# Virtualization Model for Processing of the Sensitive Mobile Data

**Andrzej Wilczyński and Joanna Kołodziej**

**Abstract** In this chapter, the k-anonymity algorithm is used for anonymization of sensitive data sending via network and analyzed by experts. Anonymization is a technique used to generalize sensitive data to block the possibility of assigning them to specific individuals or entities. In our proposed model, we have developed a layer that enables virtualization of sensitive data, ensuring that they are transmitted safely over the network and analyzed with respects the protection of personal data. Solution has been verified in real use case for transmission sports data to the experts who send the diagnosis as a response.

## 1 Introduction

### *1.1 Data Virtualization*

Virtualization usually referrs to the situations where applications can use resources no matter where they are located, how they are stored or implemented and where they come from. Data virtualization is a variation of virtualization, where we can distinguish an abstract layer that provides a simpler interface and methods for accessing data. Data sources may be many, but the user who relies on this data will see one abstract layer. User does not have to know for instance what database languages are used to retrieve data from their physical storage, what type of API is used or what is the message structure. The end user may have the impression that this is one large database. Rick van der Lans describes data virtualization as follows: *Data*

A. Wilczyński (✉) · J. Kołodziej
Cracow University of Technology, Warszawska st 24, 31-155 Cracow, Poland
e-mail: and.wilczynski@gmail.com

A. Wilczyński
AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Cracow, Poland

J. Kołodziej
e-mail: jokolodziej@pk.edu.pl



J. Kołodziej et al. (eds.), *Modeling and Simulation in HPC and Cloud Systems*,
Studies in Big Data 36, https://doi.org/10.1007/978-3-319-73767-6_7

*virtualization is the technology that offers data consumers a unified, abstracted, and encapsulated view for querying and manipulating data stored in a heterogeneous set of data stores*, [1].

## 1.2 Mobile Data Virtualization

Still developing and increasingly sophisticated mobile applications need access to business data. The security of such data is very important and communication protocols must meet the appropriate trust level. There are two issues with mobile data transfer:

- Mobile Application Developers, which for mobile application developers is the standard for creating queries for downloading business data. This standard greatly improves application performance by omitting the integration process with each of the data providers separately.
- Development and Operations, due to the sensitivity of the data, provides the right level of security, easy access to data. Management is one of the most important issues.

Existing mechanisms for accessing business data from mobile applications are in most cases based on the API. However, there are some reputable faults associated with this type of solution. First and foremost, the API is different for each data provider and may change over time, so it is important to take care of the different ways of integration with these providers, which involves a large and continuous workflow. A typical mobile Api platform is an abstract backend-as-a-service (MBaaS) layer that defines a source and makes data available to potential mobile applications.

## 1.3 Mobile Cloud Computing Data Virtualization Security Issues

Nowadays, the use of mobile applications is practiced by the majority of the population. Applications are constantly expanding and require ever-increasing computing resources. Due to the fact that they are executed on mobile devices their performance has some limitations. This is related to limited energy and computing resources. It happens that the huge amount of data that they can deliver also does not fit on the device. According to the demand appeared the concept of mobile cloud, where data processing and storage can take place externally. This gives the ability to use very complex applications even on weak devices. It is therefore possible to build applications that deal with such operations/problems as image processing, natural language processing, crowd computing, GPS/Internet data sharing, sensor data applications, multimedia search, social networking, [2]. Mobile Cloud Computing (MCC) combines cloud computing with mobile cloud.

There are some security challenges that need to be addressed in this kind of systems, [3]:

- **Volume** - data is transferred between the different layers and then combined, which can cause problems in the integrity and inviolability of the data.
- **Velocity** - the speed of data collection forces the use of such encryption algorithms, which ensure the proper flow of data.
- **Variability** - data privacy must be ensured when data is no longer valid and should be deleted.

Each MCC system should ensure that the processing and transfer of data is consistent with the above security issues.

## 2 Motivation

Very often, data processing or computing operations on mobile devices consume large amounts of computing resources, which in turn puts a heavy burden on power and batteries. Processing of this data can take place in the cloud system and processing result can be returned to the mobile device as a response. It also happens that these data must be examined by experts so that further analysis is possible. These kind of data are often sensitive data that can not be shared with third parties. The main goal of this chapter is to design a model for sending sensitive data from mobile devices to cloud computing system (CC). Model should provide satisfying level of security and ability to analyze or process data by experts or third parties. They can not know the identity of the person who is associated with data due to compliance with data protection standards. The aim of this chapter is to design and implement a model meeting above requirements.

## 3 Related Work and Existing Solutions

One of the working examples is the approach proposed by Rocket Software. They provide integrated virtual views that allow direct access to mainframe data without having to transform them, called Rocket Data Virtualization (RDV). This technology enables real-time data usage without time–consuming ETL (extract, transform, load) operations. Typically, mainframes use log-based replication to locate operational data in System Management Facility (SMF). This type of data must be processed in such a way that it can be read in RDV, moreover, these data have to be transferred several times. The amount of data does not always allow you to quickly transfer them to the data warehouse. RDV delivers data in the right format without having to process it, so access to data is several times faster, [4].

Next approach of data virtualization is Red Hat JBoss Data Virtualization. This tool allows to combine data from heterogeneous environments, create virtual models and data access views, process them and share using simple interfaces. Data from multiple sources can be shared using SQL, such as JDBC or web services such as REST. For the user the data source is a one logical virtual model. Red HAT provides a graphical interface for easy creation this type of model. Each model allows to map source data to target formats required by end-applications. The application has the ability to integrate as data sources such tools as Hadoop, NoSql, SaSS, Data Warehouses and many type of files (xml, csv, excel), [5].

One possible solution for mobile data virtualization is the solution proposed by MadMobile. They define 3 elements that should comprise the mobile data virtualization platform:

- Mobile Data Sources
- Data Access APIs
- Mobile Data Catalog

The most interesting of the above three elements is Mobile Data Catalog, which is something like a repository that contains all the data sources. MadMobile has designed The KidoZen Mobile Data Virtualization Platform. This application creates a virtual representation for all data that can be downloaded by mobile applications, called Data Catalog. Each new element can be added to Data Catalog by specifying: data source, data source name, connector, operation, parameters, caching options. Next KidoZen makes the data available through the API using the Open Data Protocol, [6].

The next article shows the use of mobile cloud computing for real-time multimedia-assisted mobile food recognition application. The authors present there a mechanism for counting calories using CC. The main functions of the application are segmentation and image processing, and the use of deep learning algorithms to classify and recognize food. This type of operation due to the limitations of mobile devices can not be done directly on them. They use the Android capabilities for parting application activities into the front part installed on the mobile device and the backend where the application processing is done on the virtual Android image located in cloud, [7].

In the next chapter Mollah, see [8], describes the challenges and security issues in mobile cloud computing. He draws attention to aspects of MCC security, namely: cloud computing data security, virtualization security, partitioning offloading security, mobile cloud applications security, mobile device security, data privacy, location privacy and identity privacy. The general security requirements that are described by authors in this article apply to: confidentiality, integrity, availability, authentication and access control, privacy requirements.

## 4 Data Anonymization

Data anonymization (DA) is a process that allows data protection. In this process we can distinguish the mechanisms of encryption and deletion of personal data, which make it impossible to associate with individuals, provide their anonymity. Sensitive data transmitted over the network can be stolen and disclosed, which increases the risk of their use to the detriment of data providers. DA provides a high level of security even in the case of uncontrolled disclosure, because they can not be linked to the people they describe. EU has defined safety regulations for sensitive data. Given these regulations data can be divided into:

- Personal data - Data that allows direct identification of the person to whom they refer through the identification number or set of characteristics describing the person.
- Anonymous data - Data that can not be linked to the person to whom they relate. This can not be done by either the processor or any other person. After this process data is no longer personal data and is not subject to EU regulations on the protection of personal data.
- Pseudonymous data - after the process of anonymity there are some personal data, so they are still personal data. Nevertheless, the process of deanonimization is not an easy process, and it is difficult to identify the right person.

There are two approaches to DA:

1. K-anonymity - private tables contain a set of attributes that define and describe a person, if this set is available externally it is called a quasi-identifier.

**Definition 1** *(k-anonymity requirement)* Each data sharing must be done so that each combination of quasi-identifier values can be matched to k different respondents.

**Definition 2** *(k-anonymity)* Let $T(A_1, \ldots, A_m)$ be a table, and $QI$ be a quasi-identifier associated with it. $T$ is said to satisfy k-anonymity with respect to $QI$ if each sequence of values in $T[QI]$ appears at least with $k$ occurrences in $T[QI]$ [9].

To better illustrate the above definition, we show some example. Lets assume that we have the following table:

Table 1 consists 5 attributes and 8 records. There are 2 methods for obtaining anonymity for some k:

- Generalization - in this method, true attribute value is generalized, for instance the value of attribute “Date of birth” - 12071990 is replaced by 11061990 < Date of birth < 01011991.
- Suppression - in this method, some true values are replaced by asteriks’*’.

With respect to the quasi-identifiers $Dateofbirth, Sex$ data has 2-anonymity, because we have at least 2 rows with the same attributes, with respect to the quasi-identifiers $Dateofbirth, Sex, Salary$ data has 1-anonymity, because we have single occurrences of values (Table 2).

**Table 1** Private table

| | Name | Date of birth | Sex | Profession | Salary |
|---|---|---|---|---|---|
| 1 | Melania Wolska | 12071990 | F | Teacher | 2500 |
| 2 | Kajetan Nowicki | 21091987 | M | Programmer | 4000 |
| 3 | Maja Nowak | 21061957 | F | Doctor | 4000 |
| 4 | Stanisaw Zakrzewski | 25061946 | M | Waitress | 2000 |
| 5 | Jan Zalewski | 12071990 | M | Waiter | 2000 |
| 6 | Igor Gorski | 25061946 | M | Plumber | 2400 |
| 7 | Zuzanna Nowakowska | 21061957 | F | Dentist | 4400 |
| 8 | Anna Baran | 12071990 | F | Doctor | 4400 |
| 9 | Wojciech Jakubowski | 21091987 | M | Teacher | 4400 |

**Table 2** Public table

| | Name | Date of birth | Sex | Profession | Salary |
|---|---|---|---|---|---|
| 1 | * | 12061990 < Date of birth < 12081990 | F | * | 2500 |
| 2 | * | Date of birth > 21091986 | M | * | 4000 |
| 3 | * | 01061957 < Date of birth < 23061957 | F | * | 4000 |
| 4 | * | Date of birth < 25061947 | M | * | 2000 |
| 5 | * | 12071989 < Date of birth < 12071991 | M | * | 2000 |
| 6 | * | Date of birth < 25061947 | M | * | 2400 |
| 7 | * | 01061957 < Date of birth < 23061957 | F | * | 4400 |
| 8 | * | 12061990 < Date of birth < 12081990 | F | * | 4400 |
| 9 | * | Date of birth > 21091986 | M | * | 4400 |

The algorithm described above has some drawbacks, it is susceptible to two types of attacks:

- Background Knowledge Attack - The case where the association of one or more quasi-identifier attributes containing sensitive values leads to a reduction in the range so that it can be deduced which record fits the individual.
- Homogeneity Attack - case where all sensitive values in set k records are identical.

2. l-Diversity - an extension of $k$-anonymity model, where the granularity of the data representation is reduced.

**Definition 3** *(l-diversity)* The table is l-diverse if for every $q* - block$

$$-\sum_{s \in S} p_{(q*,s)} log(p_{(q*,s\prime)}) \geq log(l) \tag{1}$$

where $p_{(q*,s)} = \frac{n_{(q*,s)}}{\sum_{s\prime \in S} n_{(q*,s\prime)}}$ is the fraction of tuples in the $q* - block$ with sensitive attribute value equal to $s$ [10].

**Table 3** 2-anonymous table

| | Name | Date of birth | Sex | Profession | Salary |
|---|---|---|---|---|---|
| 1 | * | 12061990 < Date of birth < 12081990 | F | * | 2500 |
| 8 | * | 12061990 < Date of birth < 12081990 | F | * | 4400 |
| 2 | * | Date of birth > 21091986 | M | * | 4000 |
| 9 | * | Date of birth > 21091986 | M | * | 4400 |
| 3 | * | 01061957 < Date of birth < 23061957 | F | * | 4000 |
| 7 | * | 01061957 < Date of birth < 23061957 | F | * | 4400 |
| 4 | * | Date of birth < 25061947 | M | * | 2000 |
| 6 | * | Date of birth < 25061947 | M | * | 2400 |
| 5 | * | 12071989 < Date of birth < 12071991 | M | * | 2000 |

**Table 4** 2-diversity table

| | Name | Date of birth | Sex | Profession | Salary |
|---|---|---|---|---|---|
| 1 | * | 01061957 ≤ Date of birth < 12081990 | F | * | 2500 |
| 8 | * | 01061957 ≤ Date of birth < 12081990 | F | * | 4400 |
| 2 | * | Date of birth > 01061957 | M | * | 4000 |
| 9 | * | Date of birth > 01061957 | M | * | 4400 |
| 3 | * | 01061957 ≤ Date of birth < 23061957 | F | * | 4000 |
| 7 | * | 01061957 ≤ Date of birth < 23061957 | F | * | 4400 |
| 4 | * | Date of birth ≤ 01061957 | M | * | 2000 |
| 6 | * | Date of birth ≤ 01061957 | M | * | 2400 |
| 5 | * | 01061957 ≤ Date of birth < 12071991 | M | * | 2000 |

Analyzing Table 3, you can see that the data are not susceptible to homogeneity attack. However, considering the records of 1 and 8, namely Melania and Anna. Melania may be Anna's neighbor whom she knows she was born on the same day as her, she also knows that she's earning 2500, so she can deduce that Anna earns 4400. After diversity process data are not susceptible to background knowledge attack as it presented in Table 4.

## 5 Data Exchange Model

We present in Fig. 1 the original contribution of this chapter is a model for transmission sensitive data via network. that model ensures that the receivers of data are not be able to assign them to real people, but they can send feedback to them. On the model we can distinguish the following elements:

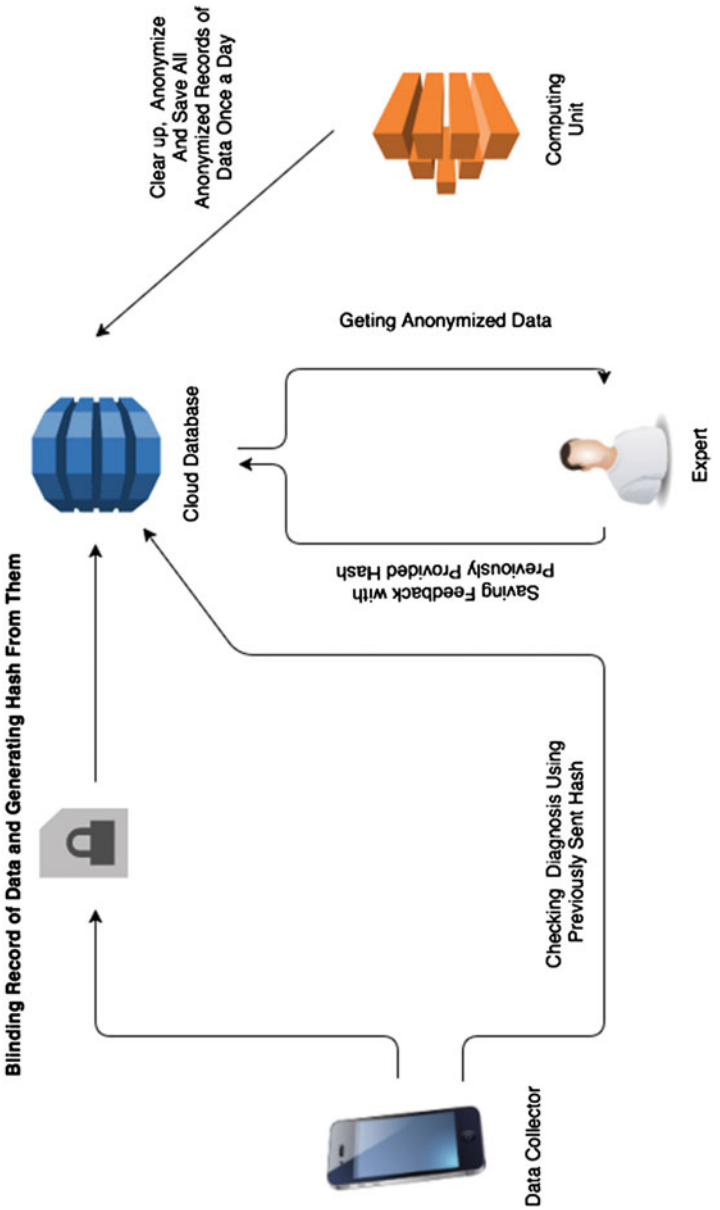


**Fig. 1** Data virtualization model and hierarchy

- Data Collector - devices for collecting data from sensors;
- Cloud Database - database for storing blinded and anonymized data;
- Computing Unit - computational unit for decrypting and anonymizing data;
- Expert - person responsible for data analysis and feedback.

Data Collector, before sending data to the database, performs blinding operations on them using the pseudo one-time pad (OTP) algorithm. OTP is an encryption technique where message is paired with a random secret key [11]. Each character is encrypted by combining it with the adequate character from key. Algorithm uses a short key and as a result returning binary data string. In this work, algorithm has been called a pseudo OTP algorithm because the same key is used many times to encrypt and decrypt various messages, in the real OTP after encrypt and decrypt message the key is thrown and a new one is randomly generated. It is used only to blind sensitive data, which is sufficient at this stage. Encryption has been extended to binary data, which means that instead of characters, a binary key is used. After blinded operation with the MD5 function [12], the hash of the resulting data is generated.

Blinded data together with hash goes to the Cloud Database. Once in a while, all data records from database are downloaded by the Computing Unit and cleared up using the same key that was used to blind them. After that, data are anonymized and stored in the database, allowing them to be analyzed. Expert analyzes sensitive data and sends the feedback, which is stored under the same hash as the blinded data. Expert does not know who the data belongs to, which ensures compliance with the personal protection laws.

After a certain period of time, data collector performs a database query to verify the result of the analysis of the previously transmitted data. The same hash that was previously generated from the blinded data is used for the search. If the result has been already saved by an expert, data collector gets it, otherwise the query is repeated in a while.

## 6 Model Use Case and Implementation

To illustrate the presented model, we prepared an implementation of the system for the collection and analysis of sports data transmitted by athlete's sensors. These sensors collect information about the heart rate of the person performing the exercise and send them to the CC. As a result, the person receives a diagnosis of the time spent in each exercise zone during the training. The data used for the analysis is the real data collected from training of 19 people. Implementation was done in JAVA.

It can be observed from Table 5 that we have 8 attributes describing each person. The heart rate attribute was cut off due to the very long representation.

Table 6 presents blinded data together with generated MD5 hash. Each attribute is first saved in JSON format, then string of JSON format is blinded. Below we present fragment of json for a single record:

**Table 5** Sport data

| | Name | Sport | Weight | Height | Heart rate (resting) | Age | Duration (min) | Heart rate |
|---|---|---|---|---|---|---|---|---|
| … | … | Running | … | … | … | … | … | … |
| 8 | Barreta Page | Running | 51 | 156 | 90 | 39 | 115 | 0, 90, 92, 98, 97, 97, 99, 106, 112, 116… |
| 9 | Lens Temple | Cycling | 74 | 162 | 91 | 37 | 115 | 0, 0, 170, 151, 0, 196, 196, 196, 196… |
| 10 | Bevise Kenyon | Cycling | 81 | 167 | 82 | 35 | 158 | 0, 221, 87, 91, 102, 109, 107, 103… |
| 11 | Adolphe Nigellus | Cycling | 88 | 174 | 83 | 35 | 174 | 0, 111, 110, 112, 115, 116, 116, 115… |
| 12 | Bentleye Nelson | Cycling | 55 | 178 | 85 | 34 | 115 | 0, 0, 0, 75, 206, 199, 201, 201, 154… |
| … | … | Running | … | … | … | … | … | … |

**Table 6** Blinded sport data

| | Hash | Data |
|---|---|---|
| | … | … |
| 8 | 782d74c56dbe99d6cc494a1c7284305d | 00000010 0011110 00011101 00001100 00001011 000110… |
| 9 | eaf29fdefc33c8780f53e1de61c88d52 | 00000010 0011110 00011101 00001100 00001011 000110… |
| 10 | 4fb95b92fa76bdf9cd208d05dafb90b4 | 00000010 0011110 00011101 00001100 00001011 000110... |
| 11 | 1e11b033a1239321a4898dd1320d5428 | 00000010 0011110 00011101 00001100 00001011 000110... |
| 12 | b1580545d102ca661c4dfacf292a9278 | 00000010 0011110 00011101 00001100 00001011 000110... |
| | ... | ... |

{"*duration*" : 115, "*heartrate*" : "00, 0, 170, 151, 0, 196, 196, 196, 196, 196, 196, 196...", ...}

In implementation we used ARX library for anonymization [13]. In this library we can distinguish 4 types of attributes: insensitive, sensitive, quasi-identifying and identifying. Our use case classifies the attributes as follows:

- {sport, heart rate (resting), duration, heart rate} - insensitive
- {name} - identifying
- {weight, height, age} - quasi-identifying

For weight, height and age generalization has been used. For each quasi-identifier attribute we have prepared individual generalization hierarchy using intervals. As a

**Table 7** 2-anonymous sport data

| | Name | Sport | Weight | Height | Heart rate (resting) | Age | Duration (min) | Heart rate |
|---|---|---|---|---|---|---|---|---|
| … | … | … | … | … | … | … | … | … |
| 8 | * | Running | [48, 60[ | * | 90 | [36, 45[ | 115 | 0, 90, 92, 98, 97, 97, 99, 106, 112, 116… |
| 9 | * | Cycling | [72, 84[ | * | 91 | [36, 45[ | 115 | 0, 0, 170, 151, 0, 196, 196, 196, 196… |
| 10 | * | Cycling | [72, 84[ | * | 82 | [24, 36[ | 158 | 0, 221, 87, 91, 102, 109, 107, 103… |
| 11 | * | Cycling | [84, 96[ | * | 83 | [24, 36[ | 174 | 0, 111, 110, 112, 115, 116, 116, 115… |
| 12 | * | Cycling | [48, 60[ | * | 85 | [24, 36[ | 115 | 0, 0, 0, 75, 206, 199, 201, 201, 154… |
| … | … | … | … | … | … | … | … | … |

model 2-Anonymity has been adopted. The weights of the individual attributes are respectively: 0.5, 0.5, 0.5. Data presented in Table 7 have been anonymized in such a way that the expert has still no problems with a diagnosis.

## 7 Simulation Analysis

The implementation has shown that it is possible to send sensitive data meeting appropriate standards without losing their quality and consistency. These data can be sent to the cloud computing where their processing or analysis can be carried out independently. This approach provides the right level of security and enables performing computational operations or processing large amounts of data type of Big Data in an independent cloud computing environment. As we can see on Table 8 the whole process of assuming blinding and anonymization of data is done very fast. Simulation was performed on a computer MacBook Pro, 2,7 GHz Intel Core i5, 8 GB 1867 MHz DDR3.

Amount of data is small because in implementation we did not want to use data generator, the real data was used. Access to this type of data is very limited. However, the results we received are satisfactory and well predicted for the future.

**Table 8** Process execution times

| Operation type | Average time (miliseconds) |
|---|---|
| Blinding one record of data | 744 |
| Anonymization 19 records of data | 320 |

## 8 Conclusions and Future Work

Data anonymization can support processes where sensitive data is sent. In the proposed model we have provided the ability to analyze data through external resources without disclosing to whom those data belong. Experts or analysts do not need to know the identities of the people they diagnose, all they have to do is to send feedback to them. Model can also be implemented in different environment. For instance, in the cloud computing in task scheduling, where the scheduler does not need to know who orders task and for what task has to be executed.

The future work will focus on optimizing the selection of hierarchies for generalizing attributes and their weights. We are considering the use of Stackelberg game [14] to decide how best to match these parameters. Currently, this process is done manually and should be automated.

**Acknowledgements** This chapter is based upon work from COST Action IC1406 High-Performance Modelling and Simulation for Big Data Applications (cHiPSet), supported by COST (European Cooperation in Science and Technology).

## References

1. van der Lans, R.: Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses. Morgan Kaufmann Publishers Inc., San Francisco (2012)
2. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: a survey. Future Gener. Comput. Syst. **29**, 84106 (2013)
3. Jakóbik, A.: Big Data Security. Springer, Berlin (2016). https://doi.org/10.1007/978-3-319-44881-7_12
4. Software, R.: Rocket data virtualization. PDF document, http://www.rocketsoftware.com/sites/default/files/resource_files/DS_Data_DVS%20012615.pdf?flag=meta&product=rocket-data-virtualization&family=rocket-data&solution=data-virtualization&resourcetype=datasheet&resourcebn=rocket-data-virtualization&resourcefbn=DS_Data_DVS%20012615.pdf
5. Redhat: Jboss data virtualization. Electronic document, https://developers.openshift.com/jboss-xpaas/data-virtualization.html
6. Kidozen: From mdm to mdm. PDF document, http://www.kidozen.website/wp-content/uploads/2015/12/Mobile_Data_Virtualization.pdf
7. Pouladzadeh, P., Peddi, S.V.B., Kuhad, P., Yassine, A., Shirmohammadi, S.: A virtualization mechanism for real-time multimedia-assisted mobile food recognition application in cloud computing. Cluster Comput. **18**, 10991110 (2015)
8. Mollah, M., Azad, M.A.K., Vasilakos, A.: Security and privacy challenges in mobile cloud computing: Survey and way ahead. J. Netw. Comput. Appl. **84**, 3854 (2017)
9. Ciriani, V., De Capitani, S., di Vimercati, S., Foresti, P.Samarati: k-anonymity. secure data management in decentralized systems. Adv. Inf. Secur. **33**, 323353 (2007)
10. A. Machanavajjhala J. Gehrke, D.Kifer, M. Venkitasubramaniam: l-diversity: Privacy beyond k-anonymity. Proceedings of the 22nd International Conference on Data Engineering, 2006. ICDE '06. pp. 24–24 (2006)
11. Bellovin, S.M.: Frank miller: Inventor of the one-time pad. Cryptologia **35**(3), 203–222 (2011). https://doi.org/10.1080/01611194.2011.583711
12. Preneel, B.: Cryptographic Hash Functions: Theory and Practice. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-17401-8_9

13. Prasser, F., Kohlmayer, F.: Putting Statistical Disclosure Control Into Practice: The ARX Data Anonymization Tool. Springer, Berlin (2015)
14. Jakóbik, A., Wilczynski, A.: Using polymatrix extensive stackelberg games in security aware resource allocation and task scheduling in computational clouds. J. Telecommun. Inf. Technol. **1**, 71–80 (2017)