Chapter 38
Security Model for Mobile Cloud Database as a Service (DBaaS)

Kashif Munir
University of Hafr Al-Batin, Saudi Arabia

ABSTRACT

There's a big change happening in the world of databases. The industry is buzzing about Database-as-a-Service (DBaaS), a cloud offering that allows companies to rent access to these managed digital data warehouses. Database-as-a-service (DBaaS) is a cloud computing service model that provides users with some form of access to a database without the need for setting up physical hardware, installing software or configuring for performance. Since consumers host data on the Mobile Cloud, DBaaS providers should be able to guarantee data owners that their data would be protected from all potential security threats. Protecting application data for large-scale web and mobile apps can be complex; especially with distributed and NoSQL databases. Data centers are no longer confined to the enterprise perimeter. More and more enterprises take their data to the Mobile Cloud, but forget to adjust their security management practices when doing so. Unauthorized access to data resources, misuse of data stored on third party platform, data confidentiality, integrity and availability are some of the major security challenges that ail this nascent Cloud service model, which hinders the wide-scale adoption of DBaaS. In this chapter, I propose a security model for Mobile Cloud Database as a Service (DBaaS). A user can change his/her password, whenever demanded. Furthermore, security analysis realizes the feasibility of the proposed model for DBaaS and achieves efficiency. This will help Cloud community to get an insight into state-of-the-art progress in terms of secure strategies, their deficiencies and possible future directions.

INTRODUCTION

DBaaS provides professional databases that can get running and ready in a matter of minutes without a lot of training or personnel. A service provider chooses most of the options, offering the "best" configuration for most needs.

While individual systems can become unique "snowflake" servers, DBaaS tends to avoid that by simplifying and normalizing the customization, management, and upkeep for administrators. Overall,

DOI: 10.4018/978-1-5225-8176-5.ch038

Figure 1. Cloud DBaaS (Krishna & Roger, 2012)

the service makes it easier to solve problems, correct mistakes, and transfer data from one system to the next. They can scale as large as necessary, fit the needs of the customers, and offer better availability and security than most in-house operations.

DBaaS is also accessible to a larger audience because, like other "as a service" cloud innovations, it is largely defined, configured, and driven by code—not commands typed into a terminal. So, instead of requiring database specialists, developers themselves can easily create and manage database-backed apps on cloud-based development platforms.

DBaaS is already responsible for much of the growth in some key technologies, particularly open-source databases like MySQL. In other words, traditional database deployment is somewhat stagnant, and most new deployments are DBaaS.The demand is so high that some tech giants started offering a managed "as a service" version of their own (Baron S, 2015).

DBaaS provides automated services where consumers can request database-oriented functionalities from a dedicated service hosted on Cloud. The model is end user driven and provides self-service provisioning. It is based on architectural and operational approach (Oracle, 2011), which provides new and distinctive ways of using and managing database services. There are many other database services which are available today but DBaaS differs from those traditional databases because its architecture has two major attributes (Oracle, 2011), Service-orientated as database facilities are available in the form of service. Customer self-service interaction model as organizations are allowed to use, configure and deploy the Cloud database services themselves without any IT support and without purchasing any hardware for specified purpose. These are the three main phases in the overall DBaaS architecture as depicted in Figure 1.

1. Consumers request the database deployment via Cloud.
2. Consumers adjust the capacity as demand changes.
3. Consumers can retire from the app when not needed.

Luca et al. (2012) advised against using any intermediary component for accessing the database on behalf of the clients, since it becomes a single point of failure. Security and availability of DBaaS services are bounded by this trusted intermediary proxy server.

Cong et al. (2013) proposed a similar approach which puts forth an idea of using third party auditors. This approach is suitable for preserving data integrity when data is outsourced to the DBaaS providers and users get access on-demand high quality services without facing maintenance burden of local data storage.

Nithiavathy (2013) proposed integrity auditing mechanism that utilizes distributed erasure-coded data for employing redundancy and homomorphic token. This techinque allows third party auditors and users to audit their logs and events at Cloud storage using light weight communication protocol at less computation cost.