with increasing dataset size.

In comparison to a data model without any encryption the additional computational time for encryption is acceptable in the given tests. In comparison to the database-based approach, the encryption time is only a small portion of the overall computation time for storing and reading data and, which is more important, is linear to the number of datasets (or bulks, respectively) processed.

There probably exists a possibility to optimize the file-based solution to get the same performance as the database-based, but the effort for such an implementation is very high. Looking at the database-based solution, there is much less implementation effort, and the storage engine of MySQL is relatively flexible.

6 CONCLUSION

Summarized, the file-based and database-based approaches have proven to be much more efficient than the cell-based implementation. Although showing better results considering the performance, the database-based approach does not need to be superior, as the file-based approach offers much more flexibility in terms of the encryption technologies applied, mainly because we do not depend on a database system vendor. Furthermore, it is very likely that the file-based solution can be optimized

for better computational times. Therefore, it could be of interest to further analyze and optimize the file-based approach. In our specific project we opted for the database-based solution because in our particular cloud-environment the flexibility of encryption algorithms is less important than the implementation effort.

With respect to the requirements of users to hide sensitive data, another scope of our research is to define an expert system to decide on how results can be presented such that all requirements are met. Here, no participant should be able to derive sensitive information, such as cost rates of suppliers, from the results of simulation experiments.

ACKNOWLEDGEMENT

This work is partially based on the "Entwicklung eines Supply Chain Simulationswerkzeugs in der Cloud unter Berücksichtigung dynamischer Verschlüsselungstechnologien" (SimChain MONSTER) project, which is funded by the Federal Ministry for Economic Affairs and Energy, Germany.

REFERENCES

Archibald, G., N. Karabakal, and P. Karlsson. 1999. "Supply Chain vs. Supply Chain: Using Simulation to Compete Beyond the four Walls". In *Proceedings of the 1999 Winter Simulation Conference*, edited by P. A. Farrington et al., 1207–1214. Piscataway, New Jersey: IEEE.

Bagchi, S., S. J. Buckley, M. Ettl, and G. Y. Lin. 1998. "Experience Using the IBM Supply Chain Simulator". In *Proceedings of the 1998 Winter Simulation Conference*, edited by D. J. Medeiros et al., 1387–1394. Piscataway, New Jersey: IEEE.

Bethencourt, J., A. Sahai, and B. Waters. 2007. "Ciphertext-Policy Attribute-Based Encryption". In *2007 IEEE Symposium on Security and Privacy (SP '07)*. May, 20th-23rd, Berkeley, California, USA, 321–334.

Casters, M., R. Bouman, and J. van Dongen. 2010. *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration*. IT Pro. Hoboken, NJ: Wiley Publishing.

De Capitani Di Vimercati, S., S. Foresti, S. Jajodia, S. Paraboschi, G. Pelosi, and P. Samarati. 2010. "Encryption-based Policy Enforcement for Cloud Storage". In *2010 International Conference on Distributed Computing Systems Workshops*. June, 21st-25th, Genova, Italy, 42–51.

Diffie, W., and M. Hellman. 1976. "New Directions in Cryptography". *IEEE Transactions on Information Theory* 22(6):644–654.

Ferraiolo, D., J. Cugini, and D. R. Kuhn. 1995. "Role-based Access Control (RBAC): Features and Motivations". In *Proceedings of 11th Annual Computer Security Application Conference*. December 11th-15th, New Orleans, Louisiana, 241–248.

Ferraiolo, D. F., R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. 2001. "Proposed NIST Standard for Role-based Access Control". *ACM Transactions on Information and System Security (TISSEC)* 4(3):224–274.

Goyal, V., O. Pandey, A. Sahai, and B. Waters. 2006. "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data". In *Proceedings of the 13th ACM Conference on Computer and Communications Security Workshops*. October, 30th - November 3rd, Alexandria, VA, USA, 89–98.

Gutenschwager, K., and K. Alicke. 2004. "Supply Chain Simulation mit ICON-SimChain". In *Logistik Management*, edited by T. Spengler, S. Voß, and H. Kopfer, 161–178. Heidelberg: Physica.

Linstedt, D., and M. Olschimke. 2015. *Building a Scalable Data Warehouse with Data Vault 2.0*. San Francisco, CA: Elsevier Science.

Mönkeberg, A., and R. Rakete. 2000. "Three for One: Role-based Access-control Management in Rapidly Changing Heterogeneous Environments". In *Proceedings of the fifth ACM Workshop on Role-Based Access Control*, 83–88. July 26th-27th, Berlin, Germany, 83–88.

Rabe, M., K. Gutenschwager, T. Fechteler, and M. U. Sari. 2013. "A Data Model for Carbon Footprint Simulation in Consumer Goods Supply Chains". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy et al., 2677–2688. Piscataway, New Jersey: IEEE.

Richardson, L., and S. Ruby. 2008. *RESTful Web Services*. Gravenstein Highway North Sebastopol, CA: O'Reilly Media, Inc.

Roeckle, H., G. Schimpf, and R. Weidinger. 2000. "Process-oriented Approach for Role-finding to Implement Role-based Security Administration in a Large Industrial Organization". In *Proceedings, Fifth ACM Workshop on Role-Based Access Control*. July 26th-27th, Berlin, Germany, 103–110.

Sandhu, R. S., E. J. Coyne, H. L. Feinstein, and C. E. Youman. 1996. "Role-based Access Control Models". *Computer* 29(2):38–47.

Stäblein, T., H. Baumgärtel, and J. Wilke. 2007. "The Supply Net Simulator SNS: An Artificial Intelligence Approach for Highly Efficient Supply Network Simulation". In *Management logistischer Netzwerke*, edited by H.-O. Günther, D. C. Mattfeld, and L. Suhl, 85–110. Heidelberg: Physica.

AUTHOR BIOGRAPHIES

KAI GUTENSCHWAGER is head of the Institute of Information Engineering at the Ostfalia University of Applied Sciences, Wolfenbüttel. He studied Business Informatics and received his doctoral degree from the Technical University of Braunschweig. He worked for SimPlan as a head of the office being a simulation expert in the field of logistics and supply chain management. From 2009 he was a professor for information systems in logistics at Ulm University of Applied Sciences and since 2013 he is a professor at Ostfalia University of Applied Sciences. His research is focused on IT-based methods for the simulation and optimization of production and logistics systems. His e-mail address is k.gutenschwager@ostfalia.de.

MARKUS RABE is full professor for IT in Production and Logistics at the Technical University Dortmund. Until 2010 he had been with Fraunhofer IPK in Berlin as head of the corporate logistics and processes department, head of the central IT department, and a member of the institute direction circle. His research focus is on information systems for supply chains, production planning, and simulation. Markus Rabe is vice chair of the "Simulation in Production and Logistics" group of the simulation society ASIM, member of the editorial board of the Journal of Simulation, member of several conference program committees, has chaired the ASIM SPL conference in 1998, 2000, 2004, 2008, and 2015, and was local chair of the WSC'2012 in Berlin. More than 190 publications and editions report from his work. His e-mail address is markus.rabe@tu-dortmund.de.

MARCEL THEILE (B.Sc.) is currently a student and working on his master thesis at Ostfalia University of Applied Sciences, where he also received his bachelor degree. Besides he is working as research assistant at Ostfalia. His research focuses on finding solutions for an encryption and right management model in the context of multi-organizational simulation applications. His email address is ma.theile@ostfalia.de.

BASTIAN WILHELM (M.Sc.) is external PhD student at IT in Production and Logistics at the Technical University Dortmund and currently working as research assistant on the Ostfalia University of Applied Sciences. He received his bachelor as well as his master degree at Ostfalia. His research focuses on finding solutions for an encryption and right management model as well as optimization models in the context of multi-organizational simulation applications. His email address is ba.schulten@ostfalia.de.