AUTHENTICATION SCHEME FOR DATABASE AS A SERVICE(DBAAS)

KashifMunir and Lawan A. Mohammed

University of Hafr Al Batin, KSA

ABSTRACT

IT Companies have shifted their resources to the cloud at rapidly increasing rate. As part of this trend companies are migrating business critical and sensitive data stored in database to cloud-hosted and Database as a Service (DBaaS) solutions.Of all that has been written about cloud computing, precious little attention has been paid to authentication in the cloud. In this paper we have designed a new effective authentication scheme for 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. We also proposed an efficient authentication scheme to solve the authentication problem in cloud. The proposed solution which we have provided is based mainly on improved Needham-Schroeder's protocol to prove the users' identity to determine if this user is authorized or not. The results showed that this scheme is very strong and difficult to break it.

KEYWORDS

Cloud Computing, NoSQL, Database Security, DBaaS, Authentication Protocol,

1. INTRODUCTION

A mobile cloud approach enables developers to build applications designed specifically for mobile users without being bound by the mobile operating system and the computing or memory capacity of the mobile device. Mobile cloud computing servicesare generally accessed via a mobile browser from a remote webserver, typically without the need for installing a client application on the recipient device.

Now, many mobile applications are developed based on mobile databases on devices and conventional databases. Database authentication is the process or act of confirming that a user who is attempting to log in to a database is authorized to do so, and is only accorded the rights to perform activities that he or she has been authorized to do.

Database as a Service or simply DBaaS provides professional databases that can get running and ready in a matter of minutes without a lot of training or personnel effort. 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, 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.

DOI: 10.5121/ijccsa.2018.8502

DBaaS isalready 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 (Schwartz, 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): 1).Service-orientated as database facilities are available in the form of service and 2). 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 below.

i. Consumers request the database deployment via Cloud.
ii. Consumers adjust the capacity as demand changes.
iii. Consumers can retire from the app when not needed.

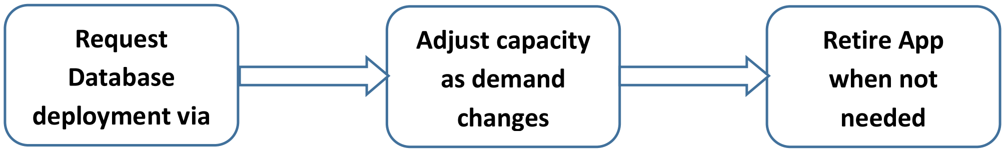
Request Database deployment via
Adjust capacity as demand changes
Retire App when not needed

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

2. LITERATURE REVIEW

Threats and vulnerabilities are a foremost challenge in the field of cloud computing. To address these challenges and to provide security and privacy (Ruchika&Rajarathnam, 2016). present a Software-as-a-Service (SaaS) application with a data model with built-in security and privacy. This data model enhances security and privacy of the data by attaching security levels in the data itself expressed in the form of XML instead of relying entirely on application level access controls. Similarly, a survey of different vulnerability attacks on cloud virtualization was performed in (Titin &Ugrasen, 2016), They also presents a concept for the removal of Cross Site Scripting (XSS) vulnerabilities to secure the cloud environment.
A Secure Data Transmission Mechanism (SDTM) was proposed in (Abdullah et., al., 2013). The authors developed SDTM enhanced with Malicious Packets Detection System (MPDS) which is a set of technologies and solutions. It enforces security policy and bandwidth compliance on all devices seeking to access Cloud network computing resources, in order to limit damage from emerging security threats and to allow network access only to compliant and trusted endpoint devices.

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.

Conget 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.

Jia et al.(2011) presents framework for secure data service with proxy re-encryption (PRE) scheme and identity based encryption (IDE) scheme .In this scheme, privacy of user is secured as the cryptography of data is done by user but it increases the energy and processing requirement of mobile device.
Huangs et al.(2011) proposed framework for authentication on MobiCloud, to achieve secure data processing.Similarly, Hsueh et al.(2011) proposed authentication mechanism in which mobile device encrypts the credential information file and stores it on cloud but infected cloud server can steal the user credential information by decrypting user's files.Recently, a comprehensive study of authentication methods in Mobile Cloud Computing (MCC) was presented in (Mojtaba etal., 2016). The aim was to describe MCC authentication and compare it with that of cloud computing. The taxonomy of the state-of-the-art authentication methods is devised and the most credible efforts are critically reviewed. Moreover, the authors present a comparison of the state-of-the-art MCC authentication methods considering five evaluation metrics. The results suggest the need for futuristic authentication methods that are designed based on capabilities and limitations of MCC environment. Finally, the design factors deemed could lead to effective authentication mechanisms are presented, and open challenges are highlighted based on the weaknesses and strengths of existing authentication methods.

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

Ferretti 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.

Similarly, (Qingji et al.(2012) investigated the issues of query integrity and a solution was proposed. The solution allows users to verify executed queries in Cloud database server along with the additional support of flexible join and aggregate queries. And, the solution proposed by (Maciej et al., 2013) covers data key management, data encryption and data integrity which ensure high data security and access efficiency.
Risk issues and challenges were presented in (Mouna&Latifa, 2012). The authors show how to solve these problems using a quantitative security risk assessment model named Multi-dimensional Mean Failure Cost (M2FC). Their scheme takes advantages of both Secret Sharing and Tornado code which can achieve the computational security and maintain low communication overhead in terms of shortened data dispersing size. The authors' model gives probabilistic proofs of Integrity of data by challenging random blocks from the server to reduce the computation and communication overhead, and also supports dynamic data operations to data shares in cloud using index table. Similar study was conducted in (Thamer, 2015), the authors highlight the different types of risks issues involved and how their existence can affect Global Software Development or simply GSD. They propose a new risk management process model. The risk model employs new processes for risk analysis and assessment. Its aim is to analyze cloud risks quantitatively and, consequently, prioritize them according to their impact on GSD objectives

General discussion of issues related to the data security management are explained in (Mohammed et al,. 2015). The authors present a proposed multi-cloud data management model called Byzantine Fault Tolerance Multi-Clouds Database (BFT-MCDB). The proposed BFT-MCDB model incorporates the Quantum Byzantine Agreement protocol and Shamir's Secret Sharing approach to secure business data storage in a multicloud environment.

3. DATABASE-AS-A-SERVICE(DBAAS)

Database-as-a-Service (DBaaS) is a service that is managed by a cloud operator (public or private) that supports applications, without the application team assuming responsibility for traditional database administration functions. With a DBaaS, the application developers donot need to be database experts, nor dothey have to hire a database administrator (DBA) to maintain the database Qingji et al. (2012)..DBaaS is a prime example of a service that's both exciting and at the same time full of difficult security issues.

Cloud providers want to offer the DBaaS service described above. In order to provide a complete DBaaS solution across large numbers of customers, the cloud providers need a high-degree of automation. Function's that have a regular time-based interval, like backups, can be scheduled and batched. Many other functions, such as elastic scale-out can be automated based on certain business rules. For example, providing a certain quality of service (QoS) according to the service level agreement (SLA) might require limiting databases to a certain number of connections or a peak level of CPU utilization, or some other criteria. When this criterion is exceeded, the DBaaS might automatically add a new database instance to share the load. The cloud provider also needs the ability to automate the creation and configuration of database instances Maciej et al. (2013).

Cloud operators are required to work on hundreds, thousands or even tens of thousands of databases at the same time. This requires automation. In order to automate these functions in a flexible manner, the DBaaS solution must provide an API to the cloud operator Hacigumus et al. (2012)The ultimate goal of a DBaaS is that the customer doesn't have to think about the database. Today, cloud users don't have to think about server instances, storage and networking, they just work. Virtualization enables clouds to provide these services to customers while automating much of the traditional pain of buying, installing, configuring and managing these capabilities. Now database virtualization is doing the same thing for the cloud database and it is being provided as Database as a Service (DBaaS). The DBaaS can substantially reduce operational costs and perform well. It is important to realise that the goal of DBaaS is to make things easier. Cloud Control Database as a Service (DBaaS) provides:

1. A shared, consolidated platform on which to provision database services
2. A self-service model for provisioning those resources
3. Elasticity to scale out and scale back database resources
4. Chargeback based on database usage

The aggressive consolidation of information technology (IT) infrastructure and deployment of Database as a Service (DBaaS) on public or private clouds is a strategy that many enterprises are pursuing to accomplish these objectives. Both initiatives have substantial implications when designing and implementing architectures for high availability and data protection. Database consolidation and DBaaS also drive standardization of I.T. infrastructure and processes. Standardization is essential for reducing cost and operational complexity. Databases deployed in the Bronze tier include development and test databases and databases supporting smaller work group and departmental applications that are often the first candidates for database consolidation and for deployment as Database as a Service (DBaaS).

Bronze is based upon single instance Oracle Databasewith Oracle Restart for auto-restart following recoverable outages. When a machinebecomes unusable or the database unrecoverable, the recovery time objective (RTO) is a function of how quickly a replacement system can be provisioned or a backup restored. In a worst case scenario of a complete site outage there will be additional time required to perform these tasks at a secondary location(Oracle, 2016).

4. SECURITY CHALLENGES TO DATABASE-AS-A-SERVICE(DBAAS)

Cloud computing and the notion of large-scale data-centers will become a pervasive technology in the coming years. There are some technology hurdles that we confront in deploying applications on cloud computing infrastructures: DBMS scalability and DBMS security. In this paper, we will focus on the problem of making DBMS technology cloud friendly. In fact, we will argue that the success of cloud computing is critically contingent on making DBMSs scalable, elastic, available, secure and autonomic, which is in addition to the other well-known properties of database management technologies like high-level functionality, consistency, performance, and reliability.In table 1 security challenges of DBaaS infrastructure along with their consequences and causes has been highlighted (Munir, 2015).

Table 1. Cloud DBaaS Security Challenges(Munir, 2015)

No.	Security Challenge	Description
1	Availability	• Temporary and permanent unavailability cause service breakdown • DOS Attacks, natural disasters, equipment failure
2	Access Control Issues	• Physical, personnel and logical control missing on organization’s internal and DBaaS Provider’s employees • Increase development and analysis cost is incurred when user management and granular access control is implemented
3	Integrity Check	• Need to avoid modification of configuration, access and data files • Require accuracy and integrity of data
4	Auditing and Monitoring	• Configuration requirements change continuously • Important for avoiding failures, backup maintenance, configuration of auto fail-over mechanisms • Require stark network and physical device , expertise and relevant resources
5	Data Sanitization	• Recovery of data by malicious sources if not properly discarded
6	Data Confidentiality	• Unencrypted data in memory, disk or in network may cause data breaches • Co-located application data is vulnerable to software bugs and errors in the Cloud • External organizations might also generate attacks
7	Data Replication and Consistency Management	• Replications between multiple servers cause management as well as consistency issues
8	Network Security	• Data flowing over the network (internet) is prone to hazardous circumstances and network performance issues. • Possible network failure reasons are: misconfiguration, lack of resource isolations, poor or untested business continuity, disaster recovery plan, network traffic modification
9	Data Locality	• Compliance and data-security privacy laws prohibit movement of sensitive data among countries • Issues faced when no one takes responsibility of data in location independent data storage
10	Data Provenance	• Complexity and time sensitiveness in provenance metadata • Intensive computations involved in getting required history • Fast algorithms, auto logs are needed
11	Insider Threats	• Employees can tap into sensitive and confidential data • Strict supply chain management and assessment is required
12	Outside Malicious Attackers	• Malicious attacks by hackers • Difficulty in synchronizing data between users and reporting corruption • Absence of authentication, authorization and accounting controls • Poor key management for encryption and decryption

5. PROPOSED SECURITYMODEL

DBaaS Security Placing a database in the cloud significantly changes its security threat landscape. While many of the traditional on-premises risks remain-data leakage risk from privileged users with access to the data, the presence of unidentified sensitive data and SQL injection attacks are some examples-the cloud introduces its own additional risks. On the other hand, there are ways to leverage the cloud by outsourcing some of the risk mitigation to the cloud provider. For example, physical access security and OS security is always the responsibility of the DBaaS provider.
To date, there is minimal work done in the field of security and privacy of DBaaS as compared to traditional data storage. Different approaches for securing DBaaS are discussed under this section with assorted categories of confidentiality, privacy, integrity and availability.

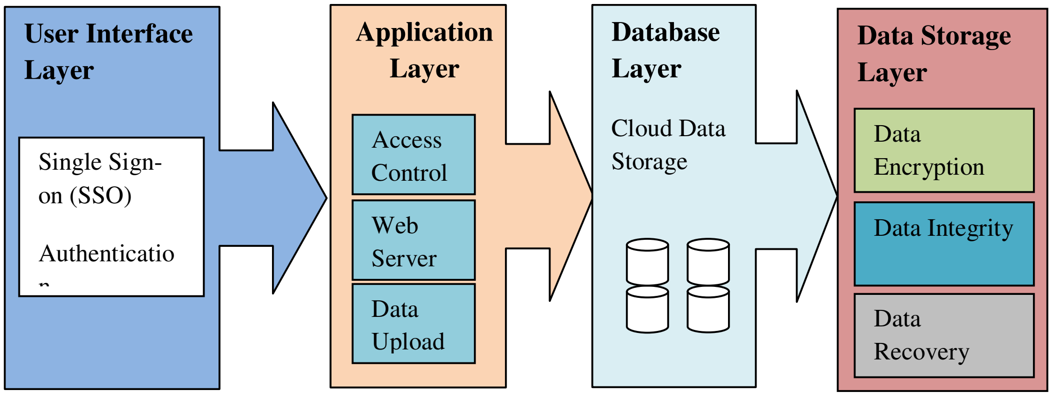

Figure 2. Secure Model for Cloud DBaaS(Munir, 2015)

State-of-the-art approaches mainly address generally adopted methods for their proposed models. Those methods are:Encryption based data security, which means hiding data content from service providers. Private information retrieval, which allows user to retrieve an item from the data server without revealing the content of that item. Information distribution, which is based on dispersing information instead of encrypting the data.

The model shown in figure 2 used Four-layer system structure, in which each layerperforms its own duty to ensure the data security of cloud layers.The first layer (User Interface Layer) isresponsible for user authentication; it is one time password authentication. User Interface Layer is used to access the service via internet. This allows users to easily utilize scalable and elastic database services available on Cloud infrastructure. The second layer (Application Layer) is used to access software services and storage space on the Cloud. As stated previously, consumers do not need to have hardware resources to accessthese services.Third layer (Database Layer) provides efficient and reliable service of managing database residing in the Cloud. It allows reuse of the query statements residing in the storage, thus saving time for querying and loading data. Fourth layer is data storage layer where Data is encrypted and decrypted at storage and retrieval stages, respectively. Data integrity and data recovery is also provided at this layer.

In the proposed model, a central console is responsible for the management of the resources. Taking backups, archiving and recovering data are now more feasible and less time-consuming because of these available features. Condition Monitoring Error detects significant changes that cause errors in storing and managing data. Storage layer also provides data management services, such as traffic analysis, compression, virtualization, security, and replication etc., through the use of tools, policies and processes. This layer also provides database upgrades when some major changes are made in the database structure or between different releases. Our solution is based on improved Needham-Schroeder Protocol as described below.

6. Improved Needham-Schroeder Protocol

Needham-Schroeder protocol is one of the most popular authentication protocols that involve two participants. The protocol uses public-key to achieve authentication between the two participants with the help of authentication center. It is regarded as the seminal protocol for public-key authentication and has been used as the model for most key encryption systems to date.

The proposed system is used to secure the link between the User (*U*), Sever (*S*), and the certification authority (*CA*). The Certification Authority is assumed to be trusted by all the parties involved in the communication. *CA* has pairs of public key and secret key (P_{CA}, S_{CA}) and a session key shared with users (*U*) and the server (*S*) K_U and K_S respectively. A certificate and a session key are issued to every user at the time of subscription;the certificate contains ID and some credentials about the server signed by *CA* while the session keys must be changed from time to time. Furthermore, since the *S* is partially trusted by the *CA*, it is good enough if *S* can act as a mediator between *U* and *CA* while *CA* acts as an authenticator.

When describing the protocols, we use the following abstract notation:

U:	*User*
S:	*Server*
CA:	*Certification Authority*
P_{CA}	*CA's Public Key*
S_{CA}	*CAs Secret Key*
K_U	*Session Key (shared session key between U and CA)*
K_S	*Session Key (shared session key between S and CA)*
P_S	*S's Public Key*
P_U	*U's Public Key*
Un	*Nonce generated by U*
Sn	*Nonce generated by S*

The process of authentication in the protocol is as follows:

1. $U \Rightarrow CA: P_S$ (*U* request *S's* public-key from *CA*)
2. $CA \Rightarrow U:\{K_S, P_S\}S_{CA}$ (*CA* sends *S's* public-key and ID to *U*, singed it with its digital signature)
3. $U \Rightarrow S$: $\{Un, P_U\}K_S$ (*U* sends a nonce and its ID to *S*)
4. $S \Rightarrow CA: P_U$ (*S* request *U's* public-key from *CA*)
5. $CA \Rightarrow S: \{K_U, P_U\}P_{CA}$ (*CA* sends *U's* public-key and ID to *S*)
6. $S \Rightarrow U: \{Un, Sn\}K_U$ (*S* generates a nonce and forward it together with *U's* nonce encrypting the message with *U's* public-key)
7. $U \Rightarrow S: \{Sn\}K_S$ (*U* sends back *S's* nonce to *S* encrypted under *S's* public-key)

According to (Burrows, 1990), *Un*, and *Sn* serve not only as nonces, but also as authentication. It was discovered by Lowe (1995) that the protocol is vulnerable to attack. Assuming an intruder *T* is masquerading, a simple example is given below.

The following additional notations will be used:

T:	*Intruder (pretending to be S)*
K_T	*Session Key (shared session key between T and CA)*
P_T	*T's Public Key*
K_{UC}	*Shared Session Key (between U and CA encrypted with CA private key)*
Un1	*First Nonce generated by U*
Un2	*Second Nonce generated by U*
Sn1	*First Nonce generated by S*
Sn2	*Second Nonce generated by S*

3(a). $U \Rightarrow T$: $\{Un, P_T\}K_T$(*U* initiate communication with *T*)

3(b). $T \Rightarrow S$:$\{Un, P_T\}K_S$ (*T* initiate communication with *S* using *Un*)

6(a). $S \Rightarrow U$: $\{Un, Sn\}K_U$ (*S* responds to *U*)

7(a). $U \Rightarrow T$: $\{Sn\}K_T$(*U* thinking that the previous message is a response from *T*, *T* now can get *Sn* to impersonate *S*)

7(b) $T \Rightarrow S$: $\{Sn\}K_S$(*T* complete the protocol with *S*)

If *Un* and *Sn* are used as authentication, *T* now has the ability to impersonate *U* to *S* for the rest of the session, although *T* cannot read messages encrypted under *S's* key. Even if digital signature are used for authentication, and *T* cannot impersonate *U*, *T* has still manage to get *U* and *S* in an inconsistent state in which *S* thinks that *U* has initiate communication with it when in fact it has not. The attack on Needham-Schroeder's protocol is depicted in figure 3 below.

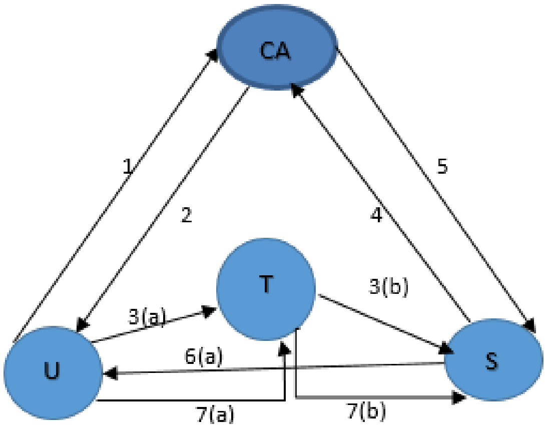

Figure 3. Attack on Needham-Schroeder's Protocol(Munir,K. 2019)

We have shown that Needham-Schroeder's protocol is vulnerable to impersonation by intruder *T*. We will now solve the problem and show how it can be implemented in our scheme. The improved protocol can be summarized in the following 7 steps and figure 4 below:

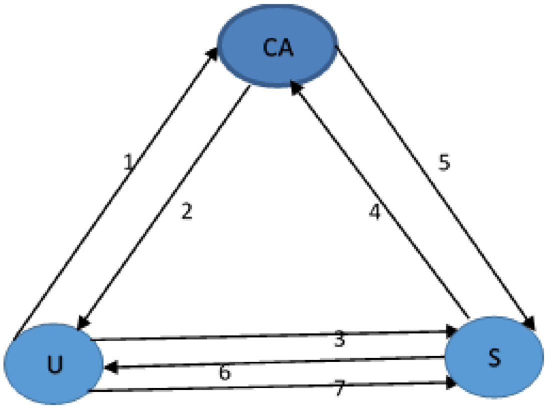

Figure 4. Improved Needham-Schroeder's Protocol(Munir,K. 2019)

1. $U \Rightarrow CA{:}\{T_B , P_T, Un1)P_{CA}$
2. $CA \Rightarrow U{:}\{(Un1, K_S, T_S)S_{CA}\}K_{UC}$
3. $U \Rightarrow S$: $\{Un2, P_T\}K_S$
4. $S \Rightarrow CA$: $\{T_S, Sn1, P_T\}\ P_{CA}$
5. $CA \Rightarrow S$: $\{(Sn1, K_U, P_T)S_{CA}\ \}K_S$
6. $S \Rightarrow U$: $\{T_S, Un2, Sn2\}K_U$
7. $U \Rightarrow S$: $\{Sn2\}K_S$

7. EXPLANATION

Note that step 1 and 2 will prevent T from misleading U since they are encrypted with *CA* private key and a shared key between *U* and *CA* (K_{UC}) hence, *T* cannot see the ID of *S* and hence cannot forward message 3(b) to *S*. In fact, even if message *T* was encrypted with *T's* public key instead of P_{CA}, *T* will not be able to generate K_{UC} since the key is only shared between *U* and *CA*. Sending messages 3(a) and 3(b) will not do any harm to the protocol unless *T* can send a nonce equivalent to *Sn1* in step 4, and this is impossible due to the random property of nonces even if *T* manage to get an old nonce sent by *S*. The use of nonces*Un1*, *Un2*, *Sn1*, *Sn2* will ensure privacy and protect both *S* and *U* from reply attack by *T*. Messages in steps 5, 6, and 7 can only be decrypted by *S* and *U* accordingly, therefore even if *T* can get the messages he/she cannot decrypt the content. The overall protocol is given in the Figure 4above.

8. CONCLUSION

In this paper, we have presented authentication scheme for DBaaS. We have described its components, discussed existing solutions and identified possible approaches to deal with different security issues related to the DBaaS. In our scheme, a new authentication protocol by using Needham-Schroeder protocol is proposed. In comparison with Needham-Schroeder's scheme, this scheme withstands some of the limitations associated with the Needham's scheme to avoid popular attacks including replay attack, and impersonation attack, and also this scheme is efficient in terms of communication and computation cost.Database as a Service (DBaaS) is an increasingly popular Cloud service model, with attractive features like scalability, pay-as-you-go model and cost reduction that make it a perfect fit for most organizations. However, no extensive research work has been done which meticulously covers each and every aspect of DBaaS. Data storage security in Cloud is a domain which is full of challenges and is of paramount importance as customers do not want to lose their data at any cost. There is a need for effective strategies, proper measurements and methodologies to control this problem by having mature practices in the form of secure architectures to make DBaaS platform more secure, and ultimately, widely-adopted.

9. FUTURE RESEARCH

The desired work to be done in future is an attempt to remove other two well-known possible attacks on Needham-Schroeder protocol; these attacks are, server spoofing attack and stolen-verifier attack on the protocol. The spoofing attack is when an intruder impersonates another device, server or user on a network in order to launch attacks against network hosts, steal data, spread malware or bypass access controls. There are several different types of spoofing attacks that malicious parties can use to accomplish this. While stolen-verifier attack (SV attack) is when an adversary steals verification data from the server in the current or past authentication sessions. Here, the verification data does not include secret keys used with XOR operation or an encryption function. She/he generates communication data using the stolen data and sends them to the server. If it succeeds, she/he impersonates a legal user from the next authentication session.

REFERENCES

[1] Munir K. (2019) Authentication Model for Mobile Cloud Computing Database Service. In: Zbakh M., Essaaidi M., Manneback P., Rong C. (eds) Cloud Computing and Big Data: Technologies, Applications and Security. CloudTech 2017. Lecture Notes in Networks and Systems, vol 49. Springer, Cham

[2] Schwartz, B. (2015, October 15). How to choose a DBaaS for Database Management. Retrieved from http://readwrite.com/2015/10/05/database-as-a-service-tips-choosing-dbaas/ 2016, November 05).

[3] Oracle Corporation,(2011). Database as a Service: Reference Architecture – An Overview.

[4] Krishna, K., Roger, L. (2012).Database as a Service (DBaaS) using Enterprise Manager 12c,Oracle Open World.

[5] Luca, F., Michele, C., &Mirco, M. (2012). Supporting security and consistency for Cloud database, Cyberspace Safety and Security:Lecture Notes in Computer Science, Volume 7672, pp. 179-193.

[6] Cong, W., Sherman, S.M.C., Qian, W., Kui, R., &Wenjing, L. (2013). Privacy Preserving Public Auditing for Secure Cloud Storage, IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 2, pp. 362-375.

[7] Jia, W., Zhu, H., Cao, Z., Wei, L.,&Lin, X. (2011). SDSM: a securedata service mechanism imobile cloud computing:ProceedingIEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, Shanghai, China.

[8] Huang, D., Zhou, Z., Xu, L.,Xing, T.,&Zhong, Y. (2011). Secure data processing framework for mobilecloud computing:Proceeding IEEE INFOCOM Workshop on Cloud Computing, INFOCOM '11, Shanghai, China.

[9] Hsueh, S.C., Lin, J.Y.,&Lin, M.Y. (2011). Secure cloud storage for conventional data archive of smart phones:Proceeding 15th IEEE International Symposium on Consumer Electronics ,ISCE '11, Singapore.

[10] Nithiavathy, R. (2013). Data Integrity and Data Dynamics with Secure Storage Service in Cloud:Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, IEEE,pp. 125-130.

[11] Qingji, Z., Shouhuai, X., &Giuseppe, A. (2012). Efficient Query Integrity for Outsourced Dynamic Databases, CCSW'12, Raleigh, North Carolina, USA.

[12] Ferretti, L., Colajanni, M., &Marchetti, M. (2012). Supporting security and consistency for Cloud database, Cyberspace Safety and Security:Lecture Notes in Computer Science Volume 7672, pp. 179-193.

[13] Maciej, B., Gracjan, J., Michał, J., Stanisław, J.,Tomasz, J., Norbert, M., Rafal, M., Adam, Z.&Sławomir, Z. (2013).National Data Storage 2: Secure Storage Cloud with Efficient and Easy. Data Access.

[14] Hacigumus, H., Iyer, B., Li, C.,Mehrotra,S. (2004). Efficient Execution of Aggregation Queries over Encrypted Relational Databases:Proc. of the 9th International Conference on Database Systems for Advanced Applications (DASFAA'04),Jeju Island, Korea, pp. 125–136.

[15] Oracle (2016).Oracle MAA Reference Architectures. Retrieved October 30, 2016, from http://www.oracle.com/technetwork/database/availability/maa-reference-architectures-2244929.pdf

[16] Lowe, G. (1995). An Attack on the Needham-Schroeder Public-Key Protocol: Information processing Letters, Vol 56, pp. 131-133

[17] Munir, K. (2015). Security Model for Cloud Database as a Service (DBaaS): IEEE Proceedings of the International Conference on Cloud Computing Technologies and Applications - CLOUDTECH 2015,. pp. 1-5, ISBN : 978-1-4673-8148-2

[18] Hexatier Survey. (2016).Database as a Service (DBaaS) Security Research 2016. .Retrieved November 05, 2016, from https://cdn2.hubspot.net/hubfs/1759710/Hexatier-Survey-2016.pdf

[19] Al-Rousan, T. (2015). Cloud Computing for Global Software Development:Opportunities and Challenges. International Journal of Cloud Applications and Computing, 5(1), 58-68

[20] Jouini,M. &Rabai, L. (2012). A Security Framework for SecureCloud Computing Environments.International Journal of Cloud Applications and Computing, 2(3), 1-25,

[21] Asija, R &Nallusamy, R. (2016). Healthcare SaaS Based on a Data Model with Built-In Security and Privacy.International Journal of Cloud Applications and Computing, Volume 6, Issue 3

[22] Nagar, N &Suman, U. (2016). Analyzing Virtualization Vulnerabilitiesand Design a Secure Cloud Environment to Prevent from XSS Attack.International Journal of Cloud Applications and Computing.Volume 6, Issue 1

[23] Alhaj, A., Aljawarneh, S., Masadeh,S,. &Abu-Taieh, E. (2013). A Secure Data Transmission Mechanism for Cloud Outsourced Data. 3(1), 34-43

[24] Alizadeh, M.,Abolfazli, S., Zamani, M.,Baharun, S., Sakurai, K.,(2016). Authentication in mobile cloud computing: A survey.Journal of Network and Computer Applications, Volume 61, Pages 59-80.

[25] Mohammed A. A., Alice S. Li, Ben, S.,, &Eric, P., .Multi-Cloud Data Management using Shamir's Secret Sharing and Quantum Byzantine Agreement Schemes.International Journal of Cloud Applications and Computing, 5(3), 35-52.

AUTHORS

KashifMunir received his BSc degree in Mathematics and Physics from Islamia University Bahawalpur, Pakistan in 1999. He received his MSc degree in Information Technology from University Sains Malaysia in 2001. He also obtained another MS degree in Software Engineering from University of Malaya, Malaysia in 2005. He completed his PhD in Informatics from Malaysia University of Science and Technology, Malaysia. His research interests are in the areas of Cloud Computing Security, Software Engineering, and Project Management. He has published journal, conference papers and book chapters.

KashifMunir has been in the field of higher education since 2002. After an initial teaching experience with courses in Stamford College, Malaysia for around four years, he later relocated to Saudi Arabia. He worked with King Fahd University of Petroleum and Minerals, KSA from September 2006 till December 2014. He moved into University of Hafr Al-Batin, KSA in January 2015.

KashifMunir is a researcher and published author/editor of 4 books on cloud computing including subjects such as Security in Cloud Computing, Mobile Cloud and Green Enterprises, (https://www.amazon.com/Kashif-Munir/e/B079KP1LFJ).

Lawan A. Mohammad, Holds a PhD degree in computer and communication systems engineering from University Putra Malaysia. Research interest include smartcard security, authentication protocols, wireless and mobile security, biometrics, mathematical programming and e-learning