Criptografía en bases de datos en cloud computing.
Cryptography in databases in cloud computing.

Deivis Eduard Ramírez Martínez, Higinio Mora Mora
Universidad de Alicante, Alicante - España
derm2@alu.ua.es; hmora@ua.es

Resumen- Los responsables de informática de las empresas que están pensando migrar sus sistemas de cómputo a la nube tienen sus reservas con respecto a la seguridad y la confiabilidad de los servicios basados en la nube, éstos aún no están plenamente convencidos de que entregar datos sensibles de las empresas o de sus clientes sea buena idea, en este contexto el uso de los sistemas de cifrado, y en especial los esquemas de cifrado homomórficos son de gran utilidad, ya que las operaciones realizadas en el proveedor cloud se realizan con la información cifrada, brindando así un nivel de confiabilidad y seguridad a las bases de datos frente a posibles ataques tanto internos como externos en el cloud computing. En el presente trabajo se propone un esquema para proteger los diferentes atributos de la información (confidencialidad, integridad y autenticación) almacenada en una BD en el Cloud.

Palabras Clave: Computación en la nube, criptografía, homomorfismo, seguridad, bases de datos

Abstract- The IT managers of companies who are considering migrating their systems to the cloud computing have their reservations about the security and reliability of cloud-based services, these are not yet fully convinced that deliver sensitive data companies or their clients is a good idea, in this context the use of encryption systems, in particular homomorphic encryption schemes are useful, since the operations in the cloud provider are made with the encrypted information, providing a level of reliability and safety databases from attacks as well as internal and external in cloud computing. This paper proposes a scheme to protect the different attributes of information (confidentiality, integrity and authentication), stored in a BD in the Cloud.

Keywords: Cloud computing, cryptography, homomorphism, security, databases.

*Autor para correspondencia.
Correo electrónico: nely_ca_be@hotmail.com (Deivis Eduard Ramírez Martínez).
La revisión por pares es responsabilidad de la Universidad de Santander.

Forma de citar: D. E. Ramírez Martínez y H. Mora Mora, “Criptografía en bases de datos en cloud computing”, Aibi revista de investigación, administración e ingeniería, vol. 2, no. 1, pp. 45-54 2014.

I. INTRODUCCIÓN

Expertos en el área de las Tecnologías de la Información y Comunicación (TICs), sostienen que el futuro en los sistemas de computación experimentará un cambio radical en su uso, sin importar el tamaño ni la actividad de la empresa a la que se esté dando soporte en los sistemas de cómputo. Los rápidos avances en las TICs, la globalización de la información, la necesidad de acceder a dicha información desde diferentes lugares y mediante diferentes dispositivos de conectividad hacen que se desarrollen términos como los de computación en la nube o *Cloud Computing*. El término de computación en la nube tiene sus orígenes en los años 90 cuando algunas compañías de telecomunicaciones empezaron a ofrecer sus sistemas de redes a través de Internet permitiéndole al usuario el acceso a determinadas aplicaciones en sus sistemas de cómputo. El National Institute of Standards and Technology (NIST) define el termino de *Cloud Computing* como *un modelo para habilitar un cómodo acceso en red bajo demanda a una fuente compartida de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se puede conformar, ampliar y proveer rápidamente con un esfuerzo administrativo mínimo o una interacción mínima con el proveedor de servicios*[1].

Existen dos tipos de nubes, las basadas en el modelo de implementación y las basadas en el modelo de servicio [2]. El modelo de implementación indica la ubicación de la nube y para el propósito que fue creada. Privada, pública, comunitaria e híbrida son los modelos de implementación. Los modelos de servicio describen el tipo de servicio que está ofreciendo el proveedor.

Actualmente las características que suelen tener los sistemas de *Cloud Computing* son: servicio bajo demanda, agrupación de recursos, servicio a medida, ampliación rápida. Los modelos de servicios que se están ofreciendo en la nube se pueden clasificar de tres formas según lo definido por el NIST en [1]:

- Software como servicio (software-as-a-service -SAAS). En este modelo el proveedor del servicio Cloud, ofrece al cliente una variedad de aplicaciones que se ejecutan en la infraestructura Cloud. Dichas aplicaciones están disponibles para varios usuarios -el número de usuarios varía según la cantidad de usuarios contratados- a través de una interfaz web-. El usuario no podrá administrar o controlar la infraestructura Cloud (sistemas operativos, servidores, sistemas de almacenamiento, redes, aplicaciones, etc.)
- Plataforma como servicio (platform-as a-service -PAAS). Las capacidades suministradas al cliente están desplegadas sobre la infraestructura Cloud, el cliente puede crear o adquirir aplicaciones creadas con lenguajes o herramientas soportadas por el proveedor. El cliente no podrá controlar o administrar la infraestructura Cloud, componentes de red, sistemas operativos, servidores, o sistemas de almacenamiento. A diferencia del modelo SAAS, el usuario tiene el control sobre las aplicaciones implementadas y posiblemente en la configuración del entorno.

Infraestructura como servicio (infrastructure-as-a-service - IAAS). El servicio suministrado al cliente es para abastecerlo de procesamiento, almacenamiento, redes, y otros recursos de computación fundamental, donde el cliente tiene la disponibilidad de ejecutar diferentes clases de software que pueden incluir aplicaciones y sistemas operativos. El cliente no puede controlar la infraestructura Cloud, pero tiene el control sobre los sistemas operativos, almacenamiento, desarrollo de aplicaciones y posiblemente control limitado de componentes de red. El número de aplicaciones ofrecidas ha ido en aumento y en la actualidad se pueden encontrar una variedad de herramientas diseñadas para operar sobre este tipo de plataforma tecnológica. Está quedando atrás el modelo en que las empresas tienen que disponer de un área de sistemas o centros de cómputo, los cuáles son los encargados de todo el proceso y desarrollo informático de la empresa que puede ir desde la planificación anual de actualización y la prevención de sistemas de software hasta la adquisición de equipos. El manejo de los sistemas de computación desvía la atención de los aspectos centrales del negocio y de ahí que ésta sea una de las razones que llevan a las empresas a migrar al modelo de *Cloud Computing*, al no tener que preocuparse por estas tareas las compañías pueden asignar más tiempo y recursos para alcanzar los objetivos y planes de negocio de las empresas. Los beneficios que suelen obtenerse al migrar los sistemas informáticos a la nube son: reducción de costes, inmediatez, flexibilidad, disponibilidad, escalabilidad, independencia de dispositivos, localización y eficiencia.

Algunas de las desventajas que se pueden encontrar en el Cloud Computing son las siguientes: dependencia de la conectividad (sin conexión a Internet no se puede acceder) y no existe una extensa cantidad de servicios disponibles para cubrir la demanda actual y la seguridad.

Uno de los inconvenientes que existe al utilizar estos modelos de servicio de Cloud Computing es la realización de diferentes operaciones o cálculos con la información sin cifrar (texto claro), el proveedor de servicios de Cloud podrá conocer y tendrá acceso a la información, incluso a los resultados obtenidos de dichos cálculos antes que los usuarios del servicio. Los responsables de informática de las empresas que están pensando migrar sus sistemas de cómputo a la nube tienen sus reservas con respecto a la seguridad y la confiabilidad de los servicios basados en la nube, estos aún no están plenamente convencidos de que entregar datos sensibles de las empresas, o de sus clientes, sea buena idea, por más reputación, reconocimiento y experiencia que tengan las grandes empresas que actualmente están ofreciendo estos servicios. Es aquí donde surge el siguiente interrogante: *¿existe alguna forma o sistema de enviar la información al Cloud, que permita hacer cálculos y operaciones con la información sin que la seguridad y confiabilidad se vea afectada?* Si disponemos de un sistema o arquitectura que permita trabajar bajo estas condiciones se podría administrar satisfactoriamente el problema de la seguridad de la información y así sacar ventaja de los servicios y beneficios de la computación en la nube. En este contexto es donde algunos sistemas de cifrados entran a jugar un papel importante. El uso más común de cifrar es proveer privacidad y confidencialidad para esconder toda la información útil de un texto plano. En la actualidad se encuentran en desarrollo técnicas que buscan dar solución al problema de seguridad y confidencialidad, la criptografía homomórfica es una de ellas.

La consideración de la información como un bien preciado que debe protegerse viene cambiando hace unos pocos años, debido a la reciente globalización de las comunicaciones, que ha incrementado de gran forma el flujo de la información. Muchos son los factores que intervienen en la seguridad de un sistema de base de datos. Las medidas de seguridad a adoptar pueden ser de tipo legal, administrativo, físico o lógico. Uno de los principales objetivos de los sistemas de seguridad de la información es proteger los datos de los cuales se abastecen los sistemas y aplicaciones, por lo tanto se debe diseñar una política de seguridad que permita alcanzar ciertos objetivos que se deben tener al momento de desarrollar una aplicación de una base de datos segura.

En el presente trabajo se propone un esquema para proteger los diferentes atributos de la información (confidencialidad, integridad

y autenticación) almacenada en una BD externalizada a través del modelo de servicio SaaS. Dentro de las categorías de modelo SaaS existentes se utilizará el modelo DAS por sus siglas en inglés (database as a service). Los objetivos que se buscan alcanzar con el esquema propuesto son: (1) que la información no pueda ser accedida por un usuario no autorizado (autenticación), (2) la información no será modificada por un usuario no autorizado (integridad) y (3) que la información confidencial no debe filtrarse a un usuario no autorizado (confidencialidad) de accesos/interceptaciones de la información de ataques internos o externos. Se construirá un prototipo para verificar la propuesta y se mostrará un caso de uso del modelo.

II. ESTADO DEL ARTE

Diversos grupos de investigación se encuentran trabajando en esquemas o técnicas que permitan brindar una mejor protección a las BD. Estos se centran en temas como detección de intrusos, control de acceso y políticas de seguridad sobre cómo usar la información, han sido algunos de los trabajos propuestos. Una de las medidas de seguridad usadas para proteger los sistemas de BD contra ataques internos o externos, durante su transmisión o en el proceso de almacenamiento es el uso de la criptografía.

Transmisiones seguras, controles de accesos y almacenamiento seguro de la información son algunos de los campos donde los esquemas de cifrado ha tenido un mayor uso. Actualmente, prácticamente todos los navegadores brindan a los usuarios canales de transmisión seguro que hacen uso de la criptografía a través de protocolos como SSL (Secure Socket Layer) [13] o (TransportLayer Security) [14].

Los algoritmos de los esquemas criptográficos se clasifican normalmente en esquemas de clave privada (o simétricos) y esquemas de clave pública (o asimétricos). En los primeros existe un emisor y un receptor de un mensaje que necesitan compartir información de forma segura a través de un canal inseguro. Para ello utilizan una clave única y común para cifrar/descifrar el mensaje, de ahí la importancia que dicha clave sea conocida solo por el emisor y el receptor del mensaje que se quiere proteger [11]. Los esquemas criptográficos de clave pública tienen un funcionamiento diferente, utilizan dos claves. Una para cifrar que es de conocimiento público y otra para descifrar que solo debe ser conocida por el usuario emisor de mensaje o información a proteger [12].

Cantidades ingentes de información son almacenadas y procesadas en los servidores de los proveedores cloud. Los sistemas de gestión de base de datos relacionales (SGBDR) juegan un papel muy importante en los servicios ofrecidos en el cloud. Los trabajos de investigación presentados que hacen uso de la criptografía en las de BD suelen clasificarse bajo dos escenarios [15]: 1) búsqueda basada en palabras claves en documentos de texto cifrados [16-18] y 2) evaluación de consultas en BDs relacionales cifradas [19-22].

George I. Davida et al. presentan un esquema de cifrado de BD que hace uso de los números primos [27]. En el mismo trabajo proponen, a su vez, un esquema basado en el teorema de residuo chino para cifrar BD usando un subconjunto de llaves para descifrar todo. También hacen uso de MACs (Mesage Authentication Codes) para conocer si la información ha sido vulnerada. En este esquema la información es almacenada en la BD de forma cifrada, pero para que el SGBDR pueda ejecutar la consulta debe descifrarla antes de ejecutar la query (consulta).

Dawn Xiaodong Song et al. presentan un esquema que tiene un gran número de opciones de búsqueda de palabras en texto cifrado [16]. Las búsquedas se pueden realizar en forma secuencial, o través de búsquedas de forma indexada.

Min Wang et al. proponen un esquema para un sistema gestor de BD relacionales (SGBDR) basado en un diccionario de seguridad [28]. En los diccionarios de seguridad, se tiene información importante de la BD como tablas, vistas, usuarios y contraseñas. Su propuesta se enfoca en la autenticidad mediante contraseñas, la mayoría de las consultas son rápidas debido a la mayoría de los campos se tiene en texto claro. Una gran desventaja se presenta al no tener todos los atributos ocultos permitiendo a un atacante acceder a información en texto claro en caso de un acceso no autorizado. Otra desventaja de este esquema se presenta cuando un usuario pierde su clave ya que no podrá recuperar completamente su información.

Otros trabajos buscan salvaguardar los datos por otras vías. Por un lado, Radek Vingralek et al. presentan una arquitectura que protege los datos contra ataques accidentales e intencionales [29]. En los primeros se generan actualizaciones constantes de datos y en los segundos se protege la información mediante el uso de criptografía, además de autenticar los datos cifrados. Luc Bouganim et al. presentan una arquitectura de hardware y software para dar seguridad a la información, utilizando una tarjeta inteligente en la que desarrolla una aplicación para cada persona o usuario de la BD, el objetivo principal de esta arquitectura es fungir como traductor de consultas [30]. Hakan Hacigümus et al. desarrollan e implantan un servicio de BD en Internet llamado NetDB2 construido sobre DB2 que suministra herramientas para desarrollar aplicaciones, crear y cargar tablas y procesar consultas [31]. Para esto realizaron pruebas con algoritmos de cifrado a nivel de software y hardware. En las pruebas de nivel de software se seleccionaron varios campos de una tabla en una BD cifrada, los resultados mostraron que al rescribir las consultas de una forma adecuada el tiempo de evaluación de la consulta era reducido. A nivel de hardware se toman todas las filas que están cifradas en la BD. Los resultados muestran una disminución drástica en los tiempos de ejecución de la consulta del cifrado a nivel se software, para mejorar el rendimiento aún más sugieren un cifrado en hardware a nivel de páginas con la ayuda de un modelo de estimación donde evalúan la sobrecarga del cifrado.

Anthony Harrington et al. proponen un nuevo mecanismo de control de acceso llamado control de acceso criptográfico [32], que depende exclusivamente de la criptografía para garantizar la confidencialidad e integridad de objetos (archivos) almacenados en servidores pocos confiables. El control de acceso se realiza implícitamente en la máquina cliente y de manera descentralizada.

Gagan Agrawal et al. presentan un esquema en el cuál la BD se fragmenta en dos servidores, con la condición que no tengan comunicación entre ellos [35]. La idea principal es dividir la tabla en dos servidores usando fragmentación vertical de la tabla, donde las llaves de cifrado pueden ir en ambos servidores y ciertos atributos pueden ser replicados en ambos servidores, si así se requiere en ambas entidades. Zheng-Fei Wang et al. proponen una arquitectura que puede ser implementada para realizar consultas SQL con datos de caracteres cifrados [36]. Su enfoque es práctico y solo necesita adicionar un módulo de descifrado/cifrado entre la aplicación del usuario y el SGBD. Para ello, utilizan bits como tipo de datos en los campos índices de las tablas, permitiendo así una reducción del 75% del tiempo de respuesta en comparación con el método tradicional de consulta.

Sergei Evdokimov et al. muestran un modelo y un esquema de cifrado probablemente seguro contra ataques de texto plano seleccionado y contra un ataque a posteriori de texto cifrado seleccionado [40]. El esquema propuesto permite realizar

eficientemente las operaciones de consulta en una BD sin afectar el tiempo de respuesta.

Sunil Sanka, et al. proponen un esquema basado en un conjunto de protocolos de seguridad para proteger información almacenada en un proveedor cloud [41]. Para ello, emplean un enfoque combinado de control de acceso y protocolos de cifrado, basándose en el modelo de intercambio de claves de D-H para acceder de forma eficiente y segura a las BD en el cloud.

Un análisis de métodos de almacenamiento, cifrado y autenticación utilizando criptografía de clave pública son presentados por Song Y. Yan et al. en [42]. La seguridad de los métodos que analizan se basa en la inviabilidad de algunos problemas teóricos- numéricos como el problema de factorización de números enteros y el problema del residuo cuadrático.

La necesidad de realizar las diferentes operaciones y consultas en las BDs en las máquinas y servidores de los proveedores cloud hacen que la criptografía homomórfica entre a jugar un papel importante. Los orígenes de la criptografía homomórfica datan de los años setenta, cuando Ronald L. Rivest et al. [6] introducían formalmente el término de privacidad homomórfica, el cual se utilizó para llamar a las funciones criptográficas que permiten realizar diferentes operaciones con información cifrada sin necesidad de descifrarla. La criptografía homomórfica se define como *un sistema de cifrado con la propiedad adicional de que existe un algoritmo eficiente para calcular un cifrado de la suma o el producto, de dos mensajes, dada la clave pública y la encriptación de los mensajes, pero no los mensajes* [5]. Es decir, un esquema de cifrado homomórfico es un esquema de cifrado especial que permite la conversión de información que se encuentra en texto claro mediante algoritmos matemáticos a texto cifrado. Matemáticamente los esquemas de cifrado homomórficos permiten realizar operaciones complejas con el texto cifrado obteniendo de los mismos resultados como si estuviera aún en su forma original.

En un esquema de criptografía homomórfica las funciones de cifrado:

$$E(p) = c$$

Toma una entrada de texto plano *p* y nos devuelven un texto cifrado *c*, un conjunto de estas funciones, las cuales tienen la propiedad de que una operación realizada en *p* sea equivalente a otra operación realizada en *c*, son denominadas funciones de cifrado homomórfico. Los sistemas criptográficos cuya función de cifrado *E*, preserva la operación de suma:

$$E(a + b) = E(a) + E(b)$$

y la operación de multiplicación:

$$E(a * b) = E(a) * (b)$$

Son llamados sistemas criptográficos homomórficos completos (ya que preservan la estructura de anillo), mientras que los que preservan solo una de las operaciones son llamados sistemas criptográficos homomórficos parciales [10].

En sus inicios algunos de los esquemas de criptografía homomórfica propuestos solo soportaban una operación matemática. Estos esquemas soportaban las operaciones de la adición o la multiplicación pero no ambas al mismo tiempo [7-8], [23]. Después, en [24] se presenta un esquema capaz de realizar ambas operaciones al mismo tiempo, sus esquemas manejan un número arbitrario de sumas y solo una multiplicación. Posteriormente, Craig Gentry presenta dos trabajos [10], [25] en los cuales se desarrolla uno de los primeros sistemas completamente homomórfico (Fully Homomorphic Encryption - FHE), capaz de evaluar un número arbitrario de sumas y multiplicaciones (y de esta forma evaluar una función) con datos cifrados.

Los esquemas de criptografía basados en homomorfismo pueden ser determinísticos o probabilísticos [5]. En los probabilísticos, se obtienen diferentes textos cifrados para un mismo texto en claro dependiendo de una variable aleatoria y en los determinísticos el cifrado depende no sólo del texto en claro sino también de un valor elegido aleatoriamente por la persona que realiza el cifrado. El criptosistema RSA que fue desarrollado por Rivest, Shamir y Adleman [6], es un ejemplo de un esquema homomórfico determinístico y el criptosistema El Gamal [7] es un ejemplo de un esquema homomórfico probabilístico. Algunos otros esquemas de cifrado homomórfico son los presentados por Paillier en [8] donde propone un esquema que tiene una seguridad semántica, se dice que un sistema es semánticamente seguro si no es factible averiguar información alguna acerca del texto en claro (información sin cifrar) a partir del texto cifrado [5]. Damgard y Jurik presentan en [9] un esquema homomórfico probabilístico con propiedad aditiva. Akinwande muestra un estudio en [5] sobre los avances en los esquemas de cifrado homomórficos, donde llega a la conclusión que el esquema del Gamal ofrece el mejor nivel de seguridad en los esquemas de cifrado homomórficos, el esquema RSA es el que más rápido que trabaja a nivel general, pero el esquema de Paillier es el esquema de cifrado homomórfico probabilístico que tiene un mejor rendimiento a la hora de realizar el proceso de descifrado.

Como todos los esquemas y protocolos de cifrado los esquemas de cifrado homomórficos no están exentos de posibles ataques. Menezes, A et al. en [3] definen dos tipos de ataques para todo esquema de cifrado: a) ataques pasivos y b) ataques activos. En los primeros el atacante solo monitorea el canal de comunicación, en este tipo de ataque se pone en riesgo la confidencialidad de la información. En los segundos, el atacante intenta eliminar, añadir, o de alguna otra manera alterar la transmisión en el canal, amenazando la integridad y autenticación de datos, así como la confidencialidad. Réka Limbek et al. en [4] hacen un análisis más detallado de los posibles ataques con los que se puede romper la seguridad en los esquemas de cifrado homomórficos. Según Limbek, los esquemas homomórficos se pueden atacar de tres formas diferentes (todos los ataques son pasivos): a) ataque con solo texto cifrado, b) ataque con texto plano conocido y c) ataque con selección de texto (cifrado o plano).

Hakan Hacigümus et al. presentan un esquema de cifrado homomórfico que permite hacer consultas de agregación con campos cifrados bajo operaciones matemáticas como suma o multiplicación en BD cifradas [33]. Una de las debilidades de este esquema es no poder realizar todo tipo de consultas con la información cifrada, ya que solo permite consultas de agregación y la no autenticación de los datos. En el mismo año, Bala Iyer et al. realizan un análisis de cómo cifrar y almacenar información de forma segura en SGBDR en [34].

Gultekin Ozsoyoglu et al. publican un esquema de cifrado homomórfico para generar consultas en BD cifradas [37]. Se enfoca únicamente en los atributos numéricos reforzando las consultas y la seguridad en este tipo de atributos. Por otro lado, Zheng-Fei Wang et al. proponen un enfoque eficaz que permite hacer consultas con rapidez sobre caracteres cifrados y datos numéricos [38]. Para ello, hacen un análisis de tipo de información y dependiendo de dato que se presente adoptan diferentes métodos para almacenar y realizar consultas sobre los datos cifrados.

Einar Mykletun et al. presentan una alternativa a los esquemas de cifrado homomórfico para consultas de agregación en BD cifradas bajo el modelo DAS [39]. Su técnica reduce la sobrecarga computacional asociada a las consultas de agregación con la información cifrada, tanto del lado del cliente como del servidor. Adicionalmente, proponen una variante del modelo DAS a la que llaman DAS modelo mixto, donde algunos atributos son sensibles (y por lo tanto se almacenan cifrados) mientras que otros no son (y por lo tanto se quedan en el claro).

Aplicaciones de los esquemas de cifrado homomórficos son: protección de agentes móviles, computación multiparte, esquema de compartimiento de secretos, esquemas de umbrales, esquemas electorales, esquemas de marcas de agua, marcas digitales, esquemas de compromisos, y esquemas de subastas [26].

La Agencia de Investigación de Proyectos Avanzados en Inteligencia (Intelligence Advanced Research Projects Activity - IARPA) y la Agencia de Investigación de Proyectos Avanzados de Defensa (Defense Advanced Research Projects Agency - DARPA) se encuentran financiando programas, proyectos o grupos de investigación para que trabajen en la mejora del rendimiento del esquema de criptografía homomórfica. El programa financiado por IARPA llamado Security And Privacy Assurance Research (SPAR) [43] pretende desarrollar técnicas prácticas para el intercambio de datos seguros que brinden seguridad y privacidad a cada una de las partes haciendo uso de la criptografía homomórfica completa (Fully Homomorphic Encryption - FHE), en escenarios como sistemas de almacenamiento de datos de terceros, consulta a base de datos complejas, sistemas de suscripción y publicación, buzones de correos. Por su parte, el proyecto financiado por DARPA es llamado Programming Computation on Encrypted Data (PROCEED) [44], en él se busca encontrar una forma más eficiente y rápida de cifrar datos y que estos puedan ser utilizados y manipulados sin descifrar.

Existen otros proyectos, como Cloutage - Open Security Foundation - [45], donde se llevan registros de los incidentes relacionados con el Cloud Computing. En dicho proyecto se dividen los incidentes en cinco categorías: perdida de datos, vulnerabilidades, cortes de luz, ataques informáticos y autofallos. La pérdida de datos en la nube puede ser muy crítica para una organización. Esta criticidad se amplía, o se minimiza, dependiendo del fenómeno presentado, la perdida de información que se presente y del tiempo que tarde en restablecerse el servicio. Por lo tanto, todos los sistemas de gestión de información, bases de datos, etc., son componentes esenciales para el funcionamiento diario de toda empresa.

Hasta hace unos pocos años los responsables de informática en las empresas tenían muchas dudas y cuestionamientos sobre la migración de sus BD a un proveedor cloud. El cambio de modelos de software tradicionales a Internet ha tomado un gran impulso, empresas líderes del sector, como Microsoft, Amazon o Apple, entre otras, están apostando fuertemente a este modelo. La tendencia actual sobre el uso del *cloud computing* en las empresas grandes como las pequeñas y medianas irá en aumento. Los esquemas de cifrado son de vital importancia para que las empresas puedan implementar sus sistemas de información, procesamiento o almacenamiento alcanzando unos niveles óptimos de seguridad.

Diversos esquemas de seguridad que hacen uso de la criptografía han sido presentados en el estado del arte. Cada esquema propuesto tiene un enfoque o finalidad muy específica y funcionan bajo determinadas condiciones o restricciones. Algunos buscan cifrar objetos (archivos), otros cifrar BDs (parcialmente/totalmente). Existen propuestas que permiten realizar consultas con información cifrada para determinados campos o atributos en BDs (numérico/carácter); otros a proteger el canal de comunicación y el control de acceso, algunos a validar la autenticidad de la información. Pero no se encuentra un modelo general que aglutine esos avances en un único sistema, permitiendo así hacer uso de algunos de los diferentes desarrollos y líneas de investigación propuestos. El alcance del esquema que se propone en el presente trabajo es tratar de cubrir todos los aspectos citados anteriormente, de una forma general, haciendo uso de algunos de los métodos y líneas de actuación presentados, proponiendo un esquema general que brinde una seguridad óptima para poder externalizar los SGBD a un proveedor cloud sin que se vea afectada la seguridad de la información.

III. CRIPTOGRAFÍA EN BASES DE DATOS EN ENTORNOS CLOUD COMPUTING

El modelo de servicio cloud bajo el cual se propondrá el esquema será DAS planteado en [16], [19], [31], [46]. El esquema esbozado deberá cumplir con las operaciones básicas de todo SGBDR como son: consultar, eliminar, insertar, modificar y almacenar información, aprovechando la infraestructura establecida y ofrecida por el proveedor Cloud. Múltiples usuarios podrán acceder a la BD desde diferentes dispositivos de conexión para realizar las operaciones anteriormente mencionadas sin necesidad de descifrar la información.

Un modelo DAS consta de las siguientes entidades: a) titular de la información: encargado de producir la información y propietario de la misma, pero con limitaciones de recursos de almacenamiento y procesamiento, b) servidor: proveedor de servicio remoto, con gran capacidad de procesamiento, y almacenamiento de información que presta un servicio al titular de la información. c) cliente de la información: puede ser el mismo que el titular de los datos o si el propietario es una organización, podría tratarse de sus empleados o sus clientes [30].

Para el esquema que se propone en este trabajo nos basaremos en las propuestas realizadas en [33] y [47] para alcanzar la idea central del presente trabajo. El esquema planteado tiene tres procesos generales: proceso de cifrado, proceso de consulta con información cifrada y proceso de descifrado. Los procesos de cifrado y de descifrado son ejecutados por un intérprete en la máquina del usuario y el proceso de la consulta con la información cifrada la realiza el SGBD en el proveedor Cloud.

Para esto deberá hacerse uso de un intérprete cuya función será adaptar la consulta realizada por el usuario antes de ser enviada al Cloud, de tal manera que el SGBD pueda llevarla a cabo en el lado del proveedor Cloud sin necesidad de descifrarla. El proceso de adaptación de la consulta se haría en la máquina del cliente.

Para realizar las consultas tomaremos como base el trabajo propuesto en [33]. Se utilizará la arquitectura de un servicio de almacenamiento que trabaja con la información cifrada propuesto en [47] que consta de tres componentes: un Procesador de Datos (PD) que es el encargado de procesar los datos antes de enviarlos al proveedor Cloud, un Verificador de Datos (VD), que se encarga de comprobar que los datos almacenados en el Cloud no han sido manipulados y un Generador de Token (GT), el cual permite al proveedor Cloud recuperar partes de información del cliente, adicionalmente también funciona con un Generador de Credenciales (GC) que se encarga de generar una política de control de acceso para expedir credenciales para las diferentes partes del sistema que interactuarán, estas credenciales permiten cifrar y descifrar los datos de acuerdo a las políticas establecidas.

Para el esquema que proponemos en este trabajo solo haremos uso de los dos primeros componentes de dicha arquitectura. Ver esquema general de funcionamiento en la figura 1.

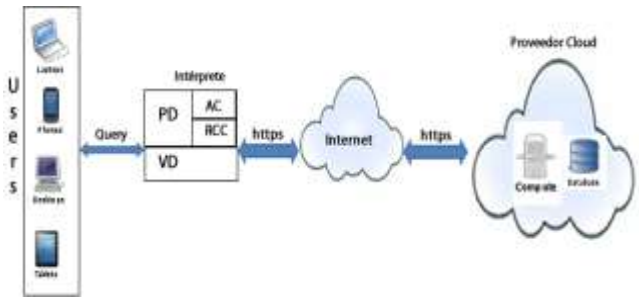

Fig. 1. Esquema general del modelo propuesto.

A continuación se detallan los aspectos más relevantes de este esquema.

Para la transmisión de la información se utilizará un canal de comunicación seguro a través de estándares de comunicaciones (TLS/SSL) que utilizan esquemas criptográficos que proporcionan canales de comunicaciones fiables entre la máquina del cliente, Internet y el proveedor cloud.

A. *Intérprete*

El intérprete es el encargado de preparar las consultas realizadas por el usuario antes de ser enviadas al proveedor cloud para que sean procesadas por el SGBD. Para usar el servicio la aplicación genera una clave criptográfica también conocida como clave maestra, se almacenará únicamente de manera local en la máquina del usuario. El usuario en ningún momento deberá compartir la clave generada con el proveedor Cloud. El intérprete está compuesto por el PD y VD y su funcionamiento general es el siguiente: cuando el usuario genere la consulta lo primero que hará la aplicación será ejecutar el intérprete, éste a su vez invocará al PD. El PD realizará un análisis de la consulta (AC) hecha por el usuario para determinar el tipo de consulta realizada (intervalos, agregación u operaciones matemáticas). Después hará una revisión de cada campo de la consulta (RCC) para determinar qué tipo de atributo tiene el campo (numérico o alfanumérico).

El proceso de cifrado sería de la siguiente forma: se recibe la consulta en texto claro la cual está constituida por varios campos (atributos) que son analizados uno a uno y a su vez son clasificados en numéricos y alfanuméricos. Si es numérico se cifra bajo dos algoritmos (homomórficos y preserva el orden). Si es alfanumérico se cifra con un esquema de cifrado por bloques y posteriormente se codifica en Base64. Este último paso es requerido para que el campo cifrado esté conformado únicamente por caracteres válidos que puedan ser almacenados en la BD. Es muy importante que el esquema de cifrado de bloque sea determinístico y por aspectos de rendimiento que sea simétrico. Los cifrados en bloques determinísticos permiten obtener siempre el mismo texto cifrado para un texto en claro, esto garantiza que el resultado cifrado del atributo que es la llave primaria en una tabla será exactamente igual al atributo que es la clave extranjera en otra tabla, permitiendo así mantener las relaciones.

Realizados los procesos anteriormente mencionados, el PD hace una adaptación de la consulta hecha por el usuario para determinados campos requeridos en el diseño de la BD – este proceso es adaptación es transparente para el usuario – y posteriormente envía la consulta al servidor Cloud para que sea procesada. En la figura 2 se puede ver el funcionamiento interno del PD del intérprete.

B. *Consultas*

Los algoritmos criptográficos por lo general buscan inyectar aleatoriedad a la información cifrada dificultando la labor inversa de descifrado a los posibles atacantes o criptoanálistas. Al trabajar bajo un esquema de BD relacional donde toda la información se encuentra cifrada, hay que tener especial cuidado en el momento de diseñar las consultas, y dependiendo del tipo de consulta que se realice deberá utilizarse un esquema homomórfico, u otro esquema de cifrado, para que las relaciones entre las tablas se mantengan.

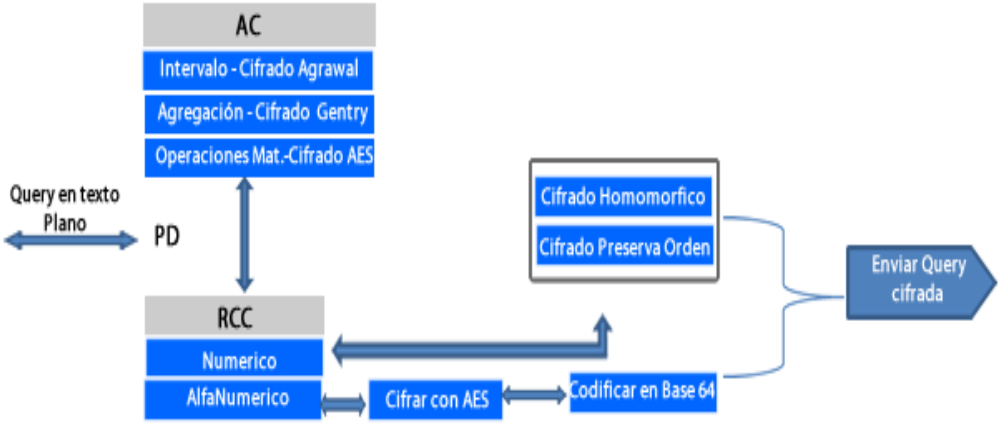

Fig. 2. Funcionamiento general del Procesador de datos

Los tipos de consultas son los siguientes: consultas de intervalos, consultas de agregación, consultas de operaciones matemáticas que se detallan a continuación.

B.1. Consultas de Intervalos

Se usan cuando se quiere obtener de un determinado rango un resultado, por lo tanto debe establecerse un límite inferior y otro superior. Para este tipo de consultas se debe utilizar un esquema de cifrado que permita preservar el orden numérico en un texto en claro. En la revisión bibliográfica se encontraron dos esquemas que tienen esa propiedad: Agrawal[48], y Boldyreva [49]. En el presente trabajo se utilizará Agrawal ya brinda mejores características y funcionalidades tales como: permite realizar operaciones de comparación que se aplican directamente sobre la información cifrada sin necesidad de descifrar los operandos, permitiendo así ejecutar sentencias de SQL como MAX, MIN y COUNT. Además puede ser integrada con BD existentes ya que ha sido diseñado para funcionar con las estructuras de indexación existentes, los resultados del proceso de consultas sobre los datos cifrados son exactos. Agrawal ya ha sido implementado en DB2 cuyos resultados de medición muestran que la sobrecarga de tiempo y espacio de este esquema son razonables para ser implementado en los sistemas reales. Un ejemplo de una consulta de intervalos sería: seleccionar todos los médicos cuyo salario es mayor o igual 1500 y menor o igual que 2200. La sentencia SQL para la anterior consulta sería:

```
SELECT NomMed, SalarioMed FROM Medicos WHERE
SalarioMed ≥ 1500 AND SalarioMed ≤ 2200
```

B.2. Consultas de Agregación

Operan sobre atributos numéricos y se usan cuando la consulta implica el análisis de varios datos para obtener un solo resultado. Para este tipo de consultas se utilizará un esquema de cifrado homomórfico. En el estado del arte se revisaron varios esquemas de cifrado homomórficos propuestos, sin embargo la mayoría trabajaban bajo la operación de la multiplicación. El objetivo del esquema propuesto en este trabajo es tratar de cubrir el mayor número de consultas posibles, por lo tanto es adecuado seleccionar un esquema homomórfico que involucrara las dos operaciones (suma y multiplicación). Esquemas de cifrado como RSA (homomórfico bajo la operación multiplicación) o Paillier [8] (homomórfico bajo la operación de la suma) eran buenas opciones ya que son esquemas de cifrado probados y reconocidos, pero teniendo en cuenta que uno de los objetivos que se pretende alcanzar es tratar de cubrir el mayor número de operaciones de consultas al servidor (suma y multiplicación) se trabajará con el esquema de Gentry [10], que es un esquema homomórfico completo. Un ejemplo de este tipo de consultas es: calcular el total de salario de los médicos. La consulta SQL para este tipo de consulta sería:

```
SELECT SUM (SalarioMed) FROM Medicos
```

B.3. Consultas de Operaciones Aritméticas

En algunas consultas se requiere utilizar operaciones matemáticas con la información numérica contenida en la base de datos. Para esto se hará uso de un cifrado en bloque determinístico que permita mantener la relaciones entre las tablas, ya que al obtener siempre el mixto texto cifrado para un texto en claro, garantiza que el resultado del atributo cifrado que es la llave primaria en una tabla será exactamente igual a la del atributo donde es una llave extranjera. Se utilizará el cifrado en bloque AES ya que es un esquema de cifrado muy utilizado e implementado en muchas aplicaciones, pero no es restrictivo. Hay opciones como IDEA o DES que se podrían utilizar también, lo importante es que sean cifrados en bloques determinísticos. Un ejemplo de este tipo de consultas es:

```
SELECT NomMed, SalarioMed*12 FROM Medicos
```

C. Ejemplo de Implementación

Los modelos de servicios en el Cloud Computing funcionan bajo el esquema de Cliente-Servidor, por lo tanto en el prototipo de esquema propuesto en este trabajo se utilizaran los siguientes componentes: un servidor web (Apache), un SGBD (MySQL) que soporta el estándar SQL, dos lenguajes de programación PHP (en el lado del servidor) y Javascript (en el lado del cliente), los componentes podrán instalarse y configurarse por separado o bajo la aplicación XAMPP, que además brinda soporte para el protocolo *Secure Sockets Layer* (SSL) que proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Un aspecto muy importante a tener en cuenta es la forma de realizar las consultas, al trabajar con toda la información cifrada se deberán tener algunas consideraciones en el momento de hacer ciertas consultas al sistema.

Los elementos básicos del esquema propuesto son: 1) Cliente: es el encargado de generar las consultas en texto claro y se conecta al proveedor cloud a través de diferentes medios de conectividad (ver figura 1). 2) Proveedor cloud: donde se encuentra almacenada y cifrada la BD. 3) Intérprete: es la interfaz encargada de realizar la consulta del cliente en texto claro de forma cifrada y que dicho proceso se realice de forma transparente para el cliente. Se ejecuta en la máquina del cliente y está compuesto por el PD (AC, RCC) y VD.

A continuación presentamos un ejemplo para ilustrar y explicar con más de detalle la forma en que se realizan las consultas. En la figura 3. podemos ver una BD básica que está compuesta por dos tablas: Médicos y Departamentos.

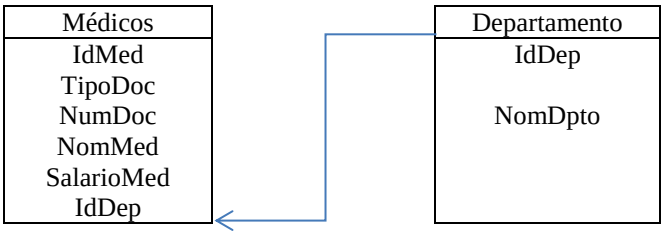

Fig. 3. Base de datos en claro.

Las operaciones de consulta permiten extraer registros de una tabla o más bajo ciertas condiciones y suelen tener los siguientes parámetros:

```
SELECT <atributos>, <función de agregación>
FROM <relaciones>
WHERE <predicados>
    GROUP BY <atributos de agrupación>
```

Una consulta básica podría ser: seleccionar el nombre y salario de todos los médicos:

```
SELECT NomMed, SalarioMed FROM Médicos
```

En la tabla 1 se muestran algunos datos de la tabla *Médicos* en claro y los atributos que contienen son: el identificador del médico (IdMed), tipo de documento (TipoDoc), número de documento (NumDoc), nombre médico (NomMed), salario médico (SalarioMed), y el identificador del departamento (IdDep) al que pertenece.

Las consultas generadas por el usuario se generan en texto claro y se realizan de forma transparente al usuario. Esta información debe ser guardada en el proveedor Cloud en forma cifrada. Dependiendo del tipo de atributo que contenga el campo se cifrará con un determinado esquema de cifrado (homomórfico, preserva el orden, bloques).

Para poder realizar consultas el diseño de la BD debe estar modificado de acuerdo a las posibles consultas que quiera ofrecer la aplicación. Partiendo del diseño de la tabla 1 y adaptándola para que se pueda ejecutar las consultas con la información cifrada la tabla1 quedaría de la siguiente manera ver tabla 2.

Donde los campos IdMedCH, SalarioMedCH, son los campos que se cifran bajo el esquema de cifrado homomórfico y los campos IdMedCPO, SalarioMedCPO, SalarioMedCPO serían los campos que se guardarían bajo esquemas de cifrado que preserva el orden. Los campos TipDocC, NumDocC, NomMedC se cifran con el esquema de cifrado AES, y después de aplicar el cifrado AES (cifrado en bloque determinístico que se usa para mantener la relación entre las tablas) se codifica en Base64.

En la tabla 3 podemos ver cómo queda la tabla 1 con sus respectivos campos después de ser modificada y adaptada para ejecutar las consultas con el texto cifrado en el lado de proveedor cloud.

Por ejemplo, al realizar la siguiente consulta basados en los datos contenidos en la tabla 1:

```
SELECT NomMed, FROM Medicos WHERE TipDoc =
NIE
```

El resultado de la anterior consulta en texto claro es el siguiente: *Pedro Flórez.* El objetivo es no revelar información al proveedor cloud, por lo tanto las operaciones de consultas y la información estarán cifradas.

El intérprete procesaría la consulta de la siguiente manera: a) Ejecutaría el AC y dependiendo del resultado sabría que esquemas de cifrado ejecutar en el proceso de RCC, adaptaría la consulta y después enviaría la consulta cifrada a la BD en el Cloud.

La consulta traducida y adaptada quedaría de la siguiente manera:

```
SELECT NomMedC, FROM Medicos WHERE TipDocC
= NIE
```

El resultado de la anterior consulta ejecutada en la máquina del proveedor cloud y antes de enviarse al cliente sería:

C'=E(Pedro Flórez) →BASE64 y su cifra sería la siguiente:

NUYgMjIgNDAgNzkgNzYgOTcgNzUgOTAgN0MgNEIgQTgg
QkMgOTggM0YgMjIgOUI=

Este resultado se enviaría a la máquina del cliente donde el intérprete ejecutaría el proceso de descifrado antes de mostrar el resultado al cliente como texto claro.

TABLA I.
TABLA MÉDICOS EN TEXTO CLARO

IdMed	TipDoc	NumDoc	NomMed	SalarioMed	IdDep
01	DNI	B3456879J	MaríaJimenez	2250	Odon
02	DNI	L8976076M	Mariano Zapatero	2070	Pedi
03	NIE	X8819090K	PedroFlorez	2650	Gene
04	PASAPORTE	PAM623677	FranciscoLopez	1550	Gene

TABLA II.
CAMPOS DE LA TABLA 1 CON LOS DIFERENTES ATRIBUTOS ADAPTADOS AL ESQUEMA PROPUESTO. CH= CIFRADO HOMOMÓRFICO, CPO=CIFRADO QUE PRESERVA EL ORDEN C=CIFRADO AES

IdMedCH	IdMedCPO	TipDocC	NumDocC	NomMedC	SalarioMedCH	SalarioMedCPO	IdDepC

TABLA III.
TABLA MÉDICOS CON INFORMACIÓN CIFRADA.

IdMedCH	IdMedCPO	TipDocC	NumDocC	NomMedC	SalarioMedCH	SalarioMedCPO	IdDepC
C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={[E(C] →BASE64}
C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={[E(C] →BASE64}
C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={ [E(C)] →BASE64}
C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'={ [E(C)] →BASE64}	C'_{CH}={E(c)}	C'_{CPO}={E(c)}	C'={ [E(C)] →BASE64}

IV. CONCLUSIONES

El Cloud Computing representa un cambio de paradigma en la informática, una nueva forma de ofrecer y vender servicios. En el presente trabajo se propuso un esquema experimental que busca proteger los diferentes atributos de la información (integridad, confidencialidad, autenticidad) haciendo uso de diferentes técnicas de cifrado en BDs, permitiendo externalizar SGBD en el Cloud Computing. Se mostró un prototipo para verificar el modelo propuesto con un caso de uso que aglutina diferentes técnicas y líneas de investigación de trabajos propuestos que hacen uso de la criptografía en BDs.

En las líneas de investigación abiertas se debe estudiar y buscar una forma de implementar alguno de los estándares de gestión de claves de cifrado como: IEEE 1619.3 o el KMIP de OASIS. El esquema propuesto basa su seguridad en el uso de diferentes esquemas de cifrado, y por lo tanto otra línea de investigación abierta sería realizar una implementación a nivel hardware implementando el esquema de cifrado homomórfico empleado para la mejora del rendimiento de las consultas. También se debe buscar una metodología de diseño de base de datos seguras, y que permitan implementar un modelo de gestión de claves.

REFERENCIAS

[1] Zhang Q., Cheng L., Boutaba R.: *Cloud computing: State of the art and research challenges. Journal of Internet Services and applications* 1, 7-18 (2010)

[2] Sosinsky, B.: *Cloud Computing Bible. Wiley Publishing*, Indianapolis (2011)

[3] Menezes, A. J., Van Oorschot, P. C., Vanstone S. A.: *Handbook of Applied Cryptography. Series: Discrete Mathematics and Its Applications.* CRC Press (1996)

[4] Limbek, R., Sziklai, P.: *Privacy homomorphisms. Scientific Association for Infocommunications*, vol. 6 pp. 37-42 (2004)

[5] Akinwande, M.: *Advances in homomorphic cryptosystems.* Journal of Universal Computer Science, vol. 15, pp. 506–522 (2009)

[6] Rivest, R., Adleman, L. and Dertouzos, M.: *On data banks and privacy homomorphisms. Foundations of Secure Computation.* pp. 169 - 177, Academic Press (1978)

[7] Elgamal, T.: *A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory*, vol. 31, pp. 469-472 (1985)

[8] Paillier, P: *Public-key cryptosystems based on composite degree residuosity classes. Advances in Cryptology EUROCRYPT'99,* of LNCS, vol. 1592 pp. 223-238. Springer, Verlag (1999)

[9] Damgard, I., Jurik, M.: *A Length-Flexible Threshold Cryptosystem with Applications.* In: Proceedings of the 8th Australasian Conference on Information Security and Privacy (ACISP 2003), LNCS 2727, Springer, New York, USA (2003)

[10] Gentry, C.: *Fully homomorphic encryption using ideal lattices.* In: Proceedings of the 41st annual symposium on Theory of computing, pp. 169–178. ACM press, New York (2009)

[11] Burke, J., McDonald, J. Austin, T.: *Architectural support for fast symmetric-key cryptography.* In: Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, USA, pp. 178-89. (2000)

[12] Blake-Wilson, S.: *Information security, mathematics, and public-key cryptography. Designs, Codes and Cryptography*, vol. 19, pp. 77-99. (2000)

[13] Freier, A., Karlton, P. and Kocher, P.: *The SSL Protocol Version 3.0, Internet.* (1996)

[14] Dierks, T. and Allen, C.: *The TLS Protocol - Version 1.0, Internet.* (1997)

[15] Hacigümüs, H., Hore, B., Iyer, B., Mehrotra, S.: *Search on Encrypted Data.* In :Secure Data Management in Decentralized Systems. vol. 33, pp 383-425. Springer US. (2007)

[16] Song, D., Wagner, D., Perrig, A.: *Practical Techniques for Search on Encrypted Data.* In: Security and Privacy Proceedings. IEEE Symposium on, pp.44-55. (2000)

[17] Chang, Y., Mitzenmacher, M.: *Privacy preserving keyword searches on remote encrypted data*. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) Applied Cryptography and Network Security. LNCS, vol. 3531, pp. 391-421. Springer, Heidelberg (2005)

[18] Golle, P., Staddon, J., Waters, B.: *Secure conjunctive keyword search over encrypted data.* In: Jakobsson, M., Yung, M., Zhou, J. (eds.) Applied Cryptography and Network Security. LNCS, vol. 3089, pp. 31-45. Springer, Heidelberg (2004)

[19] Hacigümüs, H.: *Privacy in Database-as-a-Service Model. Ph.D. Thesis, Department of Information and Computer Science, University of California, Irvine,* 2003.

[20] Bouganim, L. and Pucheral, P.: *Chip-Secured Data Access: Confidential Data on Untrusted Servers.* In: Proceedings of the 28th Very Large Data Bases Conference, pp. 131-142. Morgan Kaufmann, Hong Kong, China (2002)

[21] Hacigümüs, H., Iyer, B. and Mehrotra, S.: *Encrypted Database Integrity in Database Service Provider Model.* In: Database Service Provider Model. Certification and Security in E-Services pp. 165-174 (2002)

[22] Hore, B., Mehrotra, S., Tsudik, G.: *A Privacy-Preserving Index for Range Queries.* In: Proceedings of the Thirtieth international conference on Very Large Data Bases, vol. 30 pp. 720-731 Toronto, Canada (2004)

[23] Goldwasser, S., Micali, S.: *Probabilistic Encryption & How To Play Mental Poker Keeping Secret All Partial.* Computing pp. 365-377. (1982)

[24] Boneh, D., Goh, E. J., Nissim, K.: *Evaluating 2-DNF formulas on ciphertexts.* In: Kilian, J. (ed.) Theory of Cryptography. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)

[25] Gentry, C.: *Toward basing fully homomorphic encryption on worst-case hardness.* In: Rabin, T. Advances in Cryptology–CRYPTO 2010, LNCS, pp. 116–137. Springer, Heidelberg (2010)

[26] Rappe, D. K.: *Homomorphic Cryptosystems and their Applications.* Thesis, Cryptology ePrint Archive, Report (2006)

[27] Davida, G. I., Wells, D. L., Kam, J. B.: *A database encryption system with subkeys. ACM Trans.* Database Syst. vol. 6, pp.312-328 (1981)

[28] He, J., Wang, M.: Cryptography and relational database management systems. In: *Database Engineering & Applications, International Symposium on.* pp. 273-284. IEEE press (2001)

[29] Vingralek, R. Gnatdb: A small-footprint, *secure database system*. In: VLDB, pp. 884–893 Morgan Kaufmann, (2002)

[30] Bouganim, L., Pucheral, P.: *Chip-secured data access: Confidential data on untrusted servers.* In: Proceedings of the 28th international conference on Very Large Data Bases, pp. 131–142. (2002)

[31] Hacigümüs, H., Iyer, B. andMehrotra, S.: *Providing Database as a Service.* In: Data Engineering Proceedings. 18th International Conference on, pp. 29-38. (2002)

[32] Harrington, A., Jensen, C. D.: *Cryptographic Access Control in a Distributed File System.* In: Proceedings of the eighth ACM symposium on Access control models and technologies. pp. 158-165 (2003)

[33] Hacigümüş H, Iyer B, Mehrotra S.: *Efficient execution of aggregation queries over encrypted relational databases.* In: Lee, Y. Li, J., Whang, K., Lee, D. (eds.) Database Systems for Advanced Applications. LNCS, vol. 2973, pp.125-36. (2004)

[34] Iyer, B., Mehrotra, S., Mykletun, E., Tsudik, G. and Wu, Y.: *A framework for efficient storage security in RDMS*. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) Advances in Database Technology - EDBT 2004. LNCS, vol. 2992, pp. 627-628. Springer, Heidelberg (2004)

[35] Aggarwal, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Srivastava, U., Thomas, D., Xu, Y.: Two Can Keep a Secret: *A Distributed Architecture for Secure Database Services.* In: Proc. of CIDR (2005)

[36] Wang, Z., Dai, J., Wang, W., Shi, B.: *Fast Query Over Encrypted Character Data in Database.* In: Zhang, J., He, J.H., Fu, Y. (eds.) Computational and Information Science. LNCS, vol. 3314, pp. 1027-1033. Springer, Heidelberg (2005)

[37] Chung, S. S., Ozsoyoglu, G.: *Anti-Tamper Databases: Processing Aggregate Queries over Encrypted Databases.* In: Data Engineering Workshops, 22nd International Conference on, pp. 98. (2006)

[38] Wang, Z., Wang, W., Shi, B.: *Storage and Query over Encrypted Character and Numerical Data in Database. Computer and Information Technology.* The Fifth International Conference on, pp.77-81. (2005)

[39] Mykletun, E. and Tsudik, G.: Aggregation Queries in the Database-As-a-Service Model. In: Damiani, E., Liu, P. (eds.) *Data and Applications Security XX. LNCS*, vol. 4127, pp. 89-103. Springer, Heidelberg (2006)

[40] Evdokimov, S. and Günther, O.: *Encryption Techniques for Secure Database Outsourcing.* In: Biskup, J., López, J. (eds.) Computer Security – ESORICS 2007. LNCS, vol. 4734, pp. 327-342. (2007)

[41] Sanka, S., Hota, C., Rajarajan, M.: *Secure data access in cloud computing. In: Internet Multimedia Services Architecture and Application (IMSAA).* IEEE 4th International Conference on, pp.1-6, 15-17 (2010)

[42] Yan, S. Y. and Maple, C.: *On-Line Database Encryption and Authentication.* In: Tan, H.(ed.). Technology for Education and Learning.Advances in Intelligent and Soft Computing. vol. 136, pp. 363-370. Springer, Heidelberg (2012)

[43] General Services Administration. Agency Office of the Director of National Intelligence *"Security and Privacy Assurance Research"* https://www.fbo.gov [Accessed: 17-Jul-2012].

[44] *Technology, "Programming Computation on Encrypted Data y abreviado PROCEED,"* http://www.grants.gov [Accessed: 17-Jul-2012].

[45] Cloutage - *Open Security Foundation*, http://cloutage.org/. [Accessed: 26-Mar-2012].

[46] Hacigümüs, H., Iyer, B., Li, C. and Mehrotra, S.: *Executing SQL over encrypted data in the database-service-provider model.* In: Proceedings of the ACM SIGMOD international conference on Management of data. (2002)

[47] Kamara, S., Lauter, K.: Cryptographic Cloud Storage. In: Sion, R., Curtmola, R., Dietrich, S., Kiayias, A., Miret, J., Sako, K., Sebé, F. (eds.) *Financial Cryptography and Data Security. LNCS*, pp. 136-149. Springer-Verlag (2010)

[48] Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: *Order preserving encryption for numeric data.* In: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 563–574. (2004)

[49] Boldyreva, A., Chenette, N., Lee, Y, O'Neill, A.: *Order-preserving symmetric Encryption.* In Advances in Cryptology- Eurocrypt 2009 Proceedings, (2009)